Study of kernel machines towards Brain-computer Interfaces

Tian Xilan

Thèse

Présentée devant

l'Institut National des Sciences Appliquées de Rouen

pour obtenir

Docteur en Sciences
Mention Informatique

Apprentissage et Noyau pour les Interfaces Cerveau-machine

par

Xilan TIAN

Equipe: LITIS
Ecole doctorale: SPMII

soutenue le 7 Mai 2012 devant la commission d'examen

Rapporteurs :	Hélène Paugam-Moisy	-	Université de Lyon
	Valérie Louis-Dorr	-	INPL
Directeur :	Stéphane Canu	-	INSA de Rouen
Encadrant :	Gilles Gasso	-	INSA de Rouen
Examinateurs :	Marie-F Lucas	-	Ecole Centrale de Nantes
	Alain Rakotomamonjy	-	Université de Rouen

Résumé Les Interface Cerveau-Machine (ICM) ont été appliquées avec succès aussi bien dans le domaine clinique que pour l'amélioration de la vie quotidienne de patients avec des handicaps. En tant que composante essentielle, le module de traitement du signal détermine nettement la performance d'un système ICM. Nous nous consacrons à améliorer les stratégies de traitement du signal du point de vue de l'apprentissage de la machine. Tout d'abord, nous avons développé un algorithme basé sur les SVM transductifs couplés aux noyaux multiples afin d'intégrer différentes vues des données (vue statistique ou vue géométrique) dans le processus d'apprentissage. Deuxièmement, nous avons proposé une version en ligne de l'apprentissage multi-noyaux dans le cas supervisé. Les résultats expérimentaux montrent de meilleures performances par rapport aux approches classiques. De plus, l'algorithme proposé permet de sélectionner automatiquement les canaux de signaux EEG utiles grâce à l'apprentissage multi-noyaux. Dans la dernière partie, nous nous sommes attaqués à l'amélioration du module de traitement du signal au-delà des algorithmes d'apprentissage automatique eux-mêmes. En analysant les données ICM hors-ligne, nous avons d'abord confirmé qu'un modèle de classification simple peut également obtenir des performances satisfaisantes en effectuant une sélection de caractéristiques (et/ou de canaux). Nous avons ensuite conçu un système émotionnel ICM par en tenant compte de l'état émotionnel de l'utilisateur. Sur la base des données de l'EEG obtenus avec différents états émotionnels, c'est-à-dire, positives, négatives et neutre émotions, nous avons finalement prouvé que l'émotion affecter les performances ICM en utilisant des tests statisques. Cette partie de la thèse propose des bases pour réaliser des ICM plus adaptées aux utilisateurs.

Mot clés: Interface Cerveau-machine, apprentissage multi-noyaux, apprentissage semi-supervisé, TSVM-MKL, LaMKL, émotionnel ICM.

Abstract Brain-computer Interface (BCI) has achieved numerous successful applications in both clinical domain and daily life amelioration. As an essential component, signal processing determines markedly the performance of a BCI system. In this thesis, we dedicate to improve the signal processing strategy from perspective of machine learning strategy. Firstly, we proposed TSVM-MKL to explore the inputs from multiple views, namely, from statistical view and geometrical view; Secondly, we proposed an online MKL to reduce the computational burden involved in most MKL algorithm. The proposed algorithms achieve a better classification performance compared with the classical signal kernel machines, and realize an automatical channel selection due to the advantages of MKL algorithm. In the last part, we attempt to improve the signal processing beyond the machine learning algorithms themselves. We first confirmed that simple classifier model can also achieve satisfying performance by careful feature (and/or channel) selection in off-line BCI data analysis. We then implement another approach to improve the BCI signal processing by taking account for the user's emotional state during the signal acquisition procedure. Based on the reliable EEG data obtained from different emotional states, namely, positive, negative and neutral emotions, we perform strict evaluation using statistical tests to confirm that the emotion does affect BCI performance. This part of work provides important basis for realizing user-friendly BCIs.

Keywords: Brain-computer Interface, multiple kernel learning, semi-supervised learning, TSVM-MKL, LaMKL, emotional BCI.

Contents

Introduction

Les Interfaces Cerveau-Ordinateur (ICM) fournissent un nouveau canal de communication entre le système cognitif d'un individu et le monde extérieur. Elles se sont popularisées ces dernières années notamment grâce au paradigme non intrusif permettant de récolter des signaux EEG (électro-encéphalogramme) sur le scalp de l'utilisateur ; ces EGG sont ensuite traités et classés de façon à reconnaître l'intention de l'utilisateur puis ensuite commander un module extérieur (par exemple un curseur sur l'écran, épeler des mots, commander une chaise roulante, ...). La mise en oeuvre de ces applications se base sur une chaîne de traitement incluant l'acquisition des signaux EEG, le pré-traitement de ces signaux en vue d'en extraire des caractéristiques pertinentes puis la classification des signaux à l'aide d'algorithmes d'apprentissage statistique en vue de reconnaître l'intention de l'utilisateur. Les performances des ICM sont alors conditionnées par la qualité des modules de traitement et de classification des signaux. De nombreuses approches ont été proposées dans la littérature pour attaquer ces problèmes. Ainsi pour le pré-traitement, différentes méthodes comme le filtrage spectral (afin d'éliminer les bruits) et spatial (déterminer la meilleure combinaison spatiale des canaux EEG) des EEG, leur analyse en composantes indépendantes ou l'extraction de l'énergie dans des bandes de fréquence pré-déterminées. Autre titre des méthodes de classification statistique, les plus populaires sont les machines à noyau, les réseaux de neurones, l'analyse discriminante linéaire ou non-linéaire ou des méthodes bayésiennes. En général les approches les plus performantes sont l'analyse discriminante et les machines à noyaux. Ces dernières présentent une souplesse de paramétrisation à travers le noyau et une parcimonie (en termes de nombre de paramètres) de la fonction de classification leur permettant d'être très efficaces pour la classification de signaux EEG.

Dans le cadre de ce travail, nous avons attaqué la problématique de traitement et de classification des signaux EEG en utilisant les machines à noyaux et plus particulièrement l'apprentissage à noyaux multiples. L'avantage de l'approche multi-noyaux est qu'elle permet de réaliser simultanément la sélection des caractéristiques pertinentes et la classification des signaux. Elle permet de sélectionner les caractéristiques soit individuellement soit par groupes tout en conservant le caractère non-linéaire de la fonction de décision. Par ce biais différentes sources d'information (par exemple différents canaux EEG) ou différentes vues des signaux (par exemple des caractéristiques fréquentielles et/ou temporelles) peuvent être intégrées dans le processus d'apprentissage et leur importance est automatiquement déterminée par l'algorithme d'apprentissage. Nous avons proposé dans ce manuscrit deux contributions basées sur l'apprentissage multi-noyaux : la classification semi-supervisée et l'apprentissage en ligne.

La mise en œuvre pratique d'un ICM suppose la calibration du système qui nécessite l'acquisition de données étiquetées sur l'utilisateur afin d'initialiser les modules de traitement et d'apprentissage. Ce processus étant fastidieux, il apparaît nécessaire de se servir de données non-étiquetées qui peuvent être récoltées en quantité conjointement avec peu de données étiquetées afin de régler les classifieurs : c'est l'apprentissage semi-supervisé. Dans le domaine des ICM, la plupart des algorithmes d'apprentissage semi-supervisé se basent sur les machines à noyaux SVM (séparateur à vaste marge) et exploitent les données non-étiquetées selon deux points de vue. La première hypothèse dite de cluster considère que les données forment des clusters, chaque cluster représentant le classe à discriminer. Par conséquent la frontière de décision doit éviter de les traverser. Un algorithme comme le SVM transductif implémente cette hypothèse. Une autre hypothèse suppose que les données vivent sur des variétés et deux points proches sur une variété partagent probablement la même étiquette. Bien que ces deux hypothèses aient conduit à des algorithmes performants, il est difficile pour une application ICM donnée de choisir la bonne approche. Dans la première partie de ce manuscrit nous avons proposé un algorithme permettant de combiner efficacement les deux hypothèses. Pour cela, nous avons formulé le problème d'apprentissage semi-supervisé comme un SVM transductif multi-noyaux où les noyaux implémentent les deux hypothèses mentionnées. Le problème résultant est non-convexe et non-différentiable. Pour le résoudre, nous avons fait appel à l'approche DC (Difference of Convex functions) qui décompose le problème d'optimisation comme la combinaison de sous-problèmes, l'un convexe, l'autre concave. La solution finale est ainsi obtenue en résolvant successivement un problème convexe issu de la linéarisation de la partie concave

autour de la solution courante. La fonction de décision que fournit notre approche est inductive en ce sens qu'elle peut prédire le label d'un point non étiqueté n'ayant pas servi à l'apprentissage du modèle. Nous avons évalué l'approche proposée sur des benchmarks classiques en apprentissage semi-supervisé puis sur les données EEG. Les résultats expérimentaux montrent des gains en performance de classification par rapport à des approches classiques. De plus dans le cadre des applications ICM, nous avons adapté l'algorithme de façon à sélectionner automatiquement la bonne hypothèse mais aussi les bons canaux EEG pour la discrimination.

Notre deuxième contribution sur l'élaboration d'algorithmes d'apprentissage concerne l'apprentissage multi-noyaux en ligne. Cette contribution est motivée par le fait que nombre d'algorithmes de machines à noyaux et particulièrement ceux basés sur les noyaux multiples traitent les données d'apprentissage en bloc (batch). Or dans les applications ICM, il est important d'adapter le classifieur au fil du temps à cause de la non-stationnarité des signaux EEG et de leur variabilité d'un sujet à l'autre. Le classifieur doit s'adapter non seulement dans ses paramètres mais également dans sa structure et notamment dans les caractéristiques pertinentes utilisées pour la discrimination. Pour cela nous avons considéré une approche multi-noyaux où le modèle de décision recherché se base sur une combinaison linéaire de plusieurs noyaux sensés représenter différentes informations. L'importance de chaque noyau est exprimée par son coefficient. Une pénalisation de type p-norme ($p > 1$) est ensuite imposée sur les coefficients. Un choix de p proche de 1 aura tendance à privilégier une combinaison parcimonieuse des noyaux alors que de grandes valeurs de p conduiront une combinaison où tous les noyaux seront conservés. Un problème SVM basé sur ce principe conduit à une formulation duale sur laquelle nous avons bâti l'algorithme d'apprentissage multi-noyaux en ligne. En s'inspirant de l'approche LASVM qui est l'algorithme de référence pour l'apprentissage en ligne d'une machine à un noyau, nous avons développé une procédure efficace appelée LaMKL et exploitant l'algorithme SMO (Sequential Minimal Optimization) pour le problème dual. Le principe de l'approche LaMKL est le suivant : étant donnée une solution courante avec son ensemble de points supports et les coefficients des noyaux, un nouveau point d'apprentissage arrivant en ligne est intégré comme point support. Si ce point viole les conditions de stationnarité de la solution du problème dual, il est maintenu dans la solution et son paramètre correspondant est mis à jour ; autrement le point est rejeté. Cette étape d'inclusion est appelée PROCESS. Toutefois l'inclusion du nouveau point peut entrainer le fait que les anciens points supports violent les contraintes de stationnarité du problème. Pour y remédier partiellement, la paire de points violant le plus sévèrement les dites contraintes est détectée et leurs paramètres sont mis à jour. Cette étape d'amélioration de la solution est appelée REPROCESS. A chaque étape, la mise à jour des paramètres est effectuée de façon à optimiser la fonction objectif du problème dual et repose sur un problème convexe unidimensionnel. Les coefficients des noyaux se déduisent alors des paramètres des points supports de façon analytique. L'algorithme LaMKL est stochastique et répète donc pour chaque donnée d'apprentissage une étape PROCESS suivie d'une étape REPROCESS jusqu'à la satisfaction d'un critère d'arrêt. Nous avons évalué l'algorithme LaMKL sur plusieurs données réelles, et les résultats expérimentaux ont démontré que l'algorithme proposé peut permettre d'obtenir des performances en classification meilleures ou aussi bonnes que celles de l'apprentissage batch avec des temps de calcul moindres et ceci pour des dizaines voire des centaines de noyaux.

La dernière contribution présentée dans ce manuscrit porte spécifiquement sur des données ICM. Deux types de problèmes ont été traités dans cette partie. La première problématique vise à explorer des stratégies simples et efficaces de sélection des hyper-paramètres intervenant dans les modules de la chaîne de traitement des signaux ICM afin d'atteindre des performances satisfaisantes en reconnaissance de l'intention de l'utilisateur. La démarche utilisée a permis de choisir judicieusement les paramètres des modules de pré-traitement des signaux (choix des fréquences de coupure des filtres appliqués aux signaux EEG, choix de la taille des fenêtres permettant d'extraire les parties intéressantes des signaux EEG, ordonnancement des différents canaux EEG en fonction de leur pouvoir discriminatif, ...) et de classification (paramètre du noyau et paramètre de régularisation pour un SVM non-linéaire) en évitant une approche exhaustive coûteuse en temps de calcul. La méthodologie a été développée pour la compétition internationale "Mind Reading organisée dans le cadre de la conférence MLSP'2010". Elle portait sur les données ICM dites P300. Ces données sont caractérisées par le fait qu'un potentiel actif apparaît dans les signaux EEG environ 300ms après un stimulus visuel. Le bruit important qui corrompt

les signaux EEG et les phénomènes de non-stationnarité rendent la détection du potentiel extrêmement compliquée. L'application de notre méthodologie sur ces données a été satisfaisante puisque l'équipe du laboratoire LITIS dont je faisais partie a obtenu la 3ème place sur 35 participants.

La deuxième problématique pratique sur les ICM que nous avons développée porte sur la prise en compte des émotions dans les interfaces cerveau-machine. En effet, le lien indissoluble qui existe entre les émotions et la cognition inspire une tendance de recherche dans les ICM visant à inclure l'état émotionnel de l'utilisateur dans la conception de nouvelles interfaces, soit comme des indicateurs d'évaluation ou en tant que composante à insérer dans la boucle d'une ICM. Pour évaluer la faisabilité d'une telle démarche, nous avons conçu en collaboration avec des collègues de Technology University of China une série d'expériences ICM durant lesquelles un type d'émotion (positive, négative ou neutre) est induit chez un utilisateur (par stimulus visuel ou audio) qui ensuite est sollicité pour réaliser une tâche mentale. Les tâches mentales à réaliser sont de type SSEVP (Steady State Visual Evoked Potential). Sur la base des données récoltées nous avons testé si l'état émotionnel influençait les performances d'un classifieur des signaux EEG obtenus. En utilisant des tests statistiques, nous avons mis en évidence l'influence de l'émotion. Particulièrement, les émotions positives ou neutres ont tendance à fournir les mêmes performances alors qu'une émotion négative dégrade celles-ci.

La thèse est organisée en quatre grandes parties. Un état de l'art succinct des Interfaces Cerveau-Machine et des approches d'apprentissage et de traitement du signal utilisées dans les ICM est présenté dans le premier chapitre. Le second et troisième chapitres sont consacrés à nos contributions sur les algorithmes d'apprentissage statistique. Le chapitre 2 présente l'apprentissage semi-supervisé dans le contexte des multi-noyaux alors que le chapitre 3 est dédié à l'apprentissage multi-noyaux en ligne. Dans le Chapitre 4, nous présentons nos propositions sur le réglage efficace de la chaîne de traitement ICM pour la compétition "Mind Reading" et l'analyse de l'émotion sur les performances d'une chaîne ICM. Enfin, nous terminons cette thèse avec quelque conclusions et perspectives.

Introduction

A Brain-computer Interface (BCI) provides a new communication channel for human and the outside world. Various successful applications have been achieved and overwhelming attention has been attracted in recent years. As an essential component, the signal processing part acts as the translator from brain signals to the output commands. Hence, the quality of signal processing part effects the whole BCI performance.

Numerous signal processing algorithms have been proposed regarding different requirements in BCI community. Popular algorithms include kernel machines, neural networks and Bayesian methods. In this dissertation, we emphasized the kernel machines that have been proved to be efficient in BCI data analysis. However, most of the popular algorithms tend to involve a time-consuming model selection procedure with respect particular BCI applications. To attain the algorithms that can detour a complex model selection procedure, we employ the multiple kernel learning (MKL) which can combine different kinds of information and determine the importance of them automatically. When applying machine learning algorithms such as MKLs to BCIs, one needs labeled data to teach the classifier. Hence, a tedious calibration measurement is necessary before starting with BCI feedback applications. In this dissertation, the studies regarding MKL in BCI are divided into two categories according to the context of labeled data.

The first study belongs to semi-supervised learning (SSL) that using large amount of unlabeled data to improve the generalization accuracy. SSL algorithms have been applied successfully in BCI system to reduce the calibration procedure. However, most of them tends to make strong model assumptions to deal with limited labeled data. Popular hypotheses are cluster assumption and manifold assumption. The first assumption aims to enforce two training points (labeled or not) that fall in the same cluster to share the same label. The resulting algorithm such as Transductive SVM (TSVM) prefers decision function avoiding high density regions. The second assumption promotes data geometry to enforce smoothness of the labels prediction over manifolds using similarity graph-based methods. Laplacian SVM is one of the representative algorithms in this community. Both of them have some successful applications in BCIs and other real life applications. While it is unclear whether and when one assumption should be preferred over another for a new problem. It is desired to obtain a SSL algorithm that can determine the kind of model assumption automatically according to specific problems. In our work, we proposed a multiple kernel version of the famous TSVM to embed both cluster view and manifold view. By defining a pool of kernels among which implementing the manifold view, the learning problem is converted to learning a linear combination of kernel matrix from the pool for the semi-supervised applications. Due to the non-convex property inherits from TSVM, DC (difference of convex functions) algorithm was introduced iteratively to solve non-convex problem. More precisely, the original problem was decomposed into a convex sub-problem and a concave sub-problem. We then approximated the concave part by its affine minorization. As a solution, we finally get an inductive classifier extendable to unseen samples.

We first evaluated the TSVM-MKL algorithm on the benchmark SSL data sets under different experimental settings, namely, the transductive setting that evaluates the performance on unlabeled data; the inductive setting that test the performance on unseen new samples; and the semi-supervised style cross validation setting which verify the performance when one has a very few labeled samples. Experimental results showed the effectiveness of TSVM-MKL in various contexts. For the BCI applications, we tested two kinds of BCI paradigms, namely, the μ and β based BCI and the motor imagery based BCI. Both of them showed the advantages of TSVM-MKL in term of classification accuracy. As MKLs can also be viewed as an extended feature selection procedure, we can select interesting groups of features by TSVM-MKL and thus apply the idea in the channel selection problem of BCIs. Such strategy is a bonus of the proposed algorithm and has been proved to be efficient in Chapter 2.

The second study belongs to supervised learning. MKL algorithms involve heavier computation burden compared with classical kernel machines. Such drawback prevents their applications to large scale problems. In practice, many problems can be regarded as online rather than batch learning problems. We thus proposed an online MKL approach by adopting the idea of Sequential Minimal Optimization (SMO), which maximizes the gain iteratively by employing a reduced optimization problem that only involves two variables, in Chapter 3. To be able to implement online MKL, we use a non-sparse version of MKL, ℓ_p-norm ($p > 1$) MKL, knowing that sparsity can be obtained by choosing the p value close to 1. This kind of MKLs can be solved in the framework of classical SVMs by seeking the final kernel matrix as a linear combination of existing kernel matrices under the same regularity conditions. The proposed LaMKL algorithm adopts a similar idea of LASVM which is related to the SMO algorithm. This choice is motivated by the fact that among online single kernel learning procedure, LASVM has shown efficiency both in terms of computation time and generalization property. Provided aforementioned motivations and methodology, the LaMKL is implemented based on three elements: (1) indices set $\mathcal{S}$ of potential support vectors that related with the final kernel matrix in the learning process; (2) coefficients of the potential support vectors; and (3) weighted gradients for selecting working set of the two-variable reduced optimization problem. We then update the decision function by executing a so-called PROCESS procedure which involves a support vector removal step from $\mathcal{S}$. The PROCESS aims at adding new samples into the potential support vector set $\mathcal{S}$ and performing a direction search to the target dual objective function. This operation can potentially leave other violating pairs in $\mathcal{S}$. To improve the results obtained from PROCESS, we tend to perform a REPROCESS to optimize the most violating pair in $\mathcal{S}$ as well as do one iteration of batch MKL algorithm. We evaluated the LaMKL algorithms on several UCI data sets, and experimental results demonstrated that the proposed algorithm can achieve similar performance with the batch learning mode while requiring less computation cost.

The third study regards real data beyond the studies from the view of machine learning. We explored other possibilities to enhance the BCI operation in the last part of this thesis. We first implemented a BCI competition data analysis, "Mind reading, MLSP 2010 Competition" to confirm the feasibility of using simple classifier model to achieve satisfying performance in off-line BCI data analysis. Because we emphasize the simple classifier model in Chapter 4, the heavy computation cost involved in this this part of work prevents its applications in the online context.

The indivisible link between emotions and cognition inspires the researcher in BCI or human computer interface to include emotional states in the design of new interfaces: either as evaluation indicators or as components to be inserted in the interface loop. We then designed an emotional BCI system by taking account for the user's emotional state. The experiments involve the Steady State Visual Evoked Potentials (SSVEP) based BCI. The short term goal is to verify whether the user's emotional state affects the BCI performance or not, and the final goal is to adapt the classifiers to adapt the emotional feedback with subject. Based on the EEG data obtained with different emotional states, namely, positive, negative and neutral emotions, we finally proved that emotion does affect BCI performance using statistical tests. In more detail, the positive and neutral emotion effect BCI performance similarly. The negative emotion performs very differently from the rest and it tends to damage the BCI operation in terms of classification accuracy.

The whole thesis is structured as follows: a review of BCI system and learning approaches is given in Chapter 1, and we emphasized the signal processing component in BCI system. In Chapter 2, we present the inductive MKL algorithm for semi-supervised learning called TSVM-MKL and tested it in several BCI applications. The online ℓ_p-norm MKL is precised in Chapter 3. In Chapter 4, we present the implementation of "Mind reading, MLSP 2010 Competition"data analysis and the exploration of emotional BCI system. Finally, we end this thesis with some conclusions and perspectives.

Contributions

In this thesis, we improve the signal processing in BCI system from two aspects, improving the classifier model from the machine learning view and ameliorating the signal processing beyond the classifier itself. The contributions of this dissertation are summarized in what follows.

TSVM-MKL combines the two popular model assumptions of semi-supervised learning community, the cluster assumption and the manifold assumption in one learning framework. It achieves an adaptation of model assumption by benefiting the advantages of multiple kernel learning algorithms, and thus improves the performance in terms of classification accuracy on both BCI data sets and other real life data sets. As a bonus, TSVM-MKL has been shown to be efficient in the channel selection problem in BCIs. This part of work is presented in depth in Chapter 2, and had been published in [Tian 2012, Tian 2011].

LaMKL is proposed to realize an online fashion of multiple learning algorithm in the dual. Providing the advantages of ℓ_p-norm MKLs that can self-adapt the sparsity degree by adjusting the value p of ℓ_p-norm, it is easy to be applied to different applications. LaMKL also inherits almost all advantages of LASVM such as fast convergence rate and small computation cost as they adopt similar optimization strategy. Experimental results have shown that it achieves close classification accuracy while involve quite small computation cost. Detail can be found in Chapter 3. And a paper is in preparation:

- X. Tian, G. Gasso, A. Rakotomamonjy, S. Canu, "LaMKL: a fast online MKL algorithm", For *ECML-PKDD 2012*.

Beyond complex classifier how to improve the signal processing in BCI is an open question discussed in Chapter 4. Based on the "Mind reading, MLSP 2010 Competition" data analysis, we confirm that careful feature (and/or channel selection) can be a counterbalance strategy to the complex classifier model in off-line BCI data analysis in Section 4.1 and had been published in [Labbé 2010]. We then designed an affective SSVEP based BCI system to verify that emotion does affect BCI performance. For more detail, the positive and neutral emotion perform similarly with each other. The negative emotion performs differently with the rest, and it tends to damage the BCI operation in terms of classification accuracy. This part of work is supported by a Franco-Chinese project "Programme Xu Guangqi" and is presented in depth in Section 4.2. As a result of a 3-month internship of Yachen Zhu, a publication is in preparation:

- X. Tian, Y. Zhu, G. Gasso, S. Canu, G. Wu, S. Wang, "Does emotion affect BCI? ", submitted *Journal of neural engineering*.

Achievements

We list the achievements of this dissertation in what follows.

Publications

International journals

1 X. Tian, G. Gasso, S. Canu. "A multiple kernel framework for inductive semi-supervised SVM learning", *Neurocomputing*[1], vol. 2012, No. 90, pages 46–58, 2012.

2 X. Tian, G. Gasso, S. Canu. "An inductive semi-supervised algorithm for BCIs", *International Journal of Bioelectromagnetism*, vol. 13, No. 3, pages 117-118, 2011.

Proceedings of conference

3 X. Tian, G. Gasso, S. Canu, "A multi-kernel framework for inductive semi-supervised learning", In *Proc. of ESANN* 2011, Bruges, Belgium.

4 B. Labbé, X. Tian, A. Rakotomamonjy, "MLSP competition, 2010: Description of third place method", In *Proc. of MLSP*, 2010, Kittila, Finland.

5 X. Tian, G. Gasso, R. Hérault, S. Canu, "Pré-apprentissage supervisé pour les reséaux profonds", In *Proc. of RFIA*, 2010, Caen, France.

Project

- Franco-Chinese project: Programme Xu Guangqi, emotional BCI system based on statistical learning.

Master thesis

- Y. Zhu, "Kernel methods for an affective SSVEP based BCI system", *University of Science and Technology of China*, 2012, China.

[1] available online 22 March 2012

Acknowledgements

I would like to thank my supervisors, Gilles Gasso and Stéphane Canu. Without their encouragements, patience and kindness, I would not be able to finish this dissertation. They guide me to be a good researcher during the PhD study, and give me important support when I am far away from my family. To my family, they always support me unconditionally and always encourage me to do what I want to do. Foremost thanks to all of them.

I would like to thank the members of jury, Hélène Paugam-Moisy, Valérie Louis-Dorr, Marie Lucas and Alain Rakotomamonjy, to accept this manuscript. I would also specially thank Alain, for the important discussion involved in Chapter 3. Thanks to Jianzhao Qin, S V N Vishwanathan and Cornelia Herbert for the meaningful discussions and suggestions for the problems met in Chapter 2, 3 and 4.

I would also like to thank Brigitte Diarra, Sandra Hague and Jean-Francois Brulard. They have given so much help during the PhD study.

Special thanks to my colleagues and friends, Rémi Flamary, Benjamin Labé, Aurelie Boisbunon, Alina Miron, Florian Yger, Abou Keita, Yachen Zhu, Julien Delporte, Romain Hérault, Maxime Berar. It's really an interesting and happy experiences to work with them. Thanks to all the people of LITIS and all my friends.

List of Acronyms and Notations

Acronymes

ALS	Amyotrophic Lateral Sclerosis
BCI	Brain Computer Interface
NIRS	near-infrared spectroscopy
CCCP	Concave Convex Procedure
CSP	Common Spatial Pattern
CV	Cross-Validation
CVM	Core Vector Machine
DC	Difference of Convex function
EEG	Electroencephalography
ECoG	Electrocorticogram
ERP	Event-related Potential
FLD	Fisher Linear Discriminant
HMM	Hidden Markov Model
ICA	Independent Component Analysis
KKT	Karush-Kuhn-Tucker
LapSVM	Laplacian SVM
LDA	Linear Discriminant Analysis
MEG	magnetoencophalography
MLP	Multilayer Perceptrons
MKL	Multi-kernel Learning
MR	Manifold Regularization
PCA	Principal Component Analysis
PSD	Power Spectral Density
QCQP	Quadratic Constraint Quadratic Programming
QP	Quadratic Programming
RBF	Radial Basis Function
RKHS	Reproducing Kernel Hilbert Space
SAM	Surface-to-Air Missile
SDP	Semi-definite Programming
SILP	Semi-infinite Linear Programming
SMO	Sequential Minimal Optimization
SRM	Structural Risk Minimization
SSL	Semi-supervised Learning
SVM	Support Vector Machine
TSVM	Transductive SVM
VEP	Visual Evoked Potential

Notations

$\mathbf{x}$	Sample
$\mathbf{y}$	Label
$\mathbf{y}_u$	Estimated label of unlabeled samples
$\mathbf{w}$	Weight
κ	Kernel
$\mathbb{P}$	Probability
$\mathbb{R}$	Real set
R	Risk
$\mathsf{R}^{\mathrm{emp}}$	Empirical risk
$\mathsf{R}^{\mathrm{GEN}}$	Generalization performance
H	Hilbert space
$\mathcal{X}$	Input space
$\mathcal{Y}$	Output space
$\Omega(f)$	Penalization term
$\mathbf{W}$	Weight matrix
$\mathbf{K}$	Gram matrix
$\mathbf{L}$	Laplacian
$\mathbf{D}$	Diagnoal matrix
$\mathbf{M}$	Point cloud norm matrix
C	Regularization parameter
b	Bias
ξ	Slackness variable
α	Lagrange multiplier
V	Loss function on labeled data
U	Loss function on unlabeled data
g	Decision function
H_s	Hinge loss
R_s	Ramp loss
ℓ	Number of labeled samples
u	Number of unlabeled samples
γ_A	Ambient regularization
γ_I	Manifold regularization
γ	Proportion
d_k	Weight value for the k^{th} kernel in MKL
a_k	Normalization value of the k^{th} kernel in MKL
m	Number of kernels in MKL

Chapter 1

Review of Brain Computer Interfaces

Contents

A Brain Computer Interface (BCI), sometimes called a direct neural interface or a brain machine interface (BMI), is a direct communication pathway between the brain and an external device. In other words, it enables users to send commands to external devices by using brain activities only, without using peripheral nerves and muscles [Lotte 2007a]. In this chapter, we first reviewed the whole BCI system, and then describe the present-day EEG-based BCIs. In the third part we summarize machine learning algorithms used in BCI. The fourth part focuses on the limitations and challenges in current BCI research. Finally, objectives and implementations of the current PhD study are induced from the current challenges and their achievements are summarized at the end of this chapter.

1.1 Overview

The basis for BCIs is that mental activities (or thoughts) can modify the bioelectrical brain activities and therefore effect on the recorded signals. As shown in Figure 1.1, a BCI system is normally composed of four components: signal acquisition, signal processing, application and feedback [Pfurtscheller 2004].

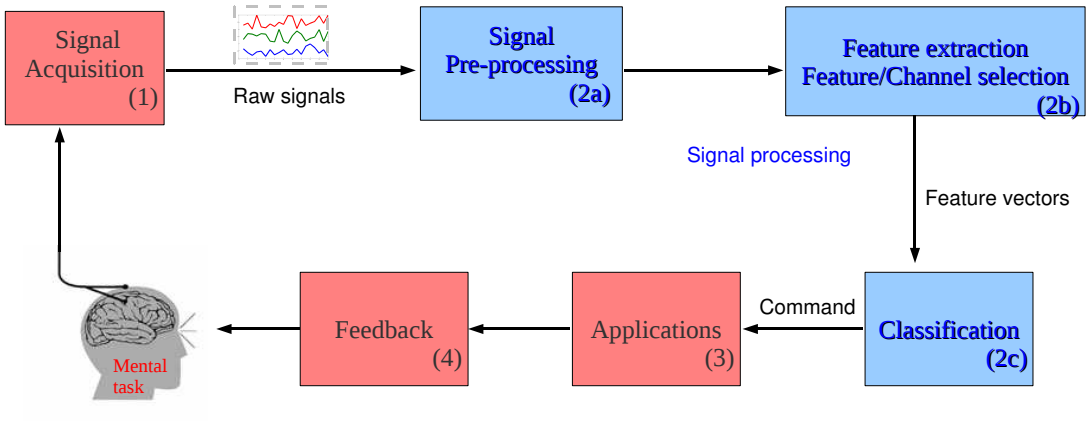


Figure 1.1: BCI components

1.1.1 Signal acquisition

BCI signal acquisition systems (corresponding (1) in Figure 1.1) are broadly divided into two classes depending on the manner in which brain signals are captured. Invasive BCIs use single-neuron activity recorded within the brain [Laubach 2000, Kennedy 1998, Georgopoulos 1986]. Non-invasive BCIs measure and record the Electroencephalography (EEG) signals using sensors arrayed across the scalp. Between these two approaches, semi-invasive approaches use epidural electrode arrays [Birbaumer 2007]. Generally, invasive BCI systems cause scar for tissue. However they can provide greater resolution of control signals compared with invasive and semi-invasive BCI systems [Berger 2007].

For invasive BCIs, most acquired signals can be divided into three types: electrocorticogram (ECoG) from subdural implanted macroelectrodes or arrays [Leuthardt 2004]; action potential spike trained from implanted microelectrodes [MacDonald 2006] and synaptic field potentials from implanted electrodes [Nicolelis 2003]. For non-invasive BCIs, three types based on EEG activities and one type based on magnetoencephalography (MEG) activities have been more throughly tested. In nature, EEG and MEG are different views of the same neural sources:

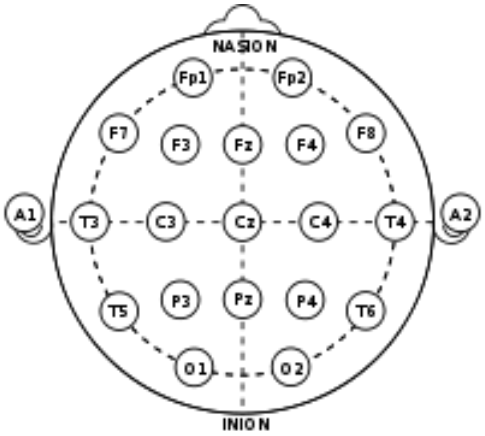
- EEG measures the potential differences on the scalp which include: Slow cortical potentials (SCP); Event-related brain potentials (ERPs), primarily the P300 that evoked approximately 300 ms after a visual stimulus appears; brain oscillations measured by EEG ranging from 4 to 40 Hz, primarily μ or sensorimotor rhythm and its harmonics (8-30 Hz from sensorimotor cortex) [Tatum 2008]. Since EEG is very sensitive, its response shows some diffusion. Generally, EEG technique is not expensive compared with other signal acquisition techniques and it is suitable for long term recording.

- MEG measures the extracranial magnetic fields. Brain oscillations measured by MEG are in the range of sensorimotor rhythm [Soekadar 2008]. It focuses on focal response and is less sensitive to tissues conductivity. Generally, it is expensive and thus only supports short recordings (ergonomic constraints).

Another new non-invasive signal acquisition tool is near-infrared spectroscopy (NIRS). It utilizes light in the near-infrared range (700 to 1000 nm) to determine cerebral oxygenation, blood flow and metabolic status of localized regions of the brain [Sitaram 2007, Coyle 2007]. Existing research can also combine several kinds of signals acquisition methods. For example, authors of [Pfurtscheller 2010] describe a hybrid BCI that simultaneously combines ERP and brain oscillations.

Current Phd study involves non-invasive BCI and mainly focuses on EEG-based BCIs. EEG system measures the impedance of all EEG electrodes. The amplitude of EEG is about 100 μv when measured

on the scalp, and about 1-2 *mv* when measured on/in the surface of the brain [Manoilov 2006]. As shown in Figure 1.2, the "international 10-20 system" is the standard naming and electrodes positioning scheme for EEG applications. It is based on an iterative subdivision of arcs on the scalp starting from craniometric reference points: nasion, inion, left and right pre-auricular points.



The letters used are:
F- Frontal lobe
T- Temporal lobe
C- Central lobe
P- Parietal lobe
O- Occipital lobe
Even number- right hemisphere
Odd number- left hemisphere
z- midline

Figure 1.2: The 10-20 System of Electrode Placement is based on the relationship between the location of an electrode and the underlying area of cerebral cortex. Each site has a letter (to identify the lobe) and a number or another letter to identify the hemisphere location (from `http://www.immrama.org/eeg/electrode.html`).

In the signal acquisition part of BCI operation, brain signals are first acquired by recording electrodes signals. The attained continuous signals are then amplified (the amplifier magnifies the amplitude of signal from μV up to several volts) and digitized (digital EEGs are acquired using bandpass filter settings typically 0.1 to 70-100 Hz) to form the raw signals which serve as the input for the signal processing block.

1.1.2 Signal processing

Signal processing (corresponding part (2) in Figure 1.1) aims at translating raw brain signals into the output commands. As shown in Figure 1.1, three components execute the translation procedure together: preprocessing and feature selection blocks transform measured brain signals such that the signal-to-noise ratio is maximized. They aim at increasing the probability of correct brain state classification. Classification block maps the obtained feature vectors into the output commands.

1.1.2.1 Signal preprocessing block

Signal preprocessing block (part (2a) in Figure 1.1). Signal preprocessing block removes artifact from the input raw signals and enhances signal-to-noise ratio of brain signals. It takes place in spatial domain (sensor or source space) and time-frequency domain. The most common types of preprocessing include artifact detection, spectral filtering and spatial filtering [Flamary 2011, Brunner 2007].

Artifact detection is one of the main tasks in EEG analysis. These artifacts are any recorded electrical potentials not originated in brain. Previous work have shown that the most severe of artifacts are due to eyeblinks and eyeball movements [Ochoa 2002]. Artifact detection attempts to find confounding signals from sources outside the brain and then attempts to remove them from the trial data or reject the trial altogether.

Spectral filtering is used to remove noise corrupting brain signals, such as slow drifts and line noise [Coyle 2010, Bashashati 2007]. It allows the user to incorporate prior information about the spectrum of brain signals by setting an appropriate cut-off frequencies of the filter [Hoffmann 2008] or the appropriate coefficients provided by wavelets decomposition for instance [Farina 2007].

Spatial filtering linearly combines signals from multiple electrodes to focus on activities at a particular location in the brain [Gerven 2009]. It is used either to focus on or reject sources based on their positions. Spatial filtering is important because raw EEG signals have a poor spatial resolution owing to volume conduction [Blankertz 2008]. Choice of a spatial filter can markedly affect the signal-to-noise ratio of a BCI system who uses μ and β rhythms (see Table 1.1 in page 12) as its signal features. Current spatial filtering strategies include:

- **Source localization** The imagination of a specific movement causes a change of the EEG in one specific location of the cortex. A spatial filter can thus be designed that only pick out information originating from the desired sources and eliminate unwanted EEG activity [Liefhold 2007].
- **Cortical mapping** Distributed source regularization involves source positions and orientations. It requires knowledge of cortical surface. One can reconstruct the potential and normal current on the scalp, skull and cortical surface, compatible with sensor measurements [Darvas 2010].
- **Surface Laplacian** [McFarland 1997] showed that EEG patterns are better detected with a surface Laplacian transformation of signals than with raw potentials.

To achieve the aforementioned preprocessing cases, many different approaches were considered in the literature. Among them, we can name Independent Component Analysis (ICA), which identifies statistically independent sources of activity, a powerful tool for artifact detection and spatial filtering [Kachenoura 2008, Naeem 2006, Wang 2004b]. It decomposes multi-channel EEGs assuming that the measured signal is a linear mixture of several independent sources in brain [Ungureanu 2004]. Common Spatial Patterns (CSP) [Hammon 2007, Blanchard 2004] and Principal Component Analysis (PCA) [Guan 2004, Chapin 1999] are also popular methods to improve the signal-to-noise ratio of brain signals. Other spatial filtering approaches include common average referencing [Cheng 2004], common spatial subspace decomposition [Fabiani 2004] and Laplace filter [Hjorth 1975].

Additionally, we should also note that the boundary between signal preprocessing and feature extraction is difficult to differentiate in some cases. Many signal preprocessing methods can also be used for feature selection, such as ICA, CSP and PCA.

1.1.2.2 Feature extraction block

Feature extraction block (part (2b) in Figure 1.1). The goal of feature extraction block is to find a suitable representation of bioelectric brain signals to simplify subsequent classification task, or detect specific thought-related patterns of brain activities. Generally, brain signal features have three main sources of information: (1) spatial information indicates the location of brain signals, it corresponds to select specific EEG channels in BCI data analysis; (2) spectral/frequential information demonstrates the varying power in some frequency bands, it corresponds to extract the power in some specific frequency bands; (3) temporal information describes the variation of signals with time, it uses the values of preprocessed EEG signals at different time points/windows as features. One feature vector could be formulated from one or more information sources and thus have the properties hereafter:

- Non-stationarity: BCI features are non-stationary since BCI signals may rapidly vary over time and more especially over different context.
- High dimensionality: EEG feature vectors can be of high dimensionality. As an example, different features are extracted from different channels and from different time segments. To cope with non-stationarity for instance, a single vector can be obtained by concatenating them.
- Small training sets: training process is time consuming and demanding for the subjects, BCI acquisition trials are short time and hence few training samples are available.

In BCI analysis, features that are generally used for classification include amplitude values of EEG signals [Vaughan 2003], time-frequency features [Coyle 2005], band powers [Pfurtscheller 1997], Power

spectral density (PSD) values [Li 2008], auto-regressive and adaptive autoregressive coefficients [McFarland 2008, Huan 2004] and Common spatial pattern (CSP) [Nasihatkon 2009]. Feature extraction methods are closely related to the specific neuromechanism(s) used by a BCI. For example, feature extraction methods in Visual Evoked Potential (VEP) based BCI are used to detect the visual evoked potentials in ongoing EEGs [Lalor 2004]. For SCP-based BCI and P300-based BCI, extracted features are mostly used to identify the specific phenomenon in brain signals [Labbé 2010, Rothman 1970].

1.1.2.3 Classification block

Classification block (part (2c) in Figure 1.1) transforms extracted features into the commands that can be executed by outside devices. This process involves identifying the optimal decision boundaries for different classes of brain signals in the feature space. A classification algorithm must be dynamic to accommodate and adapt to the continuing changes of brain signals [Mak 2009]. Classification block is covered by the field of machine learning and classification performances mainly depend on: the type of classifier, the number of extracted features, the amount of training data and the experimental paradigm.

Many applications and adaptation of standard learning methods coupled with signal processing were reported in literature. A good review of these methods and guidelines and analyses of the choice of a particular classifier suited for a particular BCI application are exposed in [Lotte 2007a]. In this work we choose, for simplicity sake, to give an overview of used tools according to family of learning methods.

Kernel machines and large margin principle The most well known kernel machine is linear Support Vector Machine (SVM) which expresses the decision function as a hyperplane maximizing separation between classes that is the margin. Nonlinear SVM is attained by using the "kernel trick" [Schölkopf 2002]. The good generalization properties and computational simplicity of kernel trick make them promising methods for BCI systems [Sitaram 2007, Labbé 2010, Kaper 2004].

Neural networks They represent another family of nonlinear decision functions. They are obtained by stacking layers of artificial neurons. Each layer performing a nonlinear transform of its inputs [Wasserman 1989]. In BCI domain, many successful applications such that Multilayer Perceptrons (MLP) [Haselsteiner 2000] and Radial Basis Function (RBF) neural network [Hoya 2003] have been reported in literature.

Bayesian methods These methods are generative contrast to kernel machines and neural networks that are discriminative approaches.

- The most representative approaches in BCI application are undoubtedly Linear Discriminant Analysis (LDA) and Fisher Linear Discriminant (FLD) [Duda 2000] which are linear methods for classification. The assumption behind these methods is the data of different classes can be modeled by Gaussian distributions. LDA and FDA have low computational burden which make them suitable for BCI system [Subasi 2010, Blanchard 2004].

- To cope with non-stationarity and time evolving features of certain BCI applications, authors have considered the Hidden Markov Model (HMM) [Duda 2000] able to deal with sequences classification. Reported results have revealed that they are promising classifiers for BCI systems [Sitaram 2007].

- Gaussian processes based methods can naturally provide probability outputs for identifying a trusted prediction [Duda 2000]. Such predictive probabilities can be used for post-processing and have attained satisfying quantity for further processing for a BCI system [Zhong 2008].

Ensemble methods Instead of learning a single decision function based on previously exposed methods, ensemble approaches rather consider combinations of classifiers [Hastie 2009, chapter 16]. Among ensemble approaches, one can cite boosting algorithms [Freund 2003], bagging [Bauer 1999] or random forests [Liaw 2002]. For a nice review of ensemble methods, we refer the reader to see [Rothman 2010]. Ensemble methods such as Voting or Stacking may be preferred for BCI applications [Lotte 2007a]. Successful applications could be ensemble of several classifiers from the same family [Rakotomamonjy 2008b] or of different types [Lee 2003].

To summarize, linear classification methods have less computation burden and thus be suitable for online BCI systems. However, the main drawback is their linearity that can provide poor results on complex nonlinear brain signals [Garcia 2003]. Beyond the simple linear ones, other models profit better generalization ability or classification accuracy. Meanwhile, they also bring a larger computation complexity. To choose a suitable classification model, one needs to achieve a trade-off between model complexity and system performance in practice.

Generally, all these algorithms are based on fully supervised data, that is, enough feature vectors and their class labels are available during the classification process. Unfortunately, that is not the case for some EEGs data because labeling could be costly, time consuming or inappropriate. When a few labeled data is available and a certain amount of unlabeled data can be leveraged to unravel the marginal distribution of BCI data, existing research resorts to Semi-supervised Learning (SSL) [Chapelle 2006] to circumvent such difficulty.

Semi-supervised learning employs unlabeled data to improve the generalization ability of classifier. It has been introduced into BCI data analysis recently. Successful references include self-training algorithm [Qin 2007, Li 2008], co-training algorithm [Panicker 2010], transductive SVM [Liao 2007], graph-based methods [Zhong 2009] and multiple kernel learning methods [Tian 2012]. Compared with supervised algorithms, they have shown better generalization performances while dealing with unlabeled data.

BCI system can involve high dimensional features. Efforts have been made in literature to couple most learning methods presented here with features selection and/or useful channel selection. In this thesis, the learning algorithms we have designed are oriented towards automatic features (independently or by groups) selection in the framework of kernel based methods.

Until now, the signal acquisition block and the signal processing block determine the information transfer rate (ITR) together. The ITR, given in bits per trial, is an important evaluation criteria for BCI performance in practice [Obermaier 2001].

1.1.3 Applications and feedback

Applications The feature vectors processed by the classification block provide the output (normally as a discrete command) to operate an outside device. Existing BCI research has been targeted as assistant communication devices for disabled people [Mak 2009,Soekadar 2008] or a new modal interaction for healthy users [Reuderink 2008]. Hence, potential application areas include: rehabilitation/functional control [Birbaumer 2007], mobility [Lin 2011], leisure/gaming/creativity [Mühl 2010], health/fitness, smart homes [Holzner 2009] and niche industrial/professional applications (military, specialised operators).

In clinical applications, potential BCI users could be individuals who are severely disabled by disorders such as Amyotrophic lateral sclerosis (ALS), cerebral palsy, brainstem stroke, spinal cord injuries, muscular dystrophies or chronic peripheral neuropathies [Mak 2009]. Benefit from BCIs, they could operate a spelling program on a computer screen by letter selection [Friman 2007], execute cursor control on a computer screen [Li 2010b], they could drive a wheelchair [Rebsamen 2007a, Rebsamen 2007b] or other assistant devices [Cincotti 2008], manipulate a robotic arm [Taylor 2003] or control movement of a paralyzed arm through a neuroprosthesis [Müller 2005, Pfurtscheller 2003a]. Figure 1.3 presents several examples of such applications. Epilepsy and attention regulation via brain regulation were also shown to be possible in recent research [Birbaumer 2007].

BCI intended for non-disabled users are designed for somewhat different applications than those for disabled users. They are expected to drive the penetration field of BCI in an even wilder range. Existed articles have reported many successful BCI designs such as, to bank a full motion aircraft simulator [Middendorf 2000b], to move a map in two dimensions [Trejo 2006], to turn or learn left or right in visually elaborate immersive 3D games [Pfurtscheller 2006a] or to evaluate user experience [Bos 2011]. According to some authors, the use of BCI is a tantalizing possible step towards the revolution of computer games [Gerven 2009].

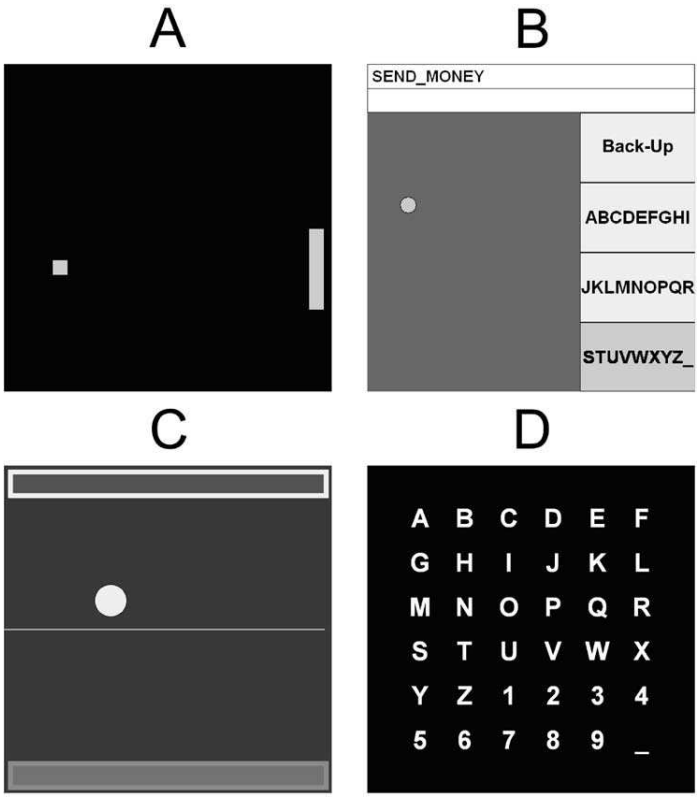


Figure 1.3: Examples of BCIs applications. (A) Sensorimotor rhythm control of cursor movement to a variable number of selections. (B) Simple spelling application using sensorimotor rhythm control. (C) Slow cortical potential (SCP) control of cursor movement to two possible selections. (D) P300-based spelling application. In (A)–(C), the cursor moves from left to right at a constant rate with its vertical movement controlled by the user's brain signals. In (D), rows and columns of the matrix flash in a block-randomized fashion (from [Schalk 2004]).

Feedback During the step of interaction with the application (part (3) in Figure 1.1), it is particularly essential to provide a feedback (part (4) in Figure 1.1) to the subject, concerning the mental state that has been recognized by the system [Lotte 2008]. With such feedback, the BCI forms a closed loop system composed of two adaptive controllers (brain and computer). In practice, continuous feedback can be provided immediately and smoothly usually by a visual cue (e.g., movement of mouse cursor) [Guger 2001,Neuper 1999], an audio cue [Omar 2011] or a haptic cue [Kauhanen 2006]. Beside continuous, feedback can also be discrete. One instance of discrete feedback is implemented by Graz BCI[1]. They presented each mental task as a colored ball. The ball lights up when the EEG sample is classified as belonging to a corresponding task. Feedback can also be graded. Graded feedback is proportional to some variables. For example, when employing "+" and "-" to indicate whether a EEG trial has been correctly classified, then the size of the "+" and "-" signs identified how well the classifier recognized the mental tasks [Pfurtscheller 1997]. Generally, an effective BCI system must provide feedback to the user and thereby substitute for the missing part of the conversation.

1.1.4 Summary

This section aims at reviewing state-of-the-art methods for each BCI component. Any BCI, regardless of its recording methods or applications or feedback, consists of four essential elements. They are signal

[1] http://bci.tugraz.at/

acquisition, feature extraction, feature translation and device output. The four elements are managed through the system's operating protocol which include: how the system is turned on and off; whether communication is continuous or discontinuous; whether message transmission is triggered by the system or by the user; the sequence and speed of interactions between user and system and what feedback is provided to the user.

Now we have depicted the general structure of BCI system, we will present the most prominent paradigms related to EEG-based human machine communication in the next section.

1.2 Present-day EEG-based BCIs

1.2.1 EEG based BCI paradigms

In BCI systems, electrophysiological sources refer to the neurological mechanisms or processes employed by a BCI user to generate control signals. Representative works categorized current BCI systems as four main groups based on the electrophysiological signals they use [Wolpaw 2002, Bashashati 2007]. These groups are respectively VEP, SCP, P300, μ and β rhythms and are shortly described hereafter.

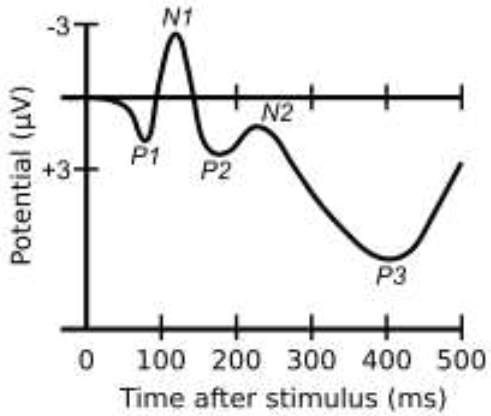


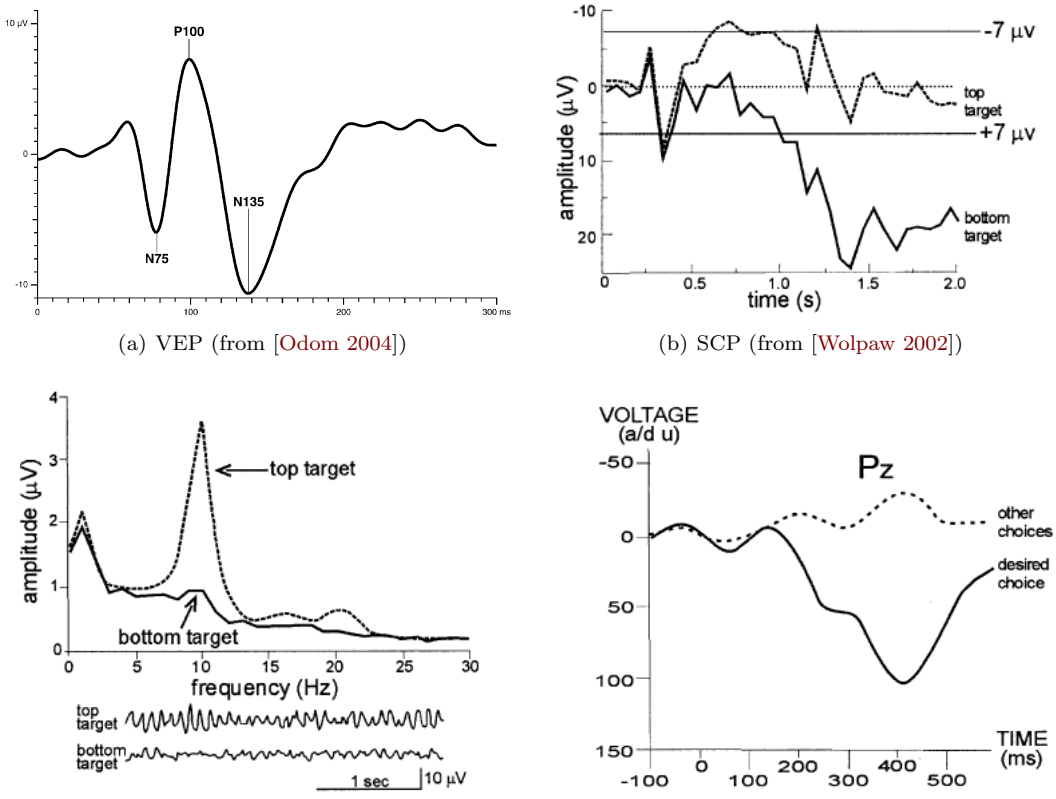
Figure 1.4: An ERP is any measured brain response that is directly the results of thought or perceptron. The wave shows several ERP components, including the N100 and P300 (from Wikipedia).

- **VEP** reflects the visual information processing mechanism in the brain. It is a tool that can identify a target on which a user is visually fixated via analysis of concurrently recorded EEG [Wolpaw 2002]. VEP based BCIs depend on muscular control of gaze direction. They are used to detect the visual evoked potentials in ongoing EEGs.

- **SCP** reflect changes in cortical polarization of the EEGs lasting from 300 ms up to several seconds. SCP based BCIs are independent BCIs; they aim at identifying the specific SCP phenomenon in brain signals.

- **P300** is an extremely robust ERP (as shown in Figure 1.4) elicited by infrequent, task-relevant stimuli. It reflects variation in attentional processing [Soekadar 2008]. P300 based BCIs are also dependent BCIs; they have been employed successfully in many applications by detecting the P300 potentials.

- **μ and β rhythms** are closely associated with motor inhibition. The versatility to cognitive manipulation of μ and β rhythms makes them ideal candidate to drive a BCI device.



(a) VEP (from [Odom 2004])

(b) SCP (from [Wolpaw 2002])

(c) SensoriMotor rhythm (from [Wolpaw 2002])

(d) P300 (from [Wolpaw 2002])

Figure 1.5: Neuromechanisms involved in current BCIs.

These four categories form the most important parts of BCI community and are shown in Table 1.1 and Table 1.2 with the involved attributes: definition, main features, some achievements, advantages and disadvantages. As shown in the tables, VEP and P300 based BCI systems normally require attention and gaze control which is intolerable for locked-in patients. Thus, they can be regarded as dependent BCI systems. Both of them benefit easy system configuration and little or even no initial user training. Generally, they have a higher information transfer rate. Temporal and spatial information is crucial for them. SCP and μ and β rhythms based systems are independent BCIs, that is, they do not depend on any normal output pathways of brain. SCP based BCI systems allow voluntary regulation of activity in different brain areas, while they need extensive training and have a lower information transfer rate. μ and β rhythms based BCIs have high long-term stability and consistency. Meanwhile, they need quantification and visualization of different mental tasks.

1.2.2 Representative BCI systems

In this section, we present several popular BCI softwares and summarize representative BCI systems proposed by some leading BCI groups.

- **BCI2000**[2] is a general-purpose system and development platform for BCI research. It consists of

[2] http://bci2000.org/BCI2000/Home.html

Table 1.1: Current BCI systems based on different Neuromechanisms

Neuromechanism		Attributes
VEP	Definition:	VEPs are small changes in the ongoing EEG (originated from the occipital cortex) caused by visual stimulus (e.g. flashing light) (as shown in Figure 1.5(a)). If a visual stimulus is presented repetitively at a rate of 5-6 Hz or greater, the response is termed steady-state visual evoked potentials (SSVEP). For low stimulus rates (less than 2Hz), a transient VEP (TVEP) is generated [Wang 2008].
	Features:	VEP has a special time course. The temporal and spatial information is crucial. VEP amplitude is mostly used.
	Achievements:	phone number selection [Cheng 2002, Gao 2003]; 3D immersive games [Lalor 2004]; cursor control [Lee 2010, Liu 2010, Lee 2010, Trejo 2006], flight simulator control [Middendorf 2000a]; muscle stimulator operation [Middendorf 2000a]; navigation in virtual environment [Martinez 2007].
	Advantages:	robustness of system performance, easy system configuration, little user training [Shen 2009], high accuracy [Devlaminck 2009] and a high information transfer rate [Wang 2010b, Liu 2010].
	Disadvantages:	requires attention and intact gaze control, requires analysis of subjects' EEGs in a natural environment and have limited degree of freedom [Gu 2009, Wolpaw 2002].
μ and β rhythms	Definition:	μ rhythms (8-12 Hz) and β rhythms (13-30 Hz) originate in the sensorimotor cortex and are displayed when a person is not engaged in processing sensorimotor inputs or in producing motor inputs (as shown in Figure 1.5(c)). They are mostly prominent in frontal and parietal locations [Kübler 2001]. Movement or preparation for movement is typically accompanied by a decrease in μ and β rhythms, particularly contralateral to the movement. This decrease is labeled "event-related desynchronization" or ERD. Its opposite, rhythm increase, or "event-related synchronization" (ERS) occurs after movement and relaxation. ERD and ERS do not require actual movement, they occur also with motor imagery (e.g. imagined movement) [Bashashati 2007, Wolpaw 2002].
	Features:	The spectral and spatial information is crucial. μ and β amplitude, ERD and ERS of μ and β rhythms are usually used. More elaborated features include Common Spatial Patterns (CSP) feature [Qin 2007], band powers [Pfurtscheller 1997], Power Spectral Density (PSD) values [Chiappa 2004], AutoRegressive (AR) and Adaptive AutoRegressive parameters [Pfurtscheller 1998], time-frequency features [Wang 2004a], inverse model-based features [Congedo 2006].
	Achievements:	cursor control on a screen by μ or β rhythm amplitude [D. 2007, McFarland 2006]; translates motor imagery (e.g. imagination of hand movements or whole body activities) into the outputs (selection a letter or extension of a lighted bar) [Pfurtscheller 2003b]; navigate robot by motor imagery [Malechka 2011], virtual reality navigation [Hashimoto 2010].
	Advantages:	can support independent BCIs (does not depend on any normal output pathways of brain); high long-term stability and consistency [Pfurtscheller 2006b, Neuper 2005].
	Disadvantages:	need quantification and visualization of different mental tasks [Vuckovic 2008].

Table 1.2: Current BCI systems based on different Neuromechanisms (continued)

Neuromechanism		Attributes
SCP	Definition:	SCPs are slow, non-movement potential changes generated by the subject (as shown in Figure 1.5(b)). In nature, they are negative or positive polarization of the EEGs or magnetic field changes in the magnetoencephalogram (MEG) that last from 300 ms to several seconds [Birbaumer 1999a, Hinterberger 2004a].
	Features:	amplitude value of SCP.
	Achievements:	control movement of an object on computer screen [Hinterberger 2004b, Birbaumer 2000], SCP-based spelling BCI [Birbaumer 1999b], seizure suppression of patients by SCP regulation [Kotchoubey 2001].
	Advantages:	allow voluntary regulation of activity in different brain areas with specific behavioural and cognitive consequences [Kotchoubey 2001]; can support independent BCI; can improve control of brain activity in short training period [Lotte 2007b].
	Disadvantages:	low information transfer rate; the users need extensive training which last over weeks or months [Wolpaw 2002], needs professional attention and continuous technical support.
P300	Definition:	Infrequent or particularly significant auditory, visual, or somatosensory stimuli, when interspersed with frequent or routine stimuli, typically evoke in the EEG over the parietal cortex a positive peak at about 300 ms after the stimulus is received (as shown in Figure 1.5(d)). This peak is called P300 [Allison 2003], The amplitude of the P300 signal is inversely related to the rate of rare event presented to the user. P300 is an extremely robust event-related potential (ERP).
	Features:	The temporal and spatial information is crucial. Features include amplitude values of P300, time domain features, linear combinations of the EEG samples' amplitudes [Lotte 2009], autoregression noise of multichannel time series [He 2010], time-frequency features between 0 and 30 Hz [Yang 2007], coefficients of wavelet transform [Wang 2010a].
	Achievements:	P300 speller [Farwell 1998], remote control device, such as wheelchair [Wang 2005, Rebsamen 2007b], movement control in a virtual environment [Piccione 2006].
	Advantages:	need no initial training, fastest acquisition and processing rate [Soekadar 2008], easily controlled and stable in performance (many P300-based BCI systems have achieved 100% accuracy) [Birbaumer 2007], is one of the most studied BCI paradigm. For ALS patients with normal vision and eye control, P300-BCI have shown the most promising results.
	Disadvantages:	relies on the selective attention and gaze control, which is intolerable for locked-in patients; needs large samples to calibrate the BCI [Lotte 2009]. As P300 used in a BCI is likely to change over time, thus P300 BCI need high adaptation by the translation algorithm. Moreover, P300 amplitude depends on the subject age [Dias 2005].

four modules: operator, source, signal processing and application. BCI2000 can also be used for data acquisition, stimulus presentation and brain monitoring applications [Schalk 2004].

- **OpenVibe**[3] is a software platform dedicated to designing, testing and using BCI. The package includes a designer tool to create and run custom applications, along with several pre-configured and demonstration programs which are ready for use. It can be used to acquire, filter, process, classify and visualize brain signals in real time.

- **BioSig**[4] is an open source software library for biomedical signal processing, featuring for example the analysis of biosignals such as EEG, ECoG, ECG and EMG. It provides solutions for data acquisition, artifact processing, quality control, feature extraction, classification; modeling and data visualization.

- **BCI++**[5] is an object-oriented BCI prototyping framework. The BCI++ features two main modules: one is dedicated to signal acquisition, storage and visualization, real-time execution and management of custom algorithms and another graphic user interface module.

- **The Berlin BCI (BBCI)**[6] aims at improving the detection and decoding of brain signals acquired by electroencephalogram (EEG). It focusses on new sensor technology, improved understanding of the brain and the analysis of brain waves using modern machine learning methods.

- **The Wadsworth BCI**[7] has concentrated on defining the topographical, spectral, temporal features of μ and β rhythm control and on optimizing the mutually adaptive interactions between the user and the system.

- **The Graz BCI**[8] was one of the first online EEG-based BCI. Various applications, including spelling devices, computer games, functional electrical stimulation and navigation in virtual environments, have been developed and tested in healthy users and several patient populations.

1.2.3 Summary

We present four basic EEG based BCI paradigms in this section. A comparison from perspectives of achievements, advantages and disadvantages is implanted in the first part. This part of work provides a basis criteria and guide for BCI researchers. Finally, we present several representative BCI systems proposed by the leading BCI groups.

In the current PhD work, we have focused on the design of learning algorithms to address some of the issues in BCI systems. Our first contributions being mostly related to classification tools together with feature/channel selection (or combination), we review in the next section basis of machine learning. Then we present the most representative methods, namely kernel methods and deep architecture, we rely on in the remaining of this document.

1.3 Machine learning for BCIs

Most BCI systems contain as a core part a machine learning algorithm, which learns from training data and yields a function that can be used to discriminate different brain activity patterns. Such machine learning part adapts the BCI system to a particular subject or context. In this session, we first present the background of machine learning and then introduce several learning algorithms that are mostly involved in BCI data analysis.

[3] http://openvibe.inria.fr/
[4] http://biosig.sourceforge.net/
[5] http://www.sensibilab.campuspoint.polimi.it/
[6] http://www.bbci.de/
[7] http://www.wadsworth.org/bci/
[8] http://bci.tugraz.at/

1.3.1 General framework

This section aims at giving a general setting for statistical machine learning. Available data is assumed to be a set of n examples $\{(\mathbf{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y}\}_{i=1}^{n}$, where $\mathbf{x} \in \mathcal{X}$ is i.i.d drawn from an unknown probability distribution $\mathbb{P}(\mathbf{x})$ and $y \in \mathcal{Y}$ (if available) i.i.d drawn from an unknown conditional probability distribution $\mathbb{P}(y|\mathbf{x})$. Sets $\mathcal{X}$ and $\mathcal{Y}$ are considered as the input space and output space separately and the unknown joint probability $\mathbb{P}(\mathbf{x}, y) = \mathbb{P}(\mathbf{x})\mathbb{P}(y|\mathbf{x})$ is defined over $\mathcal{X} \times \mathcal{Y}$. Let f be a function from $\mathcal{X}$ to $\mathcal{Y}$. To measure the adequacy of f and its prediction, one considers a loss function V to evaluate the discrepancy between predicted output $\hat{y} = f(x)$ and the ground truth y which is defined as,

$$\begin{aligned} V: \mathcal{Y} \times \mathcal{Y} &\longrightarrow \mathbb{R}^+ \\ (\hat{y}, y) &\longmapsto V(\hat{y}, y) = V(f(\mathbf{x}), y). \end{aligned} \tag{1.1}$$

For a given known hypothesis space $\mathcal{H}$, the target of a learning problem is to find the best decision function which is solution of the expected risk minimization,

$$f^* = \underset{f \in \mathcal{H}}{\operatorname{argmin}} \ \mathsf{R}(f) \tag{1.2}$$

where $\mathsf{R}(f) = \int_{\mathcal{X} \times \mathcal{Y}} V(f(x), y) d\mathbb{P}(\mathbf{x}, y)$ is the expectation of risk. When $\mathcal{H}$ is chosen large enough, f^* is arbitrary closed to the best possible prediction. Because $\mathbb{P}(x, y)$ being unknown, problem (1.2) cannot be solved in practice. One strategy is the use of empirical risk in place of $\mathsf{R}(f)$, that is,

$$\mathsf{R}^{\text{emp}}(f) = \frac{1}{n} \sum_{i=1}^{n} V(f(\mathbf{x}_i), y_i). \tag{1.3}$$

But determining the decision function $\tilde{f} = \underset{f \in \mathcal{H}}{\operatorname{argmin}} \ \mathsf{R}^{\text{emp}}(f)$ may be unsuitable. Indeed, if $\mathcal{H}$ is sufficiently large and dense, it is possible that one find a function f that perfectly fits to training data but fails to generalize to new incoming data. This phenomenon is known as over-fitting. To ensure empirical risk minimization will well-behave, the uniform convergence condition,

$$\lim_{n \to \infty} \ \mathbb{P}\{\sup_{f \in \mathcal{H}} |\mathsf{R}(f) - \mathsf{R}^{\text{emp}}(f)| > \varepsilon\} = 0, \quad \forall \varepsilon > 0$$

should hold. However, this asymptotic condition does not indicate how close the empirical risk $\mathsf{R}^{\text{emp}}(f)$ is from the expected risk $\mathsf{R}(f)$. More refined and non-asymptotic condition was proposed [Vapnik 1995, von Luxburg 2011]: with probability at least $1 - \eta$, $\eta > 0$ the risk should be uniformly bounded by

$$\mathsf{R}(f) \ \leq \ \mathsf{R}^{\text{emp}}(f) + \Phi(n, \text{capacity}(\mathcal{H}), \eta), \quad \forall f \in \mathcal{H} \tag{1.4}$$

where $\Phi(n, \text{capacity}(\mathcal{H}), \eta)$ is a deviation term and $\text{capacity}(\mathcal{H})$ accounts for the capacity of hypothesis space $\mathcal{H}$. It indicates how rich is the set of functions we are searching in. VC dimension and Rademacher complexity are two ways to implement $\text{capacity}(\mathcal{H})$. It is expected that the term $\Phi(n, \text{capacity}(\mathcal{H}), \eta)$ decreases while the number of samples n is growing and it increases with richness of $\mathcal{H}$. Hence, one has to achieve a trade-off between data fidelity $\mathsf{R}^{\text{emp}}(f)$ and model complexity. Common approach in machine learning domain to achieve this trade-off is Structural Risk Minimization (SRM) [Vapnik 1995]. It can be seen as a three-step procedure:

- **Step 1** Define nested sets $\mathcal{H}_1 \subset \mathcal{H}_2 \subset \cdots \subset \mathcal{H}_i \subset \cdots \subset \mathcal{H}$ of hypothesis spaces.
- **Step 2** Retrieve the function in each set minimizing the empirical risk,

$$\hat{f}_i = \underset{f \in \mathcal{H}_i}{\operatorname{argmin}} \ \mathsf{R}^{\text{emp}}(f).$$

- **Step 3** Select the best model f_{i^*} according to the following optimization problem,

$$i^* = \underset{i}{\operatorname{argmin}} \ \mathsf{R}^{\text{emp}}(\hat{f}_i) + \Phi(n, \text{capacity}(\mathcal{H}_i), \eta).$$

The SRM principle defines a trade-off between the approximating quality of samples and the complexity of approximating function. A common way to implement this principle is to define a sequence $\{\mathcal{H}_i, i \in \mathbb{N}\}$ based on an increasing sequence of real scalars $\{t_i, i \in \mathbb{N}\}$ such that $\mathcal{H}_i = \{f \in \mathcal{H} | \Omega(f) < t_i\}$, where $\Omega(f)$ is a penalization term that controls complexity of f (typically $\Omega(f) = \|f\|^2_{\mathcal{H}}$). In this case, very often (due mostly to convexity), step 2 leads to a penalized (or abusively regularized) cost minimization:

$$\hat{f}_i = \underset{f \in \mathcal{H}}{\operatorname{argmin}} \ \mathsf{R}^{\text{emp}}(f) + \lambda_i \Omega(f) \tag{1.5}$$

where $\lambda_i \geq 0$ is a scalar roughly inversely proportional to t_i. Then, model selection of step 3 is turned into the selection of the "best" regularization parameter λ_i. As SRM principle yields to mitigated performances [Hastie 2009, chapter 7] model selection methods, in practice, range from simple yet powerful cross validation to the optimization of cost functions penalized for model complexity as AIC, BIC [Guyon 2010]. Let $\mathsf{R}^{\text{GEN}}(f)$ be the generalization performance used to assess the quality of the models. The procedure pursued in this thesis can be summed up by the two-stage optimization problem [Guyon 2010]

$$\begin{aligned} i^* &= \underset{i}{\operatorname{argmin}} \quad \mathsf{R}^{\text{GEN}}(\hat{f}_i) \quad \text{subject to} \\ \hat{f}_i &= \underset{f \in \mathcal{H}}{\operatorname{argmin}} \quad \mathsf{R}^{\text{emp}}(f) + \lambda_i \Omega(f) \end{aligned}$$

Once the model chosen, the learned function is $\hat{f}_i^*$ or any improved post selection estimates. Obviously if hyper-parameters other than λ_i are introduced in the learning problem, their optimization is carried out by embedding them in the two-level procedure.

Now let us provide some details about elements involved in the general learning framework, that is, the loss function V, the regularization (or penalization function) Ω and hypothesis space $\mathcal{H}$.

1.3.2 Learning problems and their criterions: V and Ω

Formulation of the learning problem is rather broad. In this part, we consider the main ones among numerous of specific problems in machine learning domain: supervised learning, unsupervised learning and semi-supervised learning (for a more detailed presentation see [Alpaydin 2004]).

Supervised learning When the output being a real value $y \in \mathbb{R}$ or a multi-dimensional vector, the learning problem is a regression estimation problem. When the output space $\mathcal{Y}$ being a discrete set, the learning problem is a classification (or pattern recognition) problem. Classical cases include binary classification ($\mathcal{Y} = \{0, 1\}$ or $\mathcal{Y} = \{-1, +1\}$) and multi-class classification [Vapnik 1995]. Another instance of supervised learning is ranking which reduces detailed measures to a sequence of ordinal numbers. For example, the output y_{ij} indicates whether $\mathbf{x}_i \geq \mathbf{x}_j$ ($\mathbf{x}_i$ ranked over $\mathbf{x}_j$) holds.

Unsupervised learning Labels are not always available in some circumstances. When only unlabeled data are available, one can resort to unsupervised learning to find the hidden structure in data. Density estimation and clustering are two typical unsupervised learning problems. Since the examples are unlabeled, there is no error or reward signals to evaluate a potential solution [Hinton 1999]. However in clustering, there exist performance measures as cluster stability or separability [Hastie 2009, chapter 14] and for instance BIC criterion in density estimation to assess model quality.

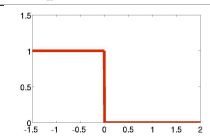
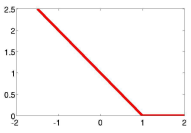
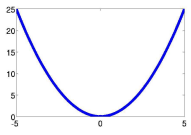
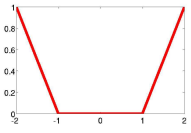
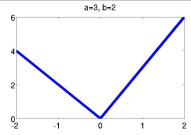
Semi-supervised learning Semi-supervised learning addresses the learning problem by using large amount of unlabeled data [Chapelle 2006] together with labeled data to build a "better" classifier. The optimization problem can be casts as,

$$\min_{f, \mathbf{y}_u} \ \mathsf{R}^{\text{emp}}(f) + \frac{1}{n_u} \sum_{i=1}^{n_u} V(f(\mathbf{x}_{ui}), y_{ui}) + \lambda \Omega(f) \tag{1.6}$$

where $\mathbf{x}_{ui}$ denotes the unlabeled data, $\mathbf{y}_u = \begin{bmatrix} y_{u1} & \dots & y_{un_u} \end{bmatrix}^\top$ is the vector of unknown labels to be estimated and n_u is the number of unlabeled samples while $\mathsf{R}^{\text{emp}}(f)$ involves only the available labeled samples.

Now we turn to the loss function and penalization term used for these learning problems.

Table 1.3: Some basic loss functions.

Loss	Definition	Properties	
0-1	$V(f(\mathbf{x}), y) = \begin{cases} 0 & \text{if } y = f(\mathbf{x}) \\ 1 & \text{if } y \neq f(\mathbf{x}) \end{cases}$		non-convex singular
Hinge	$V(f(\mathbf{x}), y) = (1 - yf(\mathbf{x}))_+ = \max\{0, 1 - yf(\mathbf{x})\}$		convex singular
Square	$V(f(\mathbf{x}), y) = (y - f(\mathbf{x}))^2$		convex differentiable
ε-insensitive loss	$V(f(\mathbf{x}), y) = \begin{cases} 0 & \text{for } \lvert f(x) - y\rvert < \varepsilon \\ \lvert f(\mathbf{x}) - y\rvert - \varepsilon & \text{otherwise} \end{cases}$		convex singular
Asymmetric ℓ_1 loss	$V(f(\mathbf{x}), y) = \begin{cases} a\lvert y - f(\mathbf{x})\rvert & \text{if } y - f(\mathbf{x}) > 0 \\ b\lvert y - f(\mathbf{x})\rvert & \text{otherwise} \end{cases}$		convex singular

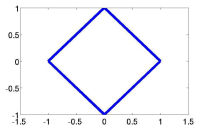
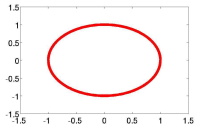
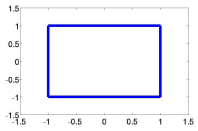
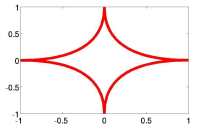
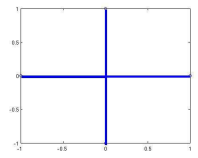
Loss function As shown in previous section, loss function is used to assess accuracy of the decision function. We list some basic loss functions that we are particularly interested in this dissertation. Based on these basic ones, one can also develop new loss functions as we do in chapter 2 (see section 2.1.2 and the ramp loss function). As shown in Table 1.3, 0-1 loss and Hinge loss are only used for classification learning tasks. The remaining ones are mainly used for the regression problem.

Regularization (or Penalization) As we have seen before, the regularization (or penalization) term can be seen as a way to measure the complexity of a decision function. Table 1.4 lists some regularizers used in the machine learning domain. Regularisation term may also have other interpretations. It can be seen as a way to obtain meaningful results from ill-posed problems and to control for over-fit. Furthermore, the regularizer $\Omega(f)$ can be regarded as a mean to introduce a priori knowledge is a bayesian view. It can also be used to impose model properties such as smoothness and sparsity.

- For smoothness: geometrically, regularization for smoothness means that we seek the least rough function that gives a certain degree of fit to the observed data. Typically the ℓ_2 norm can be used to control smoothness of the learned function.
- For sparsity: sparseness is an important criteria for the solutions of learning problem. A vector $z \in \mathbb{R}^d$ is said to be sparse if the condition $\|z\|_0 \leq d$ holds. Sparsity is normally divided into two categories in machine learning domain, that is, unstructured sparsity and structured sparsity.

Unstructured sparsity, for example ℓ_1 norm (which has led to Lasso problem [Tibshirani 1996]) in convex case and pseudo ℓ_p $(0 < p < 1)$ norm in non-convex case, can be used to select appropriate features or variables for BCI data analysis. In structured sparsity, not all sparse patterns are equally likely. For group sparsity, such as group Lasso [Friedman 2010], coefficients within the same group are more likely to be zeros or nonzero simultaneously. Such structured sparsity can be used to select group of variables or even channels in BCI data analysis [Huang 2009, Szafranski 2008].

Table 1.4: Popular basic regularizers for vector $\mathbf{w} \in \mathbb{R}^d$.

Regularizer	Defnination	Properties
ℓ_1 norm	$\Omega(\mathbf{w}) = \sum_{i=1}^{d} \lvert\mathbf{w}_i\rvert$	 non-differentiable and convex, used to measure sparsity of f.
ℓ_2 norm	$\Omega(\mathbf{w}) = \mathbf{w}^\top\mathbf{w}$	 differentiable smoothy and convex, the most commonly used norm.
ℓ_∞ norm	$\Omega(\mathbf{w}) = \max_{i=1,\cdots,d.}\{\lvert\mathbf{w}_i\rvert\}$	 non-differentiable and convex.
ℓ_p norm	$\Omega(\mathbf{w}) = (\sum_i \lvert\mathbf{w}_i\rvert^p)^{\frac{1}{p}}$	 when $0 < p \leq 1$, resulted in sparse and non-convex solution.
ℓ_0 pseudo norm	$\Omega(\mathbf{w}) = \lVert\mathbf{w}\rVert_0$	 this norm counts the number of active variables.

1.3.3 Family of hypothesis $\mathcal{H}$

Different choices of $\mathcal{H}$ are possible. In this part, we detail three common choices which involve mostly in current dissertation: linear model space, kernel space and deep architecture.

1.3.3.1 Linear models

Space of linear models is a basic structure in machine learning. It consists of a family of functions f such that the decision can be determined by a linear combination of inputs, that is,

$$\mathcal{H} = \{f | f(\mathbf{x}) = \langle \mathbf{w} \cdot \mathbf{x} \rangle + b = \mathbf{w}^\top\mathbf{x} + b\} \tag{1.7}$$

where $\mathbf{w} \in \mathbb{R}^d$ for $\mathbf{x} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ represents the bias term. They are probably the most popular learning algorithm in BCI applications [Lotte 2007a]. Two main kinds of linear representations have been used successfully in BCI data analysis, namely, Linear Discriminant Analysis (LDA) [Hastie 2009] and linear Support Vector Machines (SVM) [Vapnik 1995].

1.3.3.2 Kernel machines

Decision function (1.7) involves the inner product $\langle \mathbf{x}, \mathbf{z} \rangle = \mathbf{x}^\top \mathbf{z}$ which can be seen as a measure of similarity between the vectors $\mathbf{x}, \mathbf{z} \in \mathbb{R}^d$. Kernel machines generalize this setup by considering a powerful notion of similarity in more complex space other than $\mathbb{R}^d$. The essential tool is kernel function from which several important algorithms for pattern analysis were derived [Shawe-Taylor 2004]. In this part, we introduce several components of kernel machines.

Kernel function κ is a function defined from $\mathcal{X} \times \mathcal{X}$ onto $\mathbb{R}$, namely,

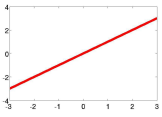
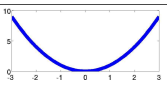
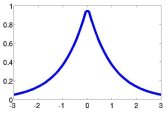
$$\begin{array}{rccc} \kappa: & \mathcal{X} \times \mathcal{X} & \longrightarrow & \mathbb{R} \\ & (\mathbf{x}_i, \mathbf{x}_j) & \longmapsto & \kappa(\mathbf{x}_i, \mathbf{x}_j). \end{array}$$

Intuitively, the "kernel" computes a similarity between two given samples. The kernel function κ is said to be *positive definite* if it fulfills the following condition for any finite positive integer n,

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j \kappa(\mathbf{x}_i, \mathbf{x}_j) \geq 0, \ \forall \{\alpha_i \in \mathbb{R}\}_{i=1}^{n}.$$

Some basic kernels and their examples are presented in Table 1.5. Let κ be a positive definite kernel on $\mathcal{X} \times \mathcal{X}$ and $\{\mathbf{x}_i \in \mathcal{X}\}_{i=1}^{n}$ be a sequence on $\mathcal{X}$. Gram matrix is a square matrix $\mathbf{K}$ of dimension n and of general term, $\mathbf{K}_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$. Matrix $\mathbf{K}$ is positive definite due to positive definiteness of $\mathbf{K}$.

Table 1.5: Some kernel functions.

Kernel	Description	$\kappa(\mathbf{x}_i, \cdot)$	
Linear	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j$		$\mathbf{x}_j = 1.$
Polynomial	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^\top \mathbf{x}_j)^r$		$\mathbf{x}_j = 1, \ r = 2.$
Gaussian	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$		$\mathbf{x}_j = 0, \ \sigma = 1.$
Laplacian	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{\sigma}\right)$		$\mathbf{x}_j = 0, \ \sigma = 1.$

From the previous definitions, and given a set of samples $\{\mathbf{x}_i\}_{i=1}^{n}$, one can define functions $f : \mathcal{X} \longmapsto \mathbb{R}$ as

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i \kappa(\mathbf{x}_i, \mathbf{x}) \quad \text{with } \alpha_i \in \mathbb{R}. \tag{1.8}$$

Let $\mathcal{H}$ be the space induced by these functions. $\mathcal{H}$ can be endowed with inner product defined as follows. Let $f, g \in \mathcal{H}$ with $g(\mathbf{x}) = \sum_{j=1}^{n} \beta_j \kappa(\mathbf{x}_j, \mathbf{x})$, to simplify the presentation. Thus the kernel inner product takes the bilinear form $\langle f, g \rangle_{\mathcal{H}} = \sum_i \sum_j \alpha_i \beta_j \kappa(\mathbf{x}_i, \mathbf{x}_j)$. Under mild conditions (see [Schölkopf 2002]), this inner product defines a norm $\|f\| = \sqrt{\langle f, f \rangle_{\mathcal{H}}}$ conferring a Hilbert space property to the closure of $\mathcal{H}$. This allows the introduction of reproducing property widely used in kernel machines.

Reproducing Kernel Hilbert Space (RKHS) The Hilbert space $\mathcal{H}$ of functions of f (defined over $\mathcal{H}$) endowed with the dot product $\langle f, g\rangle_{\mathcal{H}}$ is called a Reproducing Kernel Hilbert Space (RKHS) if there exists a kernel function with the following properties

- $\forall \mathbf{x} \in \mathcal{X}$, $\kappa(\mathbf{x}, \cdot)$ is a function of $\mathcal{H}$.
- $\forall \mathbf{x} \in \mathcal{X}$, $f(\mathbf{x}) = \langle f, \kappa(\mathbf{x}, \cdot)\rangle_{\mathcal{H}}$ (reproducing property).

Representer theorem Let $\mathcal{H}$ be a RKHS with kernel κ, V be a loss function from $\mathcal{X}$ to $\mathbb{R}^+$ and Φ a non-decreasing function from $\mathbb{R}^+$ to $\mathbb{R}$. If there exists a function f^* minimizing,

$$\min_{f \in \mathcal{H}} \sum_{i=1}^{n} V(f(\mathbf{x}_i), y_i) + \lambda \Phi(\|f\|_{\mathcal{H}}^2) \tag{1.9}$$

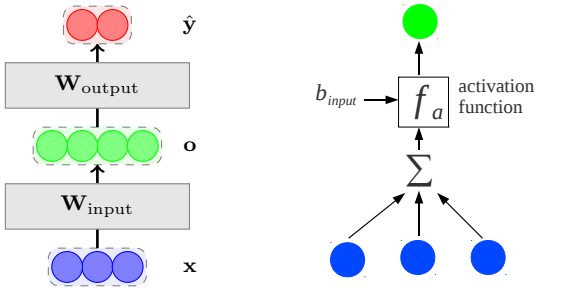
then there exists a vector $\boldsymbol{\alpha} \in \mathbb{R}^n$ such that,

$$f^*(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i \kappa(\mathbf{x}, \mathbf{x}_i) \tag{1.10}$$

that is, the optimal function f^* is a linear combination of (a finite set of) functions given by the data. For a detailed proof of this theorem, we refer the reader to [Schölkopf 2002]. Such model is linear in its parameters α_i but corresponds to a non-linear model in the input space. Algorithms capable of operating with kernels include SVM, Gaussian process, LDA and PCA [Bottou 2007, Yang 2005, Schölkopf 1997].

1.3.3.3 Mutilayer Perceptron (MLP)

MLPs are representative feedforward structures in artificial neural network family. MLP consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one [Bishop 1996]. Figure 1.6 presents a toy example which contains only one hidden layer. f_a is the activation function



(a) A 3-layer MLP. (b) Operation inter-layer.

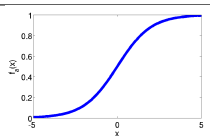
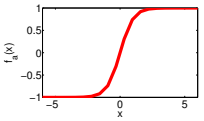
Figure 1.6: Illustration of MLP: 1.6(a) is composed of an input layer with 3 neurons, $\mathbf{x} \in \mathbb{R}^3$, one hidden layer (several ones in general case) with 4 neurons, $\mathbf{o} \in \mathbb{R}^4$ and the output layer with 2 neurons, $\hat{\mathbf{y}} \in \mathbb{R}^2$. $\mathbf{W}_{\text{input}}$ denotes the weight matrix from input layer to hidden layer and $\mathbf{W}_{\text{output}}$ denotes the output weight matrix. 1.6(b) illustrates the operation for each neuron. Its input is connected with the output of previous layer's neurons. The output is attained by the transformation of activation function.

which determines the mapping strategy from input of a neuron to its output. Table 1.6 shows some activation functions in practice. For a MLP with h hidden layers, layer k computes an output vector $\mathbf{o}_k$ using the output of previous layer $\mathbf{o}_{k-1}$ starting with the input $\mathbf{o}_0 = \mathbf{x}$,

$$\mathbf{o}_k = f_a^k(\mathbf{W}_k \mathbf{o}_{k-1} + b_k), \ \forall k = 1, \cdots, h. \tag{1.11}$$

where $\mathbf{W}_k$ denotes the weight matrix from the $k-1^{th}$ to k^{th} hidden layer, $\mathbf{o}_k$ denotes output of the k^{th} hidden layer and f_a^k is taken pointwisely if $\mathbf{W}_k\mathbf{o}_{k-1}+b_k$ is a vector. The top output hidden layer $\mathbf{o}_h$ is then used for making a prediction. This layers stacking lead to a deep architecture (compare to the shallow architecture (1.8) of kernel machines) with a powerful capacity of expression.

Table 1.6: Some activation functions in neural network family.

Activation function	Description	Applications
Logistic	$f_a(\mathbf{x}) = \text{sigm}(\mathbf{x}) = \frac{1}{1+\exp(-\mathbf{x})}$ 	regression/ranking /classification (when label $\{0,1\}$)
Tanh	$f_a(\mathbf{x}) = \tanh(\mathbf{x}) = \frac{\exp(\mathbf{x})-\exp(-\mathbf{x})}{\exp(\mathbf{x})+\exp(-\mathbf{x})}$ 	regression/ranking /classification (when label $\{+1,-1\}$)
Softmax	$f_a(\mathbf{x})_i = \text{softmax}(\mathbf{x}) = \frac{\exp(y_i)}{\sum_{k=1}^{Ny}\exp(y_k)}$ Ny: the number of neurons in the output layer	ranking/classification

1.3.4 Associative learning algorithms

We present several popular algorithms and illustrate their characteristics in this section. Firstly, we present SVM as an important case in BCI domain. Secondly, we introduce a deep learning strategy for the MLPs with many hidden layers. Finally, we take Semi-supervised SVM as an instance for solving a specific problem, that is, only little labeled samples available and with (large) amount of unlabeled samples in some BCI applications.

1.3.4.1 Support Vector Machine (SVM)

SVM-large margin based model SVM for binary classification is an margin-based learning machine which aims at maximizing the margin between two classes. In this part, we consider a data set $\mathcal{D} = \{(\mathbf{x}_i, y_i) \in \mathcal{X}\times\mathcal{Y}\}_{i=1}^n$ with $\mathcal{Y} = \{-1, 1\}$ and present the SVM problem hereafter.

Primal optimization Defining $f(\mathbf{x}) = f_0(\mathbf{x}) + b$ with $f_0 \in \mathcal{H}$ a RKHS induced by a kernel κ and $\|f\|_{\mathcal{H}} = \|f_0\|_{\mathcal{H}}$, the primal problem of SVM takes the form,

$$\min_f \ \frac{1}{2}\|f\|_{\mathcal{H}}^2 + C\sum_{i=1}^n V(y_i, f(\mathbf{x}_i)) \tag{1.12}$$

with $V(f(\mathbf{x}_i), y_i) = \max(0, 1-y_i f(\mathbf{x}_i))$ chosen as the Hinge-loss function. Obviously this problem can be recast as (1.5) with the regularization parameter $\lambda = \frac{1}{C}$. By rephrasing the hinge loss based empirical error, one gets the usual and well known SVM formulation [Schölkopf 2002, Vapnik 1995],

$$\begin{aligned} \min_{f,\boldsymbol{\xi}} \quad & \tfrac{1}{2}\|f\|_{\mathcal{H}}^2 + C\textstyle\sum_{i=1}^n \xi_i \\ \text{subject to} \quad & y_i f(\mathbf{x}_i) \geq 1-\xi_i, \qquad \forall i = 1,\cdots,n \\ & \xi_i \geq 0, \qquad \forall i = 1,\cdots,n \end{aligned} \tag{1.13}$$

where ξ_i is a slack variable that measures the misclassification degree of the datum $\mathbf{x}_i$.

Dual problem Introducing Lagrangian of the previous problem and taking into the optimality conditions according to f_0 and b, one attains the dual problem,

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \tfrac{1}{2}\boldsymbol{\alpha}^\top \mathbf{H}\boldsymbol{\alpha} - \mathbf{e}^\top \boldsymbol{\alpha} \\ \text{subject to} \quad & \boldsymbol{\alpha}^\top \mathbf{y} = 0 \\ & 0 \leq \alpha_i \leq C \qquad \forall i = 1, \cdots, n \end{aligned} \tag{1.14}$$

with $\boldsymbol{\alpha} \in \mathbb{R}^n$ the vector of Lagrange parameters, $\mathbf{y} = \begin{bmatrix} y_1 & \dots & y_n \end{bmatrix}^\top$, $\mathbf{H} \in \mathbb{R}^{n \times n}$ the matrix with entries $\mathbf{H}_{ij} = y_i y_j \kappa(\mathbf{x}_i, \mathbf{x}_j)$, $i, j = 1, \cdots, n$ and $\mathbf{e} \in \mathbb{R}^n$ a vector of ones.

The solutions of the primal and dual problems satisfy regularity conditions termed as **Karush-Kuhn-Tucker (KKT) conditions**. These conditions are read as:

- Stationarity:

$$\begin{aligned} f_0(\mathbf{x}) - \textstyle\sum_{i=1}^n \alpha_i y_i \kappa(\mathbf{x}_i, \mathbf{x}) &= 0 \\ \text{and} \qquad \textstyle\sum_{i=1}^n \alpha_i y_i &= 0 \\ C - \alpha_i - \beta_i &= 0 \quad \forall i = 1, \cdots, n \end{aligned}$$

- Primal feasibility:

$$\begin{aligned} y_i f(\mathbf{x}_i) &\geq 1 - \xi_i, \quad \forall i = 1, \cdots, n \\ \text{and} \qquad \xi_i &\geq 0, \quad \forall i = 1, \cdots, n \end{aligned}$$

- Complementary slackness:

$$\begin{aligned} \alpha_i (y_i f(\mathbf{x}_i) - 1 + \xi_i) &= 0 \quad \forall i = 1, \cdots, n \\ \text{and} \qquad \beta_i \xi_i &= 0 \quad \forall i = 1, \cdots, n \end{aligned}$$

- Dual feasibility:

$$\begin{aligned} 0 \leq \alpha_i &\leq C, \quad \forall i = 1, \cdots, n \\ \text{and} \quad 0 \leq \beta_i &\leq C, \quad \forall i = 1, \cdots, n \end{aligned}$$

The associated lagrangian function of Problem (1.13) is

$$L(f, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\|f\|_{\mathcal{H}}^2 + C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i \big(y_i f(\mathbf{x}_i) - 1 + \xi_i\big) - \sum_{i=1}^n \beta_i \xi_i$$

Note that the first stationarity condition is obtained by computing the differential of the Lagrangian according to f_0 and using the reproducing property $\langle f_0, \kappa(\mathbf{x}, \cdot) \rangle = f_0(\mathbf{x})$.

Finally, the desired decision function is taken as $\text{sign}(f(\mathbf{x}))$.

1.3.4.2 Deep learning

Similar to kernel machines, a MLP can be obtained by optimizing a penalized empirical risk $\mathsf{R}^{\text{emp}}(f) + \lambda\Omega(f)$ where f is derived as

$$f(\mathbf{x}) = g^h \circ g^{h-1} \circ \cdots \circ g^1(\mathbf{x})$$

with $g^k(\mathbf{o}_{k-1}) = f_a^k(\mathbf{W}_k \mathbf{o}_{k-1} + b_k)$. The function realizes the nonlinear mapping between a hidden layer input $\mathbf{o}_{k-1}$ and its output $\mathbf{o}_k = g^k(\mathbf{o}_{k-1})$. The high non-linearity of f does not allow the use of batch optimization algorithm as the parameters $\mathbf{W}_k$, b_k, $k = 1, \cdots, h.$ involved in the network are of high dimension. The remedy is to train the network using gradient method [Bottou 2004] leading to the popular gradient Backpropagation (BP) algorithm. Unfortunately, it is not effective when the error information must be propagated across multiple non-linearities.

General idea of deep architecture is to pretrain the hidden layers in a greedy way followed by fine tuning of the overall network. Pretraining stage acts as a refined way of initializing the weights $\mathbf{W}_k$, b_k [Hinton 2006]. Many of the deep architectures are based on the autoassociators or restricted Boltzmann machines. They can be categorized as two groups from perspective of pretraining strategy, namely, unsupervised pretraining and supervised pretraining.

Unsupervised pretraining As shown in Figure 1.7, autoassociator is a simple unsupervised algorithm for learning a one-layer model that computes a representation for its input. Because training an autoassociator seems easier than training a deep network, they have been used as building blocks to train deep networks, where each level is associated with an autoassociator that can be trained separately.

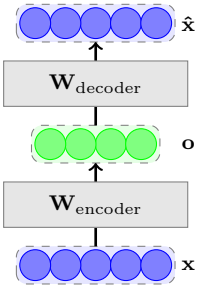
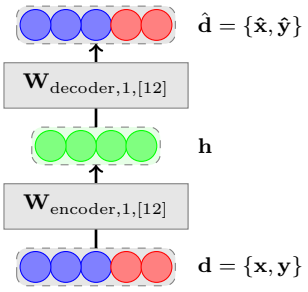


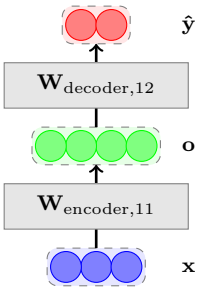
Figure 1.7: Illustration of the autoassociator with encoder/decoder architecture: the green units are hidden units and the blue ones are input units. The encoder maps the inputs $\mathbf{x}$ to a hidden layer $\mathbf{o}$, and the decoder reconstructs the input $\hat{\mathbf{x}}$ from the hidden layer. $\mathbf{W}_{\text{encoder}}$ and $\mathbf{W}_{\text{decoder}}$ denote the weight matrix for encoder and decoder seperately.

A typical training procedure for the deep multi-layer neural networks can be summarized as follows: (1) pretrain the weights of each layer individually by an autoassociator in a greedy layer-wise way, that is, the previous hidden units' outputs are then used as input for another layer. (2) Take the last hidden layer output as input to a supervised layer and initialize its parameters. (3) Fine-tune all the parameters of the deep architecture with respect to supervised criterion [Bengio 2009]. Another typical framework involved stacked restricted Boltzmann machines, [Hinton 2006] initialized the weights of each layer individually in a purely unsupervised way and fine-tunes the entire network using labeled data. Note that the above algorithms only involve the inputs in the pretraining procedure.

Supervised pretraining Supervised pretraining Deep Neural Network (DNN) can also be realized with autoassociators. One autoassociator provides initial weight for one layer. As a greedy strategy, outputs of the former autoassociator will be selected as inputs of the following one. Providing an appropriate



(a) Autoassociator used for supervised pretraining.



(b) Initial model of DNN.

Figure 1.8: A toy example of a supervised pretraining strategy. 1.8(a): an autoassociator learned on input $\mathbf{d} = \{\mathbf{x}, \mathbf{y}\}$. 1.8(b): the initial last hidden layer and the initial output layer ($\mathbf{W}_{\text{encoder},11}$ is the part of $\mathbf{W}_{\text{encoder}}$ that corresponding to input $\mathbf{x}$, $\mathbf{W}_{\text{decoder},12}$ is the part of $\mathbf{W}_{\text{decoder}}$that corresponding to output $\mathbf{y}$).

initial mapping from the last hidden layer to the output layer is essential for the supervised pretraining.

Figure 1.8 illustrates one supervised pretraining strategy: (1) A new data $\mathbf{d} = \{\mathbf{x}, \mathbf{y}\}$ is composed to introduce the label information in pretraining procedure. (2) An autoassociator with $\mathbf{W}_{\text{encoder},1,[12]}$ and $\mathbf{W}_{\text{decoder},1,[12]}$ was trained according to the new data set $\mathbf{d}$. (3) To initialize the DNN, $\mathbf{W}_{\text{encoder},11}$ and $\mathbf{W}_{\text{decoder},12}$ were set as the initial weight matrix for the DNN, and a fine tuning is followed finally. We have shown the effectiveness of this kind of pretraining strategy that exploits the label information in [Tian 2010]. Other types of supervised pretraining strategy could also be found in the same paper. As the main machine learning algorithms involved in this thesis are kernel machines, we won't present detail of this part of work in the following dissertation.

1.3.4.3 Semi-supervised SVMs

Semi-supervised SVMs aim at learning an SVM that exploits the information conveyed by the unlabeled data. As shown in figure 1.9, the general picture is to determine a decision function able to classify the labeled data and to correctly predict the class of unlabeled samples while separating the two classes.

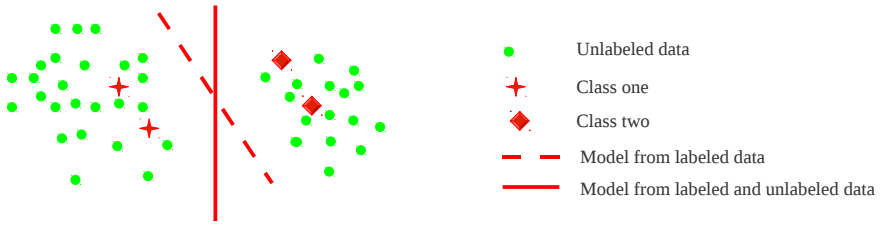


Figure 1.9: Illustration of Semi-supervised learning.

Generally speaking, semi-supervised SVM algorithms rely on the following optimization problem,

$$\min_{f,\mathbf{y}_u} \Omega(f) + C\sum_{i=1}^{\ell} V(f(\mathbf{x}_i), y_i) + C^* \sum_{i=\ell+1}^{n} V(f(\mathbf{x}_{ui}), y_{ui})) \tag{1.15}$$

where the decision function is defined as $f(\mathbf{x}) = f_0(\mathbf{x}) + b$ with f_0, a function in a RKHS $\mathcal{H}$ and b a real scalar, $\mathbf{x}_{ui}$ denotes the unlabeled data, y_{ui} is the unknown label to be estimated, $\mathbf{y}_u = \begin{bmatrix} y_{\ell+1} & \dots & y_n \end{bmatrix}^\top$. The first term in (1.15) represents the regularization term. The last two terms are respectively the fitting errors for the labeled and unlabeled samples which are evaluated through the margin loss functions. The regularization parameters C and C^* balance the importance of those errors.

From this general problem, two main families of learning problems were derived based on particular assumptions beneath the marginal distribution $\mathbb{P}(X)$ of the data. The cluster assumption believes that two points that belong to the same cluster (that is points connected via high-density paths) likely share the same label. Therefore it promotes decision function avoiding high density regions and led to Transductive SVM (TSVM) [Vapnik 1977, Bennett 1998]. Manifold assumption assumes that points lie on a low dimensional manifold embedded in a higher dimensional space, and it gives rise to Laplacian SVM [Belkin 2006]. In practice, it is unclear whether and when one assumption should be preferred over the other. Nevertheless, empirical evidences have shown that the choice is application dependent and it was recommended to dig the combination of both assumptions in a single framework [Chapelle 2008b]. Our conducted research has shown that the semi-supervised SVMs based on the two assumptions could be a good choice in BCI domain [Tian 2012].

1.3.5 Summary

In this section, we present the general framework and backgrounds for machine learning. Several learning algorithms involved in current dissertation were also introduced. According to the basis, we adopted kernel machines as the leading tool in this PhD study on both supervised and semi-supervised learning framework.

1.4 Current limitations and challenges

The development of reliable and practical BCI systems shall deliver high-speed communication and control. Such a successful system has many potential applications, especially for patients who are paralyzed. Most BCI systems are tested in laboratory environments. Even for the groups that have realized assistant device control, many of them remain at the demonstration stage in laboratories. In the subsequent subsections, we briefly shed some lights on the reasons of current limitations.

1.4.1 Sensory interfacing problem

To date there is no sensory modality that is accurate and safe. Invasive sensory interfaces achieve the most accurate BCIs. While EEG signals will gradually deteriorate with time in practice [Geddes 2003]. EEG sensors have the best safety and lowest invasiveness but have the poorest accuracy.

According to the international assessment report [Berger 2007], several major questions need to be addressed for the development of both invasive and non-invasive sensors. These are as follows:

- How long do current sensors really last?
- How to develop sensors that last for five or twenty years?
- How to make dry EEG electrodes that allow for ease of application and use?
- How to develop a system and scientific approach to designing "biologically-based" implanted microelectrodes and surface electrodes?

New BCI research proposed some solutions of these questions. For example, novel dry EEG recording technologies were proposed recently [Popdscu 2007, Zander 2011]. Their investigated electrodes provide a potential to be applicable in BCI applications, while further research need to be implemented to evaluate the dry electrode sensory.

1.4.2 Limited knowledge of neuromechanism

One important challenge lies in the brain itself. BCI control does not work for a portion of users (estimated 15% to 30%). Such phenomenon is called "BCI illiteracy" [Blankertz 2009]. It is still an open question that why BCI systems exhibit illiteracy in a significant minority of subjects and what can be done about it.

On the other hand, most of the existing research has focused on cue-based BCIs, where the mental states are more or less well defined. Self-pace BCIs, where a number of distinct patterns have to be reliably detected in ongoing brain activities, offer more natural human-machine interaction [Tsui 2009]. But it is a great challenge to train and adapt a self-paced BCI online because the user's control intention (or mental state) and timing are usually unknown. Continuous identification/evolution of the mental state helps the adaptability of the user to the interface and vice versa [Mourino 2002]. Utilization of the user's emotional state to adapt the BCI classification algorithms is one possible solution to realize such self-paced BCIs[9]. Affective BCIs were proposed recently, they are systems that measure signals from the peripheral and central nervous system, extract features related to affective states of the user and use these features to adapt human-computer interaction [Nijboer 2009].

The inextricable link between emotions and cognition inspires researchers in human computer interface to include emotional states in the design of new interfaces: either as evaluation indicators or as components to be inserted in the interface loop. Utilization of the users' emotional state to adapt the BCI classification algorithms is a new trend in BCI research domain.

The brain itself is a highly adaptive device. One problem confronting BCI designers is that: how much of the learning should be relegated to the machine and how much should be left to the brain. This is also called the "brain-computer co-adaptation" problem. Current approaches usually rely on both user

[9] http://emotion-research.net

adaptation and machine learning strategies [Millan 2007, Buttfield 2006, Blumberg 2007]. Simultaneous online adaptation by the user and the BCI remains a topic of active research.

1.4.3 Signal processing issues

The signal processing scheme is an essential component in the design of a successful BCI system. It translates signals produced by the brain into useful device commands. Following [Krusienski 2011], critical questions existing in current BCI research could be categorized as the following types:

- *What promising feature extraction and translation techniques deserve more attention?* Generally, a machine learning based BCI design consists of two phases: (1) a time-consuming preparation stage which is known as the calibration of BCI classification schemes. This calibration phase normally involves the acquisition of training data (including data preprocessing and feature extraction stage) and model optimization process; (2) use phase where the obtained models are used to operate a BCI. Such design results in the following challenges for a promising signal processing system: reduce the calibration procedure and improve the classification accuracy.

- *How can feature and classifier adaptation be used to cope with signal non-stationarities and aid user training?* Current BCI research suffers two main problems in this field: the curse-of-dimensionality and the bias variance trade-off. For general classification task, the amount of data needed to properly describe the different classes increases exponentially with the dimensionality of feature vectors [Friedman 1997]. Unfortunately this cannot be applied in all BCI systems as generally, the dimensionality is high and the training set small. This "course" is a major concern in BCI design, as EEG signals are non-stationary, training sets coming from different sessions are likely to be relatively different. This problem is also known as covariate shift problem [Krusienski 2011].

1.4.4 Summary

In this section, we summarize the limitations and challenges in existing BCI research from three aspects: (1) sensory interfacing problem; (2) limited knowledge of neuromechanism; (3) signal processing challenges. These limitations and challenges partly guide the current dissertation and we explore the solutions of few of them in the following section.

1.5 Some solutions: from challenges to current PhD study

In this dissertation, we try to pursuit and address some aspects of the last two challenges existed in BCI systems. The main part of the work can be divided into two categories according to their objective solutions:

- The first category is to explore the neuromechanism of brain. We designed an SSVEP based BCI system. Such system introduces the emotional state of the user into the BCI operation loop, corresponding operating protocol will be adjusted with the change of emotional states. Concretely, we investigate whether or not emotion has an influence in learned model.

- The second category is focused on the signal processing issues and mainly aims at reducing the calibration procedure and to improve the classification performance in the use phase. To reduce the calibration procedure, we propose a new inductive semi-supervised learning algorithm; to improve the classification performance, we propose an online multi-kernel algorithm named LaMKL. We will elaborate on the details of LaMKL in Chapter 3 to achieve automatic feature selection in online way.

1.5.1 A multi-kernel framework for inductive semi-supervised learning

In Chapter 2, we present an inductive semi-supervised learning framework named TSVM-MKL. This is a multiple kernel version of TSVM. The motivations and implementations of this algorithm are illustrated in what follows:

- When applying machine learning approaches to BCIs, one needs labeled data to teach the classifier. To this end, the user usually performs a tedious calibration measurement before starting with BCI feedback applications. To reduce this process, a solution consists of employing semi-supervised learning which utilize large amount of unlabeled data [Qin 2007]. Therefore, a few labeled data along with unlabeled EEG signals are needed in the beginning of acquisition session to design a good learner.
- As shown in Section 1.3.4.3, most of the semi-supervised learning algorithms make strong model assumptions to deal with limited labeled data and available unlabeled data. For specific applications such as BCI data analysis, it is unclear whether and when one assumption should be preferred over the other. Ideally, the learning algorithm shall determine the type of assumption and/or what extent of the adopted assumption will be effected automatically. To this end, we employ the multiple kernel learning (MKL) [Rakotomamonjy 2008a, Bach 2004] to fuse the two popular assumptions into one learning framework.

The proposed TSVM-MKL aims at digging the combination of both assumptions in a single framework in order to expect beneficial effect in terms of classification performances. Promising results on benchmark data sets and the BCI data analysis suggest and support the effectiveness of proposed work. As a bonus, the proposed TSVM-MKL algorithms also have a satisfying application in the feature and/or channel selection in BCI domain.

1.5.2 An online multiple kernel learning: LaMKL

BCI systems require adaptation of classifiers through time because of non-stationarity of EEG data. In Chapter 3, we propose a way to adapt in online fashion that the classifier based on supervised and non-sparse multiple kernel learning. The motivations and realizations are as follows:

- The MKL framework is used to combine in an automatic way the features (and/or channels) of a BCI processing block. This combination is expected to vary along time and is tracked by adapting the classifier for every seen sample.
- The implemented online learning operates on the dual problem of a ℓ_p-norm ($p > 1$) MKL problem. Exploiting the dual proposed in [Vishwanathan 2010], we have designed an algorithm which dynamically improves current classifier following an idea pursued by [Bordes 2005] who had developed an effective online SVM for a single kernel.

Finally, we attain an online LaMKL algorithm that achieves similar performance with the batch MKLs while requiring less computation time.

1.5.3 Improving BCI performance beyond machine learning algorithms

In this part, we explore to improve BCI performance beyond developing new machine learning algorithms. For this sake, we first confirm the feasibility of employing simple classifier with careful model/feature selection in BCI system by a BCI competition data analysis: "Mind reading, MLSP competition 2010". We then designed an affective SSVEP based BCI system. Roughly, induce the predefined emotional states (negative, positive and neutral emotional state) during the experiments and recording the EEGs corresponding to the BCI tasks. Such EEGs can be regarded as the EEGs with emotions. And the following data analysis is implemented based on those emotional EEG data. We employed two different statistical tests and a unify conclusion can be attained as follows: (1) emotion does affect the BCI performance; (2) A user with neutral and positive emotional states perform better than one with negative emotion.

1.5.4 Summary

In this section, some solutions of current challenges in this dissertation were summarized from two different perspectives: (1) from the view of machine learning algorithms, we mainly employ the multiple kernel learning. (2) Beyond developing new machine learning algorithms, we investigate the feasibility of improving BCI performance from careful model/feature selection or taking account for the emotion in BCI in Chapter 4.

1.6 Conclusions

In this chapter, we review the BCI system, as well as the machine learning algorithms for BCI data analysis. For this sake, we first present BCI components and then emphasize the EEG-based BCI paradigms that mostly involved in this dissertation. In order to derive current PhD study, we analyzed the limitations and challenges in existing BCI research. The main tasks of this dissertation and their implementations were thus derived.

The main goal of this research is to improve the BCI performance from machine learning pespective. In Chapter 2, an effective multiple kernel learning framework for inductive semi-supervised learning is proposed to reduce the calibration procedure in BCI system. In Chaper 3, we implement multiple kernel learning algorithm which named as LaMKL to profit the advantages of online learning and multiple kernel learning. Effectiveness have been shown by the experimental results. Except the exploration from the view of machine learning, we also implement other strategies to improve the BCI performance in Chapter 4. After confirming that simple classifier model can also achieve satisfying performance with careful model/feature selection operation, we design an emotional SSVEP-based BCI system. Experimental results demonstrate that emotion does affect the BCI performance and provides the feasibility of adopting user's emotion to adapt the classifier model in real time BCI operation.

Chapter 2

Semi-supervised learning in BCI

Contents

In many applications such as text classification, image categorization, spam detection or BCI, data labelling can be costly, time consuming or inappropriate. Take the EEG signals as example, they are naturally non-stationary, noisy and different from subject to subject and context. To achieve a satisfying BCI system, one needs to search for suitable machine learning algorithms to fit the specific characteristics of the user's brain signals. However, acquisition of labeled data to guide the design of suitable classifier can be cumbersome for the user due to the long time for BCI system calibration and also because operation of BCI system can be exhausting. In that situation the recourse to a learning methodology combining labeled data and unlabeled samples, that is semi-supervised learning can be helpful.

The importance gained by semi-supervised learning (SSL) these past years in machine learning is due to the difficulty to label the increasing size data sets in order to apply well-established supervised algorithms. The aim of SSL is to predict unknown labels by exploiting altogether available labeled samples and (statistical or geometrical) information conveyed by unlabeled data. Applied to BCI applications, existing SSL algorithms achieve satisfying solutions for such BCI systems, including self-training algorithm [Qin 2007, Li 2008], co-training algorithm [Panicker 2010], transductive SVM [Liao 2007], graph-based methods [Zhong 2009] and so forth. All these algorithms made strong model assumptions to deal with limited labeled data and available amount of unlabeled data. Common hypothesis are cluster assumption and manifold assumption. The first assumption aims to enforce two training points (labeled or not) that fall in the same cluster to share the same label. The resulting algorithms prefer decision function avoiding high density regions [Vapnik 1977, Chapelle 2005, Joachims 1999]. The second assumption rather promotes data geometry to enforce smoothness of the labels prediction over manifolds using similarity graph-based methods [Zhu 2002, Joachims 2003, Belkin 2006].

In practice, it is unclear whether and when one assumption should be preferred over the other. Bad matching of problem structure with model assumption can lead to degradation in classifier performance. Nevertheless, empirical evidences have shown that the choice is application dependent and it was recommended in [Chapelle 2008b] to dig the combination of both assumptions in a single framework in order to expect beneficial effect in terms of classification performances. To reach this goal, some approaches were proposed in the literature. [Mallapragada 2009] presented a boosting framework for semi-supervised learning to exploit both manifold and cluster assumptions; [Goldberg 2009] employed a "cluster-then-label" strategy for the data consists of multiple interesting manifolds; [Dai 2007] proposed a regularization framework for semi-supervised learning machines by integrating the two assumptions. The limitations of these models always reside in the high computation complexity or the transductive nature. By transductive nature, we mean algorithms which rather focus on the prediction of unknown labels given a training set composed of labeled and unlabeled data. The resulting decision functions are limited as they cannot deal with out-of-samples. The approach of solution we pursue in the current chapter consists in the design of inductive classifiers able to cope with unlabeled training data as well as unseen test data.

More specifically, our approach of solution takes place in large margin and kernel framework and is based on Support Vector Machines (SVM). Indeed, inductive classifiers are attainable with SVM and the flexibility of kernel machines allows the choice of appropriate kernels via multiple kernel learning [Rakotomamonjy 2008a]. To embed the cluster view and the manifold view in a single framework, we consider a multiple kernel version of Transductive SVM (TSVM). Even though the appellation "TSVM" refers to transductive nature, the obtained classifier is inductive (see [Chapelle 2006, chapter 25]). Moreover, TSVM somehow implements the cluster assumption. Therefore, in the adopted strategy, we consider a pool of kernels, some implementing similarity graph constraints or different a priori informations and we design an efficient learning algorithm based on previous supervised multiple kernel learning [Rakotomamonjy 2008a] to select the kernels suited for our semi-supervised application.

This leads us to formulate a multiple kernel TSVM which inherits the non-convexity (and non-smoothness) of TSVM. The optimization algorithm we propose comes with the usual caveats of non-convex problems. It is built upon DC (difference of convex functions) algorithm [Tao 1998] and is able to find in an efficient way a local solution. As a solution, we get an inductive classifier extendable to unseen samples and thereby alleviate the drawbacks of method in [Dai 2007].

This chapter is organized as follows. In Section 2.1, we first review background of semi-supervised learning. In Section 2.2, we formally present the multi-kernel framework of TSVM, which combines the cluster assumption and manifold assumption in one learning task. And then derive the optimization algorithm used to solve the problem in Section 2.3. Connections of our approach to related algorithms are detailed in Section 2.4. We then report the experimental results on a series of benchmark data sets and demonstrate the effectiveness of our algorithm in Section 2.5. Applications of the proposed TSVM-MKL algorithm are presented in Section 2.6 and we end this chapter with some conclusions in Section 2.7.

2.1 Semi-supervised SVMs

Let $\mathcal{D} = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_\ell, y_\ell), \mathbf{x}_{\ell+1}, \cdots, \mathbf{x}_{\ell+u}\}$ denote the entire dataset for simplicity. Without loss of generality, we assume the first ℓ samples are labeled $\{(\mathbf{x}_i, y_i) \in \mathcal{X} \times \{-1, 1\}\}_{i=1}^{\ell}$ and are followed by u unlabeled samples $\{\mathbf{x}_i\}_{i=\ell+1}^{\ell+u}$. The unknown labels are binary entries of the vector $\mathbf{y}_u = [y_{\ell+1} \dots y_{\ell+u}]^\top$. The goal of semi-supervised learning is to predict correctly $\mathbf{y}_u$ based on the training data $\mathcal{D}$.

2.1.1 Problem setting: preliminaries

Let $g(\mathbf{x})$ be the decision function. Semi-supervised learning approaches under SVM framework we will consider in this chapter rely on the following optimization problem

$$\min_{f,b,\mathbf{y}_u} \Omega(f) + C\sum_{i=1}^{\ell} V(g(\mathbf{x}_i), y_i) + C^* \sum_{i=\ell+1}^{\ell+u} U(g(\mathbf{x}_i), y_i) \tag{2.1}$$

where the decision function is defined as $g(\mathbf{x}) = f(\mathbf{x}) + b$ with f, a function in a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$ and b a real scalar. The two last terms are respectively the fitting errors for the labeled and unlabeled samples which are evaluated through the margin loss functions V (labeled data) and U (unlabeled data). The regularization parameters C and C^* balance the importance of those errors in the optimization process. From this general problem, two main families of learning problems were derived based on particular assumptions beneath the marginal distribution $\mathbb{P}(\mathbf{x})$ of the data, namely the cluster assumption which has led to TSVM [Vapnik 1977] and the manifold assumption giving rise to Laplacian SVM [Belkin 2006]. The formulations of these methods are described below.

2.1.2 Transductive SVM

Transductive SVM implements the first strategy termed as cluster assumption [Seeger 2002] which postulates that two points that belong to the same cluster (that is points connected via high-density paths) likely share the same label. Therefore it promotes decision function avoiding high density regions. In its first version, TSVM attempts to solve the following problem [Joachims 1999, Chapelle 2008b]

$$\min_{f,b,\mathbf{y}_u} \frac{1}{2}\|f\|^2_{\mathcal{H}} + C\sum_{i=1}^{\ell} V(g(\mathbf{x}_i), y_i) + C^* \sum_{i=\ell+1}^{\ell+u} U(g(\mathbf{x}_i), y_i) \tag{2.2}$$

where V and U employ the same margin loss, e.g. hinge loss or its square version:

$$V(g(\mathbf{x}), y) = H_s(yg(\mathbf{x}))^q \quad \text{with} \quad q \in \{1, 2\} \quad \text{and} \tag{2.3a}$$

$$H_s(z) = \max(0, s - z), \quad 0 \leq s \leq 1. \tag{2.3b}$$

Notice here that contrary to Chapter 1, we will adopt for the hinge loss functions the notation $H_s(yg(\mathbf{x}))$ instead of $H_s(g(\mathbf{x}), y)$ in order to ease the reading.

To avoid the trivial solution of problem (2.2) where the unlabeled data are all assigned to the same class, a balancing constraint is added to the problem

$$\frac{1}{u}\sum_{i=\ell+1}^{\ell+u} \max(0, y_i) = r. \tag{2.4}$$

This constraint enforces a chosen proportion r of unlabeled samples in the positive class. The choice of r is user-dependent and can rely on the putative proportion of positive samples in the labeled data. Problem (2.2) presents a cumbersome aspect: the optimization is carried over the unknown and discrete labels $\mathbf{y}_u$ and continous variables (f, b) rendering the standard optimization methods [Nocedal 2006] inapplicable.

A review and comparison of algorithms to address this problem is exposed in [Chapelle 2008b]. Roughly speaking, the existing approaches can be divided in two categories: the first category includes combinatorial techniques which attempt to solve directly problem (2.2) while the second category transforms the original problem in order to eliminate the unknown labels $\mathbf{y}_u$. A brief description of these methods is presented hereafter.

2.1.2.1 Combinatorial methods

Their finality is oriented toward a transductive learner. Among existing scalable approaches, one can point out S3VM$^{\text{light}}$ [Joachims 1999], a well-known software. It is based on labels-switching-model retraining

procedure to find a local minimum of the optimization problem. To get rid of the discrete labels, a relaxation is possible: the labels $\mathbf{y}_u$ are replaced with a continuous vector $\mathbf{p}$ with entries $p_i = \mathbb{P}(y_i = 1)$ traducing the probability to assign $\mathbf{x}_i$ to the positive class. Then the objective function of the problem reads [Sindhwani 2006, Wang 2009]:

$$J(f, b, \mathbf{p}) = \frac{1}{2}\|f\|_{\mathcal{H}}^2 + C\sum_{i=1}^{\ell} V(y_i g(\mathbf{x}_i)) + C^* \sum_{i=\ell+1}^{\ell+u} [p_i U(g(\mathbf{x}_i), 1) + (1-p_i)U(g(\mathbf{x}_i), -1)]$$

Sindhwani et al. [Sindhwani 2006] have proposed to solve this new problem via determinist annealing (DA) method. For this sake a regularizing entropy term on $\mathbf{p}$ is included in the process. Also, an adaptation of the balancing constraint (2.4) is adopted leading finally to the problem [Sindhwani 2006]:

$$\min_{(f,b),\mathbf{p}} \quad J(f, b, \mathbf{p}) - T\sum_{i=\ell+1}^{\ell+u} [p_i \log(p_i) + (1-p_i)\log(1-p_i)] \tag{2.5a}$$

$$\text{s.t.} \quad \frac{1}{u}\sum_{i=\ell+1}^{\ell+u} p_i = r$$

with $T \geq 0$. DA approach starts from an "easy" problem, and gradually deforms it to the objective function (2.5) as in continuation methods. It is guaranted to converge toward a local solution.

2.1.2.2 Continuous methods

These techniques do not focus on unknown labels estimation but rather seek an inductive semi-supervised classifier. Indeed, problem (2.2) can be seen as

$$\min_{f,b} \frac{1}{2}\|f\|_{\mathcal{H}}^2 + C\sum_{i=1}^{\ell} V(g(\mathbf{x}_i), y_i) + C^* \sum_{i=\ell+1}^{\ell+u} U(|g(\mathbf{x}_i)|) \tag{2.6}$$

Here also we slightly abuse notation by writing $U(|g(\mathbf{x}_i)|)$ instead of $U(g(\mathbf{x}_i), \text{sign}\,(g(\mathbf{x}_i)))$ that is one considers the unknown label to be either -1 or 1. While it solely involves continuous unknowns, this problem is highly non-convex as the loss function $U(|z|)$ is non-convex and non-smooth. This fact is illustrated in figure 2.1.

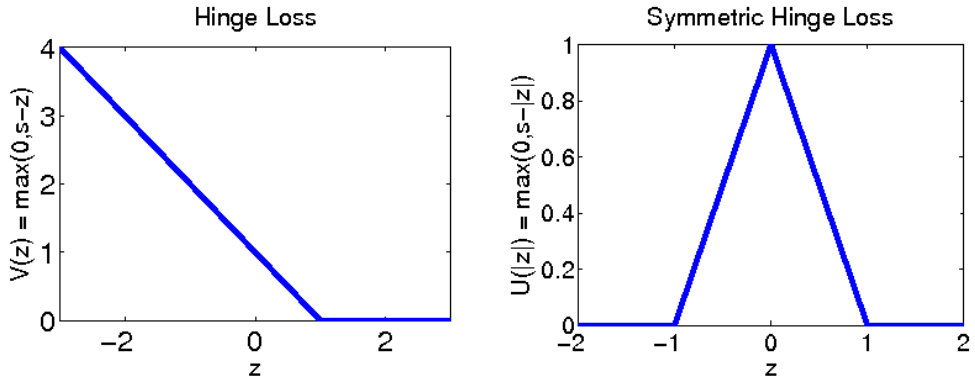


Figure 2.1: Illustration of the non-convexity of problem (2.6) if we define $V(g(\mathbf{x}), y)$ as the the classical hinge loss function $H_1(yg(\mathbf{x}))$ and $U(|g(\mathbf{x})|)$ as $H_1(|g(\mathbf{x})|)$. Symmetric hinge loss $H_1(|g(\mathbf{x})|$ is a non-convex function.

Numerous optimization methods exist. However, let mention that these methods employ gradient techniques, continuation methods, Newton based methods, convex-concave procedure (see [Chapelle 2008b] for a review) to find a local minimum of the problem. There is no convincing evidence of the superiority

of a particular method. Nevertheless, we will mainly be concerned in the sequel by convex-concave algorithms [Collobert 2006, Zhao 2008, Wang 2009] which prove efficient in practice and are able to handle large scale applications. Precisely, we will build upon algorithm of Collobert et al. [Collobert 2006], one of the fastest methods capable to deal with kernels[1] and exhibits the advantage to be easily adaptable to semi-supervised multiple kernel learning using off-the-shelf toolboxes. The adaptation of this algorithm to our concern is exposed in Section 2.3. Before delving into these details, let examine the second assumption exploited by semi-supervised SVM learning.

2.1.3 Laplacian SVM

The principle of Laplacian SVM roots in graph-based methods. Its underlying hypothesis assumes the data lie on low dimensional manifolds the decision function should avoid to traverse. The manifold framework considers that if two points are close in the intrinsic geometry of the marginal distribution $\mathbb{P}(\mathbf{x})$ of the data, they share the same conditional density and, then they share similar label in manifold.

To enforce the smoothness of the decision function along the manifold, a graph is used to measure proximity of samples (labeled points as well as unlabeled ones). Compared to equation (2.1), Laplacian SVM set C^* to zero and transfers the influence of the unlabeled samples in a data-dependent manifold regularization. The corresponding optimization problem is

$$\min_{(f,b)} \quad \frac{\gamma_A}{2}\|f\|^2_{\mathcal{H}} + \frac{\gamma_I}{2}\|f\|^2_{\mathcal{M}} + \sum_{i=1}^{\ell} V(g(\mathbf{x}_i), y_i) \tag{2.7}$$

where γ_A and γ_I specify a trade-off between ambient regularization and manifold deformation. The term $\|f\|^2_{\mathcal{M}}$ models the smoothness assumption over the manifold $\mathcal{M}$ and can be approximated as

$$\|f\|^2_{\mathcal{M}} = \mathbf{g}^T\mathbf{L}\mathbf{g}.$$

Here $\mathbf{g} \in \mathbb{R}^{\ell+u}$ is a vector comprises of the outputs $g(\mathbf{x}_i), i = 1, \cdots, \ell+u$ and $\mathbf{L}$ represents the Laplacian, a square matrix of dimension $\ell + u$. To define it, let $\mathbf{W} \in \mathbb{R}^{(\ell+u)\times(\ell+u)}$ be the adjacency matrix of the similarity graph (as shown in Figure 2.2) with weights

$$\mathbf{W}_{ij} = \begin{cases} \exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma_L^2\right) & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ neighbors} \\ 0 & \text{otherwise} \end{cases} \tag{2.8}$$

The neighborhood of each sample is defined according to its N nearest neighbors and σ_L provides the width of similarity measure in the N neighbors. Let $\mathbf{D}$ be a diagonal matrix such that $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$. The Laplacian is defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$. A normalized variant of this Laplacian can be computed by $\mathbf{L}_n = (\mathbf{I} - \mathbf{D}^{-1}\mathbf{W})$.

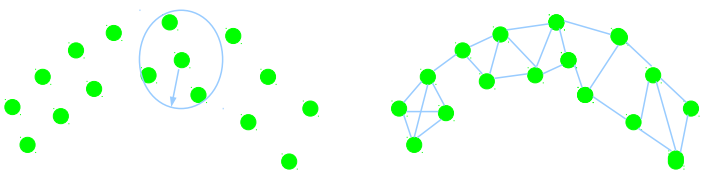


Figure 2.2: Illustration of the construction of graph laplacian. Left: choose the nearest N neighbors. Right: the final graph laplacian.

Let the decision function g belong to an RKHS $\mathcal{H}$ induced by the kernel function κ. A nice property of this manifold regularization was established by [Sindhwani 2005] and Sindhwani et al. stated that

[1] The algorithm of Zhao et al. [Zhao 2008] exploits cutting plane procedure. It is limited to the linear case as its kernelization will require the computation of the coordinates of each sample in a KPCA basis. This operation will harm computation efficiency especially in multiple kernel context we explore hereafter.

problem (2.7) can be advantageously replaced by a classical SVM only over the labeled data with a deformation kernel $\tilde{k}$ expressed as

$$\tilde{\kappa}(\mathbf{x}_i, \mathbf{x}_j) = \kappa(\mathbf{x}_i, \mathbf{x}_j) - \mathbf{k}_{\mathbf{x}_i}^\top \left(\mathbf{I} + \mathbf{M}\right)^{-1} \mathbf{M}\, \mathbf{k}_{\mathbf{x}_j} \tag{2.9}$$

with $\mathbf{k}_{\mathbf{x}} = [\kappa(\mathbf{x}_1, \mathbf{x}) \dots \kappa(\mathbf{x}_{\ell+u}, \mathbf{x})]^\top$. The point cloud norm matrix is $\mathbf{M} = \dfrac{\gamma_I}{\gamma_A}\mathbf{L}^p$, p being an integer. This property will easy the inclusion of manifold assumption in TSVM through our proposed semi-supervised multiple kernel learning scheme. The formulation of the latter problem is the matter of the next section.

Semi-supervised SVMs have been applied successfully in BCI applications. TSVM was proved to be effective for reducing the calibration time in BCI and achieving good performance in classification accuracy [Liao 2007]. [Zhong 2009] compared to the Laplacian SVM and TSVM in a three-task mental imagery BCI experiment. According to their results, Laplacian SVM had a better classification accuracy than TSVM. Other types of Semi-supervised SVMs on BCI applications have also been reported, they always rely on the incremental learning mode, namely, the initial labeled set is enlarged iteratively by the unlabeled data (with their predicted labels) [Qin 2007, Li 2008]. In this chapter, we investigate to explore the manifold information in the framework of TSVM. Expected algorithm shall adjust the cluster assumption or manifold assumption automatically according to the BCI task. In next section, we introduce a multiple kernel version of TSVM to realize such an algorithm.

2.2 Multiple kernel learning for Transductive SVM

Multi-kernel learning (MKL) is a way to incorporate information from different sources to tackle a learning problem in the kernel machinery framework. Numerous efficient methods were proposed recently [Rakotomamonjy 2008a, Xu 2010, Kloft 2011]. Given a set of m kernels κ_k inducing the RKHSs $\mathcal{H}_k, k = 1, \cdots, m$, these methods aim at learning a linear combination of the kernels i.e. $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k=1}^m d_k \kappa_k(\mathbf{x}_i, \mathbf{x}_j)$ with $d_k \geq 0$.

Inspiring from this framework, we propose to include manifold type information in TSVM, a cluster assumption based semi-supervised learning algorithm. Exploiting the flexibility of multiple kernel learning, we intend to design an automatic procedure that we will learn the appropriate assumptions by assigning the appropriate weights d_k to the corresponding kernels. Therefore, we propose, based on equation (2.6), the following formal setup for our TSVM-MKL problem:

$$\min_{\{f_k \in \mathcal{H}_k\}, b, \mathbf{d} \geq \mathbf{0}} \quad \frac{1}{2}\sum_{k=1}^m \frac{a_k}{d_k}\|f_k\|_{\mathcal{H}_k}^2 + C\sum_{i=1}^{\ell} V(g(\mathbf{x}_i), y_i) + C^* \sum_{i=\ell+1}^{\ell+u} U(|g(\mathbf{x}_i)|) \tag{2.10a}$$

$$\text{s.t.} \quad \|\mathbf{d}\|_1 \leq 1 \tag{2.10b}$$

$$\frac{1}{u}\sum_{i=\ell+1}^{\ell+u} g(\mathbf{x}_i) = \frac{1}{\ell}\sum_{i=1}^{\ell} y_i \tag{2.10c}$$

with the convention $\frac{t}{0} = 0$ whenever $t = 0$ and ∞ otherwise. Here the decision function is defined as

$$g(\mathbf{x}) = \sum_{k=1}^m f_k(\mathbf{x}) + b \tag{2.11}$$

where f_k are functions defined over different RHKSs induced by different kernels κ_k. Those kernels can be defined according to some a priori knowledge. In the context of this paper, the kernels will preferably be defined in order to bind manifold and cluster assumptions in a single framework. The vector $\mathbf{d}$ with entries d_k $(1 \leq k \leq m)$ acts as the selector of appropriate kernels (or assumptions). We enforce through equation (2.10b) a ℓ_1-norm penalization on $\mathbf{d}$ to promote sparsity over selected kernels. Finally, a_k is a normalization term, usually set as the trace of the kernel matrix $\mathbf{K}_k$ defined over the training samples. Before we discuss solution of problem (2.10), let precise some elements of its formulation.

2.2.1 Balancing constraint

As stated in Section 2.1.2, transductive kernel machines suffer drawback of unlabeled data assigned to only one class if a balancing constraint such as (2.4) is not imposed. However, relation (2.4) implicitly supposes the knowledge of the unknown labels; thus it has to be approximated. Reminding that $y_i \in \{-1, 1\}$ and a proportion r of positive samples is assumed, it is straightforward to see that (2.4) equivalently reads

$$\frac{1}{u}\sum_{i=\ell+1}^{\ell+u} y_i = 2r - 1.$$

The latter equation still requires the unknown labels, hence they are approximated by the output of the decision function leading to

$$\frac{1}{u}\sum_{i=\ell+1}^{\ell+u} g(\mathbf{x}_i) = 2r - 1.$$

Unfortunately, r is unknown; it is replaced in practice by [Chapelle 2006]

$$r = \frac{1}{2\ell}\sum_{i=1}^{\ell}(1 + y_i)$$

which brings us to the constraint (2.10c).

2.2.2 Loss functions

If we define the loss functions V and U involved in (2.10a) based on the classical hinge loss, we get the shapes displayed in figure 2.1 and especially the shape of symmetric hinge loss $U(|z|) = H_1(|z|)$. However, an effective and flexible definition of $U(|z|)$ can be expressed as

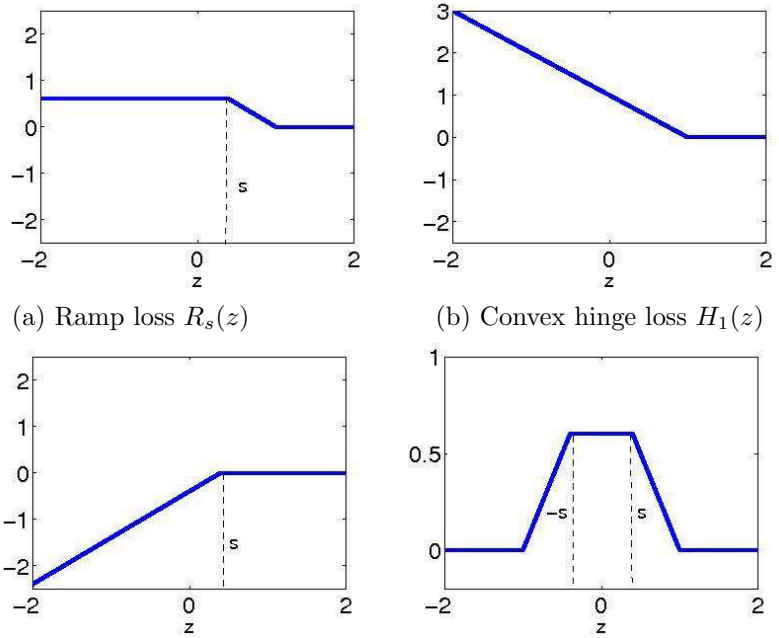
$$U(|z|) = R_s(z) + R_s(-z) - (1 - s) \quad \text{with} \quad 0 \leq s < 1. \tag{2.12}$$

Here $R_s(z)$ is the Ramp loss defined as $R_s(z) = H_1(z) - H_s(z)$ with the expression (2.3b) of $H_s(z)$. An illustration of these functions is shown in Figure 2.3. It should be noticed that when $0 < s < 1$, we obtain a clipped symmetric hinge loss function [Collobert 2006] we plot in Figure 2.3 (d). The normal symmetric hinge loss is recovered by setting $s = 0$. The main invoked reason at the favor of the clipped symmetric hinge loss is the gain of sparsity in the number of support vectors yielded by the optimizer [Collobert 2006].

As a direct consequence of expression (2.12), solving the optimization problem with the clipped symmetric hinge function is equivalent to solve a SVM-type problem with the labeled data and also the unlabeled data counted twice with the artificial labels $\{-1, 1\}$. A hinge loss function is applied for labeled data while the non-convex ramp loss function will be considered for the unlabeled data. Hence, without loss of generality and in accordance with [Collobert 2006], we adopt the following convention for the putative labels of unlabeled samples: we set,

$$y_i = \begin{cases} 1 & \text{when} \quad \ell + 1 \leq i \leq \ell + u \\ -1 & \text{when} \quad \ell + u + 1 \leq \ell + 2u. \end{cases} \tag{2.13}$$

Although this trick facilitates the use of efficient off-the-shelves SVM solvers, it increases the complexity of the problem as the training set size is increased by u the number of unlabeled data.



(a) Ramp loss $R_s(z)$ (b) Convex hinge loss $H_1(z)$

(c) Concave hinge loss $-H_s(z)$ (d) Clipped ramp loss

Figure 2.3: Illustration of Ramp loss $R_s(z) = H_1(z) - H_s(z)$ and the clipped symmetric hinge loss function $U(|z|)$ for unlabeled data.

2.2.3 New formulation of TSVM-MKL

Based on previous analyses, the following new TSVM-MKL optimization problem is attained:

$$\min_{\{f_k\},b,\mathbf{d}\geq\mathbf{0}} \quad \frac{1}{2}\sum_{k=1}^{m}\frac{a_k}{d_k}\|f_k\|_{\mathcal{H}_k}^2 + C\sum_{i=1}^{\ell}H_1(y_i g(\mathbf{x}_i)) + C^*\sum_{i=\ell+1}^{\ell+2u}R_s(y_i g(\mathbf{x}_i)) \tag{2.14a}$$

$$\text{s.t.} \quad \|\mathbf{d}\|_1 \leq 1 \tag{2.14b}$$

$$\frac{1}{u}\sum_{i=\ell+1}^{\ell+u}g(\mathbf{x}_i) = \frac{1}{\ell}\sum_{i=1}^{\ell}y_i. \tag{2.14c}$$

To address it, we resort to DC (Difference of Convex functions) algorithm [Tao 1998] which is closely related to the Concave Convex Procedure (CCCP) [Yuille 2001].

2.3 Solving the multiple kernel TSVM problem

TSVM-MKL inherits the non-convexity and non-smoothness of TSVM which is related to the clipped symmetric hinge loss or the ramp loss. Similar to [Collobert 2006], we employ the DC (Difference of Convex functions) programming to circumvent this shortcoming of TSVM. The choice of this particular optimization method is motivated by two main reasons: first, the ramp loss function in (2.14a) being defined as the difference of two hinge loss functions, DC method directly applies. Second, DC method solves a non-convex (an possibly non-smooth) through multi-stage convex problems issued from a linearization procedure. Solving each convex problem can be achieved by adapting existing convex problem solvers. Hence, we begin with reviewing the materials of DC programming. Next we present its application to handle the problem (2.14).

Algorithm 1 Iterative scheme of DC programming

Set an initial estimation $\boldsymbol{\theta}^0$
repeat
Solve the convex problem

$$\boldsymbol{\theta}^{t+1} = \text{argmin}_{\boldsymbol{\theta}} J_1(\boldsymbol{\theta}) - \left\langle \nabla_{\boldsymbol{\theta}} J_2(\boldsymbol{\theta}^t), \boldsymbol{\theta} \right\rangle \tag{2.15}$$

$t = t + 1$
until convergence of $\boldsymbol{\theta}$

2.3.1 Principle of DC programming

Consider the general case of a non-convex optimization problem: $\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$. DC programming decomposes the criterion $J(\boldsymbol{\theta})$ as the difference of two convex functions (the decomposition is not unique) $J(\boldsymbol{\theta}) = J_1(\boldsymbol{\theta}) - J_2(\boldsymbol{\theta})$ and solves iteratively the problem. The iterative scheme yielded is summarized by Algorithm 1, where at each iteration the concave part ($-J_2(\boldsymbol{\theta})$) of the cost function is approximated by its affine minorization. Notice that in relation (2.15), $\nabla_{\boldsymbol{\theta}} J_2(\boldsymbol{\theta}^t)$ denotes a sub-gradient [2] [Rockafellar 1996] of J_2 at the current solution $\boldsymbol{\theta}^t$. One can easily see that the cost $J_1(\boldsymbol{\theta}) - J_2(\boldsymbol{\theta})$ decreases after each iteration by summing the following two inequalities resulting from (2.15) and from the concavity of $-J_2$

$$\begin{aligned} J_1(\boldsymbol{\theta}^{t+1}) - \langle \nabla_{\boldsymbol{\theta}} J_2(\boldsymbol{\theta}^t), \boldsymbol{\theta}^{t+1} \rangle &\leq J_1(\boldsymbol{\theta}^t) - \langle \nabla_{\boldsymbol{\theta}} J_2(\boldsymbol{\theta}^t), \boldsymbol{\theta}^t \rangle \\ -J_2(\boldsymbol{\theta}^{t+1}) &\leq -J_2(\boldsymbol{\theta}^t) - \langle \nabla_{\boldsymbol{\theta}} J_2(\boldsymbol{\theta}^t), \boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^t \rangle. \end{aligned}$$

The convergence of this algorithm to a local minimum is guaranteed [Tao 1998, Yuille 2001]. A similar procedure applies when the optimization problem comes with non-convex constraints. A work around proposed by [Smola 2005] consists in also expressing the DC decomposition of the constraints, linearizing the objective function and the constraints at the current solution. Hence each iteration simplified to a constrained convex problem.

2.3.2 Application to TSVM-MKL problem

2.3.2.1 Algorithm derivation

Problem (2.14) is non-convex because of the non-convexity of the Ramp loss function. Its careful examination shows that constraints (2.14b - 2.14c) are convex. Therefore, we solely need to find the DC decomposition of the objective function (2.14a) and run algorithm 1 with the mentioned constraints. Using the definition of the Ramp loss function $R_s(z) = H_1(z) - H_s(z)$, we attain the following DC decomposition of (2.14a):

$$\begin{aligned} J_1(\boldsymbol{\theta}) &= \frac{1}{2} \sum_{k=1}^{m} \frac{a_k}{d_k} \|f_k\|^2_{\mathcal{H}_k} + C \sum_{i=1}^{\ell} H_1(y_i g(\mathbf{x}_i)) + C^* \sum_{i=\ell+1}^{\ell+2u} H_1(y_i g(\mathbf{x}_i)) \\ J_2(\boldsymbol{\theta}) &= C^* \sum_{i=\ell+1}^{\ell+2u} H_s(y_i g(\mathbf{x}_i)) \end{aligned} \tag{2.16}$$

Parameter vector $\boldsymbol{\theta}$ comprises of f_k $(1 \leq k \leq m)$, bias term b and weights of kernels $\mathbf{d}$. Now, let find the dot product:

$$\left\langle \boldsymbol{\theta}, \nabla_{\boldsymbol{\theta}} J_2(\boldsymbol{\theta}^t) \right\rangle = C^* \sum_{i=\ell+1}^{\ell+2u} \left\langle \boldsymbol{\theta}, \nabla_{\boldsymbol{\theta}} H_s(y_i g^t(\mathbf{x}_i)) \right\rangle$$

[2] Let $J_2(\boldsymbol{\theta})$ a convex function defined over $\mathbb{R}^d$. A sub-gradient of J_2 at the point $\boldsymbol{\theta}^t$ is an element of the sub-differential, a set defined as $\partial J_2(\boldsymbol{\theta}^t) = \left\{ \boldsymbol{\beta} \in \mathbb{R}^d : J_2(\boldsymbol{\theta}) \geq J_2(\boldsymbol{\theta}^t) + \langle \boldsymbol{\theta} - \boldsymbol{\theta}^t, \boldsymbol{\beta} \rangle, \forall \boldsymbol{\theta} \in \mathbb{R}^d \right\}$. The sub-differential reduces to the gradient when J_2 is differentiable at $\boldsymbol{\theta}^t$.

where $\nabla_{\boldsymbol{\theta}} H_s(y_i g^t(\mathbf{x}_i))$ is the gradient taken at the current decision function $g^t(x)$. As $J_2(\boldsymbol{\theta})$ is independent of $\mathbf{d}$, it should suffice to calculate $\langle \boldsymbol{\theta}, \nabla_{\boldsymbol{\theta}} H_s(y g^t(\mathbf{x})) \rangle$ which will involve terms related to f_k and the bias b. Recalling the definition (2.3b) of $H_s(z)$ and using the reproducing property of Hilbert space i.e. $f_k(\mathbf{x}) = \langle f_k, \kappa(\mathbf{x}, \cdot) \rangle_{\mathcal{H}_k}$, we obtain the following relations

$$\begin{aligned} \nabla_b H_s(y g^t(\mathbf{x})) &= \nu y \\ \nabla_{f_k} H_s(y g^t(\mathbf{x})) &= \nu y \kappa_k(\mathbf{x}, \cdot) \end{aligned}$$

where the scalar ν [3] represents the gradient of hinge loss $\partial H_s(z)$ at $z = y g^t(\mathbf{x})$:

$$\nu = \begin{cases} -1 & \text{if } y g^t(\mathbf{x}) < s \\ 0 & \text{otherwise} \end{cases} \tag{2.17}$$

It is worth mentioning that hinge loss function is differentiable everywhere except in $z = s$. To be consistent, we should consider the sub-gradient at that point. However, following [Collobert 2006] we arbitrary set $\nu = 0$ at $z = s$. Gathering all informations, we get

$$\begin{aligned} \left\langle \boldsymbol{\theta}, \nabla_{\boldsymbol{\theta}} H_s(y g^t(\mathbf{x})) \right\rangle &= \nu y b + \nu y \sum_{k=1}^{m} f_k(\mathbf{x}) = \ \nu y g(\mathbf{x}) \\ \left\langle \boldsymbol{\theta}, \nabla_{\boldsymbol{\theta}} J_2(\boldsymbol{\theta}^t) \right\rangle &= C^* \sum_{i=\ell+1}^{\ell+2u} \nu_i y_i g(\mathbf{x}_i). \end{aligned}$$

With all these elements, the application of DC programming to TSVM-MKL leads to algorithm 2. One can notice that this problem simply turns out to solve iteratively a fully supervised multiple kernel SVM with additional balancing constraint which does not harm the solution computation. So we can benefit from any efficient off-the-shelf sparse MKL solver as those presented in [Rakotomamonjy 2008a, Xu 2010].

Algorithm 2 Iterative procedure to solve TSVM-MKL

Set an initial estimation $\mathbf{d}^0$, b^0, f_k^0 and $t = 0$.
repeat
Calculate the terms ν_i, $i = \ell + 1, \ldots, \ell + 2u$ using (2.17).
Determine $\mathbf{d}^{t+1}, b^{t+1}, f_k^{t+1}, k = 1, \ldots, m$ solution of

$$\begin{aligned} \min_{\{f_k\}, b, \mathbf{d} \geq \mathbf{0}} \quad & J_1(f_k, b) - C^* \sum_{i=\ell+1}^{\ell+2u} \nu_i y_i g(\mathbf{x}_i) \\ \text{s.t.} \quad & \|\mathbf{d}\|_1 \leq 1, \quad \text{and} \quad \frac{1}{u} \sum_{i=\ell+1}^{\ell+u} g(\mathbf{x}_i) = \frac{1}{\ell} \sum_{i=1}^{\ell} y_i \end{aligned} \tag{2.18}$$

with the expression of J_1 given by (2.16)
until a convergence criterion is satisfied.

2.3.2.2 Solving each iteration of TSVM-MKL

For completeness sake, we present in the sequel an adaptation of SimpleMKL of [Rakotomamonjy 2008a] to handle convex problem (2.18). Natively, the approach is iterative and can be summarized as follows. Assume the weights d_k are fixed, problem (2.18) turns to be a normal SVM. Let $\tilde{J}(\mathbf{d})$ be the minimum.

[3] Formally, we should have written ν^t to emphasize the fact that the gradient is taken at iteration t. However, for simplicity sake, we will the notation ν in the sequel.

As f_k, $k = 1, \cdots, m$ and b explicitly depend on $\mathbf{d}$, the coefficients d_k are therefore derived by solving the convex problem:

$$\min_{\mathbf{d} \geq \mathbf{0}} \tilde{J}(\mathbf{d}) \quad \text{s.t.} \quad \|\mathbf{d}\|_1 \leq 1 \tag{2.19}$$

The optimization can be achieved by a gradient method

$$\mathbf{d} \quad \leftarrow \quad \mathbf{d} - \tau \nabla_{\mathbf{d}} \tilde{J}(\mathbf{d}) \tag{2.20}$$

projected onto the positive orthant of the ℓ_1-ball to ensure feasibility of the solution. The new solution $\mathbf{d}$ is therefore plugged back into (2.18) which is solved for f_k and b. The procedure alternates between the calculation of $\mathbf{d}$ and the computation of f_k and b until a convergence criterion is met. In our simulation, convergence is deemed reached when $\mathbf{d}$ does not evolve anymore.

To complete our description, it just remains to present how the problem (2.18) is solved for fixed $\mathbf{d}$. The corresponding lagrangian is:

$$\mathcal{L} = J_1(f_k, b) - C^* \sum_{i=\ell+1}^{\ell+2u} \nu_i y_i g(\mathbf{x}_i) - \alpha_0 \left(\frac{1}{u} \sum_{i=\ell+1}^{\ell+u} g(\mathbf{x}_i) - \frac{1}{\ell} \sum_{i=1}^{\ell} y_i \right)$$

with α_0 the Lagrange parameter. Using properties of convex functions, the sub-gradient of the Hinge loss writes:

$$\partial H_1(z)/\partial z = \begin{cases} 0 & \text{if } z > 1 \\ -1 & \text{if } z < 1 \\ -\tilde{\eta} & \text{if } z = 1 \quad \text{with } 0 \leq \tilde{\eta} \leq 1 \end{cases}$$

Let $\eta = -\partial H_1(z)/\partial z$ a parameter in the range $(0, 1)$. Therefore, the optimality condition w.r.t to primal variable f_k leads to:

$$\frac{a_k}{d_k} f_k - C \sum_{i=1}^{\ell} y_i \eta_i \kappa_k(\mathbf{x}_i, \cdot) - C^* \sum_{i=\ell+1}^{\ell+2u} y_i (\eta_i + \nu_i) \kappa_k(\mathbf{x}_i, \cdot) - \frac{\alpha_0}{u} \sum_{i=\ell+1}^{\ell+2u} \kappa_k(\mathbf{x}_i, \cdot) = 0$$

from which we obtain:

$$f_k(\mathbf{x}) \;=\; \frac{d_k}{a_k} \sum_{i=0}^{\ell+2u} (\alpha_i y_i + C^* \gamma_i) \kappa_k(\mathbf{x}_i, \mathbf{x}), \quad \forall k = 1, \cdots, m.$$

with the following notations and conventions:

- $\alpha_i = C\eta_i$, $\forall\, i = 1, \cdots, \ell$. Due to the definition of η, we naturally have the box constraint $0 \leq \alpha_i \leq C$.
- $\alpha_i = C^* \eta_i$, $\forall\, i = \ell + 1, \cdots, \ell + 2u$. Similarly the associated box constraint is $0 \leq \alpha_i \leq C^*$.
- $y_0 = 1$ and $\kappa_k(\mathbf{x}_0, \mathbf{x}) = \frac{1}{u} \sum_{i=\ell+1}^{\ell+u} \kappa_k(\mathbf{x}_i, \mathbf{x})$.
- $\gamma_i = \nu_i y_i, \forall\, i = 0, \cdots, \ell + 2u$ with the convention $\gamma_i = 0, \forall\, i = 0, \cdots, \ell$.

$\mathbf{x}_0$ is a virtual sample used to encode easily the balancing constraint as in [Collobert 2006]. In the same manner, the optimality condition w.r.t. the bias term b gives $\sum_{i=0}^{\ell+2u} (\alpha_i y_i + C^* \gamma_i) = 0$. Finally the dual of (2.18) is the QP problem

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & -\frac{1}{2} \sum_{i,j=0}^{\ell+2u} (\alpha_i y_i + C^* \gamma_i)(\alpha_j y_j + C^* \gamma_j) \kappa(x_i, x_j) + \sum_{i=1}^{\ell+2u} \alpha_i + \frac{\alpha_0}{\ell} \sum_{i=1}^{\ell} y_i \\ \text{s.t.} \quad & \sum_{i=0}^{\ell+2u} (\alpha_i y_i + C^* \gamma_i) = 0, \quad 0 \leq \alpha_i \leq C, \quad \forall\, i = 1, \ldots, \ell \\ & 0 \leq \alpha_i \leq C^*, \quad \forall\, i = \ell + 1, \ldots, \ell + 2u \end{aligned} \tag{2.21}$$

where the kernel κ is simply

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k=1}^{m} \frac{d_k}{a_k} \kappa_k(\mathbf{x}_i, \mathbf{x}_j).$$

This QP problem involves $\ell+2u+1$ variables all box-constrained except α_0. At convergence, the objective value of the dual coincides with $\tilde{J}(\mathbf{d})$ up to the duality gap. Hence the entries of the gradient involved in (2.20) are easily obtained as

$$\nabla_{d_k} \tilde{J} \;=\; -\frac{1}{2a_k} \sum_{i,j=0}^{\ell+2u} (\alpha_i + C^* \gamma_i)(\alpha_j + C^* \gamma_j) \kappa_k(\mathbf{x}_i, \mathbf{x}_j).$$

Finally, algorithm 3 recapitulates the main steps of our TSVM-MKL solver. Although we have presented our solution of TSVM-MKL from the angle embraced in [Rakotomamonjy 2008a], any other MKL approach straightforwardly applies. Hence, we emphasize that to gain in computation efficiency, the described MKL solver can be advantageously replaced by any new MKL solvers.

Algorithm 3 Complete algorithm to solve TSVM-MKL problem

Solve a fully supervised multiple kernel learning using the label data to initialize f_k^0, b_0 and d_k^0, $k = 1, \ldots, m$.
Set $t = 0$.
repeat
  Calculate the terms $\nu_i, i = 1, \ldots, \ell + u$ using (2.17).
  Determine $d^{t+1}, b^{t+1}, f_k^{t+1}$, $k = 1, \ldots, m$ by running the following loop
  repeat
    Solve the dual problem (2.21) for $\mathbf{d}$ fixed.
    Update $\mathbf{d}$ according to (2.20).
  until Convergence of $\mathbf{d}$ or satisfaction of other convergence criterion.
  Set $t = t + 1$.
until convergence of ν_i or other criterion satisfaction

2.3.2.3 Numerical complexity of TSVM-MKL

The proposed algorithm presents a certain computation burden we study hereafter. As TSVM-MKL relies on multi-kernel framework of simpleMKL [Rakotomamonjy 2008a] and TSVM [Collobert 2006], the overall complexity of the algorithm is tied to the complexity of these methods.

For instance, when solving TSVM via CCCP approach, training amounts to solving a series of single kernel SVM optimization problems with $\ell + 2u$ variables. Hence it has a worst case complexity of $O((\ell+2u)^3)$. However, a few iterations are needed in practice to obtain convergence of TSVM. Moreover Collobert et al. [Collobert 2006] empirically found that such a CCCP-TSVM scheme scales quadratically.

Our TSVM-MKL has a similar behavior but with a greater computation demand. Indeed each iteration requires solving a multiple kernel problem which results in calculation of several SVM problems with $\ell + 2u$ variables. As for TSVM, a few iterations nI (in average 5-10 iterations in our empirical evaluations) of DC outer loop are typically necessary to observe convergence of our algorithm. Hence the complexity of proposed method can be approximated as nI multiple of the complexity of a convex SVM-MKL method. In comparison with TSVM, the increase in computational cost of our TSVM-MKL is mostly due to multiple kernel problem solving. Nevertheless, TSVM-MKL does not require a tedious search of kernel parameters as in TSVM or Laplacian SVM but rather leverages different assumptions on the underlying marginal distribution of the data.

2.4 Related work

In this section, we summarized the related work on inductive semi-supervised learning and the SSL algorithms based on both cluster assumption and manifold assumption. Finally, we point out their relation to the proposed method, and highlight the advantages of TSVM-MKL.

- Manifold Regularization (MR) [Belkin 2006]. MR exploits the geometry of the probability distribution that generates the data and incorporates it as an additional regularization term. Aforementioned Laplacian SVM is a special case of this framework. Contrast to the variety of purely graph-based approaches, the MR framework with an ambient defined RKHS and the associated Representer theorems result in a natural out-of-sample extension from the data set (labeled and unlabeled) to novel examples. They rewrite the optimization as follows:

$$\underset{f\in\mathcal{H}_k}{\operatorname{argmin}} \quad \sum_{i=1}^{\ell} V(g(\mathbf{x}_i), y_i) + \lambda_k \|f\|_k + \lambda_I \|f\|_I$$

  where λ_k and λ_I are trade-off parameters. Let sub-manifold $\mathcal{M}$ to be the support of the input density function $\mathbb{P}(\mathbf{x})$. A natural choice for $\|f\|_I$ is $\int_{\mathcal{M}} \langle \nabla_{\mathcal{M}} f, \nabla_{\mathcal{M}} f \rangle$ which essentially accounts for the gradient of the function on the data manifold. This work has also been extended to multiple view setting where we have two views to the instance space $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$. Each view has its own kernel, and its own geometry on the instance space. Data might be sitting on different manifolds, each of which corresponding a different view.

- Mutli-manifold framework [Goldberg 2009] designs a "cluster-then-label" learning when the data consist of multiple intersecting manifolds. It consists of three main steps: (1) use the unlabeled data to form a small number of decision sets in the ambient space; (2) estimate the target function within a particular decision set by a supervised learner; (3) predict a new test point by the target function in the decision set it falls into. For each decision set, they perform spectral clustering on the graph of labeled and unlabeled points, each resulting cluster represents a seperate manifold. Their method involved Hellinger-distance-based graphs and size-constrained manifold clustering which induces a highly complex model.

- SemiBoost [Mallapragada 2009] is a boosting framework for semi-supervised learning. It exploits both manifold and cluster assumption in one training classification model. Efficient computation can be achieved by iterative boosting.

- One existing method closely related to our algorithm relies on regularization framework. Indeed, [Dai 2007] proposed graph Laplacian kernels selection by reformulating problem (2.5) in the multiple kernel sense. However, the found solution is restrictive as the combination of both assumptions is performed in purely transductive way. Hence, it cannot be extended to handle the out-of-sample cases as we will do in our BCI application.

- Another view of the problem is adaptive regularization for TSVM [Xu 2009] which learns different predictions issued from classifiers with different strengths of cluster assumption. These predictions are linearly binded under a manifold regularization. Although the method empirically proves performing, it suffers the same drawback as [Dai 2007] because it is transductive in essence.

Compared with previous methods, our proposed TSVM multiple kernel framework exhibits the following advantages: (1) TSVM-MKL is an inductive model and can handle the out-of-sample case effectively. (2) The adaptive regularization of [Xu 2009] hierarchizes manifold and cluster assumptions while our TSVM-MKL relies on base kernels and manifold kernels to implement these two assumptions separately. Thereby TSVM-MKL gains from the flexibility of multiple kernel learning, and profits the efficiency of new MKL solvers. (3) When we discard all the manifold kernels from the kernel pool, this algorithm can be regarded as a pure cluster-assumption based method. While for those problems that match the manifold assumption perfectly, we can only keep the manifold kernels in the kernels pool to enhance the effect of

manifold assumption. Compared with the algorithms proposed by [Goldberg 2009, Mallapragada 2009], TSVM-MKL has a smaller computation complexity, and a larger flexibility.

2.5 Numerical evaluation

To evaluate the effectiveness of our TSVM-MKL, we conduct an extensive comparison with the single-assumption-based semi-supervised SVM algorithms. TSVM [Collobert 2006] and Laplacian SVM (LapSVM) [Belkin 2006] are adopted as the representative algorithms that based on cluster and manifold assumption respectively. TSVM problem solved by DC programming involves the setting of the kernel parameter σ, the regularization parameters C, C^* (see Eq 2.6) and the hyper-parameter s of the ramp loss function. Besides the specification of the kernels (base kernels and/or manifold kernels (2.9)), TSVM-MKL also requires the specification of the same hyper-parameters. Laplacian SVM requires the choice of γ_A, γ_I and the kernel parameter. For this algorithm, we use authors [Sindhwani 2005] own implementation. The methods were evaluated in three different ways: we first conducted experiments based on transductive and inductive settings in order to compare our approach with TSVM and LapSVM following the same experimental protocol as in [Sindhwani 2005]. Then, we extended the empirical evaluation to a setup we will term semi-supervised learning cross validation style. These settings are clarified and the observed results are exposed in subsequent sections.

2.5.1 Evaluation under transductive and inductive settings

2.5.1.1 Experimental setting

As summarized in Table 2.1, five binary classification benchmark data sets (G50c, Text, Page, Link and Pagelink) were selected from [Sindhwani 2005]. Semi-supervised learning can be either transductive or inductive. A transductive learner only works on the labeled and unlabeled training data, and cannot handle unseen data contrary to the inductive classifier. Hence we apply the following setups:

- **Transductive setting:** in transductive setting, the training set comprises of n samples, ℓ of that are labeled. Performance of each algorithm is evaluated by predicting the labels of $n-\ell$ unlabeled samples.
- **Inductive setting:** in the inductive setting, the training set comprises of $\ell+u$ samples (ℓ labeled as before, and u unlabeled) and the test set comprises of $n-\ell-u$ samples. With the same implementation in [Sindhwani 2005], we divide the remaining $n-\ell$ samples into five equal folds. At each time, one fold is selected as the unseen test set and the rest four folds serve as unlabeled set (also as the validation set). We repeat this procedure until all the five folds have been selected as the test set. Algorithm is evaluated by the mean performance on predicting the novel out-of-sample test examples.

Table 2.1: Benchmark data sets used in our experiments. Labeled data number ℓ are for transductive setting and inductive setting.

Data set	dimensionality	labeled ℓ	total points n
G50c	50	50	550
Text	7511	50	1946
Page	3000	12	1051
Link	1840	12	1051
Pagelink	4840	12	1051

In this section, we evaluate the performance of TSVM-MKL in both transductive setting and inductive setting to compare with the results reported in [Sindhwani 2005, Collobert 2006]. For this sake we use

the same number of labeled samples as described in Table 2.1 and the same splits of the data sets into labeled and unlabeled sets. In our experiments, we set $C = C^\star$, the values of C and s are selected by grid search over [10 100 1000] and [0 : 0.2 : 0.6] respectively. Gaussian kernels and euclidean nearest neighbor graphs with gaussian weights were used on G50c and Text. Linear base kernel and cosine nearest neighbor graphs with gaussian weights were used for the remaining data sets following [Sindhwani 2005]. Based on the classification accuracy on unlabeled data, finally selected values for σ in both transductive setting and inductive setting experiments are shown in Table 2.2. The obtained results are reported in Tables 2.3 and 2.4.

Table 2.2: Finally selected σ in the experiments

Data set	Values	Data set	Values
G50c	$\{2^{-2},\ 2^{0},\ 2^{2},\ 2^{4},\ 2^{6}\}$	Text	$\{2\ , 3,\ 4\}$
Page	$\{2^{-2},\ 2^{-1},\ 2^{0}\}$	Link	$\{2^{-2},\ 2^{-1},\ 2^{0}\}$
Pagelink	$\{2^{-2},\ 2^{-1},\ 2^{0}\}$		

Table 2.3: Transductive setting: misclassification rates (in percent) on unlabeled data

Data set	G50c	Text	Link	Page	Pagelink
SVM	9.7	18.9	26.7	20.8	14.2
LapSVM	5.4(0.6)	10.4(1.1)	14.9(8.8)	10.5(0.7)	6.3(0.6)
TSVM	5.7(1.6)	**6.0(1.1)**	11.6(2.9)	10.6(8.5)	8.6(7.3)
TSVM-MKL	**4.4(0.7)**	6.2(1.6)	**10.0(6.4)**	**8.3(5.2)**	**5.6(5.8)**

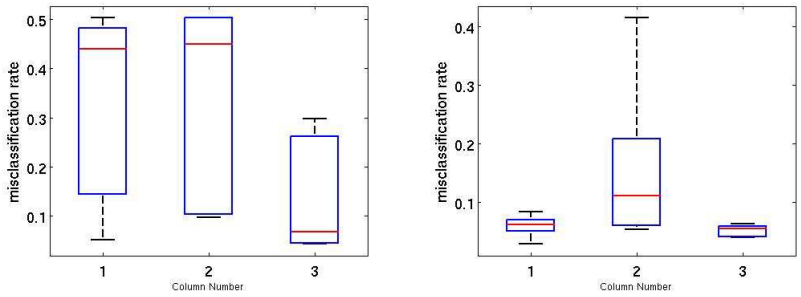
Table 2.4: Inductive setting: misclassification rates (in percent) on unlabeled and test data

Data set	G50c	Text	Link	Page	Pagelink
Algorithm	Unlab Test	Unlab Test	Unlab Test	Unlab Test	Unlab Test
SVM	9.7	20.9	24.8	23.8	25.1
	9.7	20.9	24.8	23.8	25.1
LapSVM	4.9	9.9	21.2(21.4)	14.1(7.1)	12.8(8.4)
	5.0	9.7	21.1(21.3)	15.5(6.1)	14.4(6.0)
TSVM	5.4(1.1)	6.5(1.1)	11.6(2.7)	11.5(8.3)	9.0(7.2)
	6.1(1.3)	6.8(1.0)	11.2(2.8)	11.6(8.6)	8.9(7.0)
TSVM-MKL	**4.5(5.0)**	**6.2(1.4)**	**9.6(6.0)**	**8.5(4.6)**	**5.6(5.7)**
	4.7(5.2)	**6.4(1.5)**	**9.4(6.1)**	**9.0(4.9)**	**6.2(5.7)**

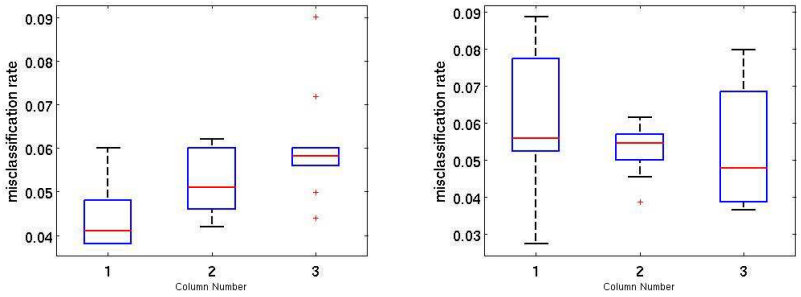
Next the influence of the proportion of the labeled set size, that is ℓ/n, on performances is analyzed on some of the data sets. We have considered the respective proportions: $1\%, 5\%, 10\%$ and 20%. The labeled and unlabeled data was generated accordingly and the simulations were carried over 10 runs. The empirical evidences are illustrated in Figure 2.4 to 2.7.

2.5.1.2 Experimental results and analysis

We first present the results when using the number of labeled samples as shown in Tabe 2.1. Table 2.3 shows the mean results and the standard deviations of involved algorithms on 10 runs in transductive setting. Table 2.4 reports the mean results and the standard deviations of TSVM-MKL on 10 runs when predicting the labels of unlabeled and test data in inductive setting.



(a) $\frac{\ell}{n} = 1\%$ (b) $\frac{\ell}{n} = 5\%$



(c) $\frac{\ell}{n} = 10\%$ (d) $\frac{\ell}{n} = 20\%$

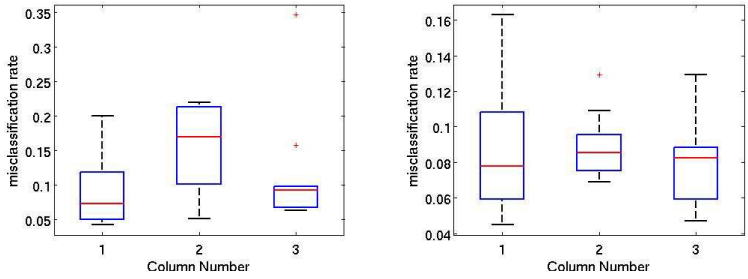
Figure 2.4: Results in terms of misclassification rates for different ratios of labeled data $\frac{\ell}{n}$ on G50C. Indexes of column number 1, 2 and 3 denote TSVM-MKL, LapSVM and TSVM.

Results of SVM and LapSVM on G50C and Text are taken from [Sindhwani 2005]. They learned a regular SVM on labeled data and predict the labels of unseen testing data. We redo all the other experiments in the same experimental setting. Experiments of SVM are implemented in this way: train an SVM on labeled set, and test it on unseen test set.

From these results we can see that TSVM-MKL achieves the best solution in most cases. It indicates that the combination of cluster and manifold assumption helps improving the classification performances. This improvement is more prominent in inductive setting where the test data are unseen by the algorithms. We can particularly remark the better performances of TSVM and TSVM-MKL over Laplacian SVM in Table 2.4. This is justified by the fact that most of the data sets are text classification applications which are well suited for cluster assumption [Chapelle 2008b]. Embedding manifold kernels in TSVM through multiple kernel learning boosts the results of TSVM and emphasizes the utility of data geometry.

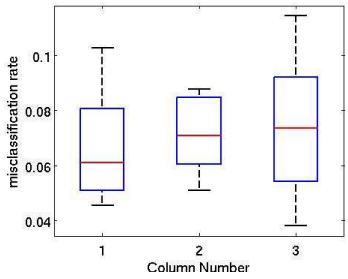
We have also investigated performances of TSVM-MKL, LapSVM and TSVM with different sizes of labeled set. To simplify the presentation, we solely report the results for the inductive setting where hold-out samples are used to assess the effectiveness of each method. Figure 2.4 to 2.7 show comparison results on relevant data sets.

The following remarks can be made. Regarding G50C (Figure 2.4), the lack of sufficient labeled data (1% of overall data set) involves a failure of TSVM-MKL and LapSVM. This tends to illustrate that the geometrical information (manifold kernels) was not fully exploited by both methods. In comparison TSVM performs well. When one increases the number of labeled samples, TSVM-MKL and LapSVM reduces the gap in performances with TSVM. It can be observed that TSVM-MKL matches up with TSVM for $\ell/n = 5\%$ and $\ell/n = 20\%$ and can be deemed superior to TSVM for a mid range of labeled set size, that is $\ell/n = 10\%$. For PAGE, LINK and PAGELINK (Figure 2.5 to Figure 2.7), the results produced by TSVM-MKL are more consistent when varying the proportions of labeled samples. As a conclusion for these data sets, TSVM-MKL exhibits better performances than LapSVM and leverages

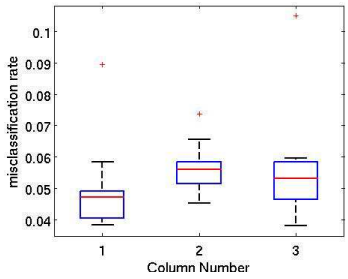


(a) $\frac{\ell}{n} = 1\%$

(b) $\frac{\ell}{n} = 5\%$

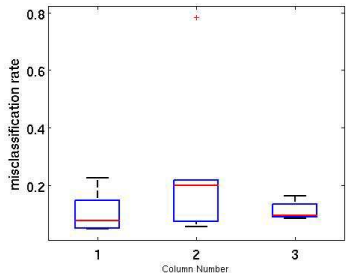


(c) $\frac{\ell}{n} = 10\%$

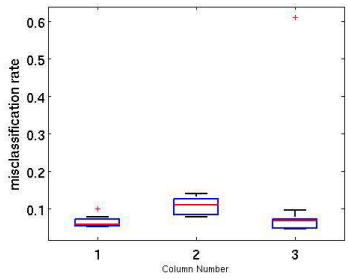


(d) $\frac{\ell}{n} = 20\%$

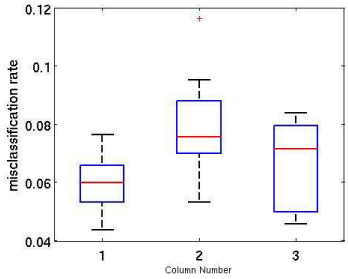
Figure 2.5: Results in terms of misclassification rates for different ratios of labeled data $\frac{\ell}{n}$ on PAGE. Indexes of column number 1, 2 and 3 denote TSVM-MKL, LapSVM and TSVM.



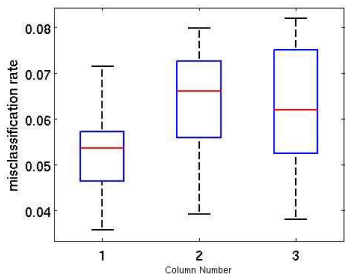
(a) $\frac{\ell}{n} = 1\%$



(b) $\frac{\ell}{n} = 5\%$

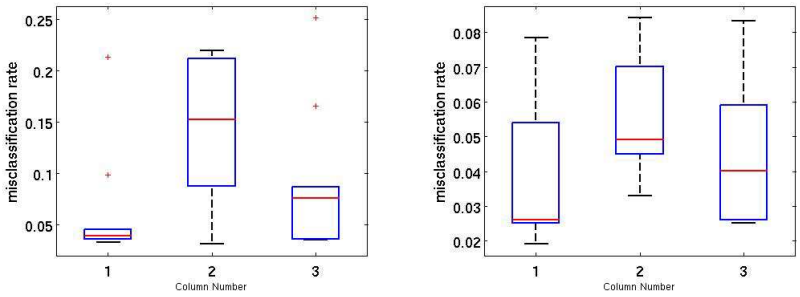


(c) $\frac{\ell}{n} = 10\%$



(d) $\frac{\ell}{n} = 20\%$

Figure 2.6: Evaluation of TSVM-MKL, LapSVM and TSVM with different ratio of labeled data $\frac{\ell}{n}$. Index of column number 1, 2 and 3 denote TSVM-MKL, LapSVM and TSVM on data set LINK seperately.



(a) $\frac{\ell}{n} = 1\%$ (b) $\frac{\ell}{n} = 5\%$

(c) $\frac{\ell}{n} = 10\%$ (d) $\frac{\ell}{n} = 20\%$

Figure 2.7: Evaluation of TSVM-MKL, LapSVM and TSVM with different ratios of labeled data $\frac{\ell}{n}$. Performances are misclassification rates. Index of column number 1, 2 and 3 denote TSVM-MKL, LapSVM and TSVM on data set PAGELINK seperately.

the manifold and cluster information to improve the misclassification rates over TSVM.

2.5.2 Evaluation under semi-supervised style cross validation setting

In this setting, the best achievable accuracy obtained by maximizing the accuracy on test set is reported. Evaluation is implemented as follows:

- split the ℓ available labeled data into nF equal folds;
- hold one fold as test data, employ the remaining folds and the unlabeled data to train a semi-supervised model;
- repeat nF times and attain the averaged accuracy.

The three steps are repeated for different combinations of involved hyper-parameters and is selected the model with the best averaged test error. Notice that doing so, the test set does not act as a genuine hold-out samples set but rather as validation set. The interest of this procedure resides in the fact that when one has a very few labeled samples, only these samples can be used to guide model tuning.

To test our model under this setting, we divide the whole data set into labeled set ($\ell = 30\%\, n$) and unlabeled set ($u = 70\%\, n$) and we consider $nF = 3$, hence the test set consists of 10% of the data. The results are averaged over 10 replications and the best achievable test errors are presented in Table 2.5.

Clearly TSVM performs the best under this particular setting. TSVM-MKL hardly attains the same level of performances as TSVM. In the cases where better results are achieved by TSVM-MKL the difference with TSVM is tiny. It seems that TSVM-MKL is more sensitive than TSVM to the size of evaluation set (as here the test set can be viewed as validation set). When model selection is performed over the

Table 2.5: SSL-style cross validation setting: the best achievable accuracy.

Data set	G50c	Text	Link	Page	Pagelink
LapSVM	5.4(1.2)	7.5(1.0)	6.7(1.9)	4.5(1.4)	3.2(1.1)
TSVM	5.7(1.5)	**3.4(0.5)**	5.6(1.8)	**3.8(0.9)**	2.8(1.0)
TSVM-MKL	5.8(2.7)	4.8(0.9)	**5.5(1.3)**	4.0(1.2)	**2.7(0.7)**

unlabeled set, TSVM-MKL tends to select better models as confirmed by the previous results. Indeed, having more validation data allows TSVM-MKL to unravel and learn the appropriate combination of kernels and permits to avoid over-fitting (as TSVM-MKL comes with potentially greater model complexity). Hence it can be expected that more validation informations can alleviate the observed limitation. Nevertheless, it is necessary to investigate data sets with more training samples to confirm or invalidate this observation and intuition. Finally, it is reassuring that TSVM-MKL still consistently performs better than Laplacian SVM.

2.6 Application in BCI data analysis

2.6.1 Application on μ and β based BCI system

Experimental data The EEG-based cursor control experiment was carried out in Wadsworth Center[4]. In this experiment, the subjects sat in a reclining chair facing a video screen and were asked to remain motionless during performance. The subjects used μ or β rhythm amplitude to control vertical position of a target located at the right edge of the video screen. The data set was recorded from three subjects (AA, BB, CC). Each subject's data included 10 sessions. Each session consists of 192 trials. As shown

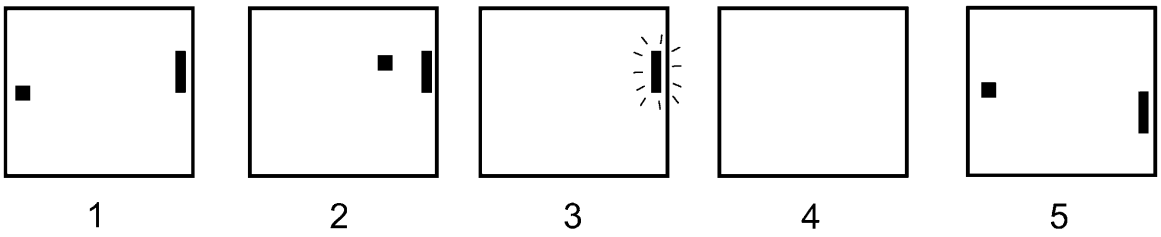


Figure 2.8: Illustration of the cursor control experiment: (1) The target and cursor are present on the screen for 1 s. (2) The cursor moves steadily across the screen for 2 s with its vertical movement controlled by the user. (3) The target flashes for 1.5 s when it is hit by the cursor. If the cursor misses the target, the screen is blank for 1.5 s. (4) The screen is blank for a 1-s interval. (5) The next trial begins (from [McFarland 2006]).

in Figure 2.8, in the one-dimensional mode, the user is presented with a target along the right edge of the screen and a cursor on the left edge. The cursor moves across the screen at a steady rate, with its vertical movement controlled by μ or β rhythm amplitude. The user's task is to move the cursor to the height of the target so that it hits the target when it reaches the right edge of the screen.

Evaluation strategy To compare with the SSL method of [Qin 2007], only the trials with the targets who are at the highest and lowest position of the right edge of the screen (96*10 trials for each subject) were used in our analysis. Data set from sessions 1-6 were used for learning, and data set from session 7-10 acted as the out-of-sample set for test. We first extracted the dynamic common spatial patterns (DCSP) (proposed by [Qin 2007]) for sessions 1-6. Next, we divided all the features into labeled set and unlabeled set. Labeled set consisted of 48 trials from session 1, the rest ones served as unlabeled set. Based on

[4] http://www.ida.first.fraunhofer.de/projects/bci/competition

the transformation matrix obtained on learning set, the DCSP features for test set are attained. We also employ the Laplacian SVM and TSVM to evaluate the proposed fuse-assumption TSVM-MKL on this EEG-based BCI task. Adopting the same criteria of performance in [Qin 2007], the best achievable classification accuracy is the objective of our data analysis.

Signal pre-processing and feature extraction In a BCI system, the signal processing block is mainly composed of three parts: pre-processing, feature extraction and the classification. We emphasize the feature selection in this part, and the famous common spatial patterns (CSP) is adopted as its tool. For the signal pre-processing, raw data is filtered by a band-pass filter with the bounds from 10 Hz to 15 Hz. The filter order is 8. Note that only the samples at the time when the user was controlling the cursor were used, that is, 368 samples each trial for subject AA and BB, 304 samples each trial for subject CC; the samples before and after cursor control were omitted.

CSP is a method that has been applied to EEG analysis to classify the normal versus abnormal EEGs. It aims at finding spatial structures of event-related (de-)synchronization [Ramoser 2000]. In what follows, we present the extraction of the dynamic non-normalized CSP feature, which is utilized in our data analysis and proposed in [Qin 2007].

First, we filter the raw EEG in μ rhythm frequency band. The following CSP feature extraction is based on the filtered signals. In order to reflect the change of brain signals during a trial, we extract a dynamic CSP feature, that is, seperate the time interval of each trial into 5 overlapped time segments[5]. For each time segment, we calculate one CSP feature vector as follows. The CSP analysis in the i^{th} $(i = 1, \cdots, 5)$ time segment involves calculating a matrix $\mathbf{W}_i$ and diagonal matrix $\mathbf{D}_i$ through a joint diagonalization method as follows:

$$\begin{aligned} \mathbf{W}_i \mathbf{Z}_i^+ \mathbf{W}_i^\top &= \mathbf{D}_i \\ \mathbf{W}_i \mathbf{Z}_i^- \mathbf{W}_i^\top &= 1 - \mathbf{D}_i \end{aligned}$$

where $\mathbf{Z}_i^+$ and $\mathbf{Z}_i^-$ are covariance matrices of EEG data matrices $\mathbf{E}_i^+$ and $\mathbf{E}_i^-$ (one row of the EEG data matrices corresponds to one channel EEG signal). "+" and "-" denote two different classes (for the cursor control experiment, they represent two different targets). Using all trials with class "+", the matrix $\mathbf{E}_i^+$ can be constructed by trial-concatenating the filtered EEG data in the i^{th} time segments of every trial. $\mathbf{E}_i^-$ is obtained similarly except that it corresponds to the trials with class "-" . The diagonal elements of $\mathbf{D}_i$ are sorted with a decreasing order.

After obtaining the transformation matrix $\mathbf{W}_i$, we now extract CSP feature in the i^{th} time segment of a trial. We first calculate a covariance matrix using the filtered EEG signals in the i^{th} time segment; then take the first two or the last two main diagonal elements of the transformed covariance matrix[6]. Note that the first two diagonal elements correspond to two largest eigenvalues in the diagonal matrix $\mathbf{D}_i$ above, the last two correspond to its 2 smallest eigenvalues. Based on the cross-validation results obtained from the training set, we find that the first 2 main diagonal elements were more significant for discriminating for subjects AA, CC and the last 2 main diagonal elements were more significant for subject BB. Thus we obtain a 2-dimensional CSP feature for each time segment. Concatenating the CSP features of five time segments, we attain a 10-dimensional CSP feature vector for each trial.

Model selection For TSVM-MKL, we defined a pool of kernels: basic kernel (Gaussian kernel with kernel width $\sigma = 1$) and manifold kernels. For the manifold kernels, relevant hyper-parameters were set as follows: deformed ratios $\frac{\gamma_I}{\gamma_A} = \{10\ 1000\}$, the neighborhood size consists of $N = 40$ samples and the kernel options for adjacency matrix $\sigma_L = \{1\ 10\ 100\ 1000\}$ (see Eq. (2.8)). For such kernel setting, we attain a kernel pool with 9 kernels. We performed a variation of 5-fold cross validation. The strategy $C = C^*$ was adopted, and the value of C and s are selected by grid search over [10 100 1000] and $[0 : 0.2 : 0.6]$ respectively.

[5] The number of overlapped time segments could be different with different types of BCI task.
[6] Selected diagonal elements depend on the subject.

Experimental analysis Table 2.6 lists the best results of our method, the normal LapSVM and TSVM on the same data set. Results of S3VM were adopted from [Qin 2007]. They first separate the unlabeled set into 9 subsets. And then trained a SVM using the dynamic power features from the labeled set, estimated the labels of the following sub-unlabeled set, selected the most confidently classified elements and added them together with their predicted labels to training set for a 1-norm semi-supervised classifiers. They repeated this procedure until enough estimated labels are available to calculate the dynamic CSP features. Note that, the transformation matrix and the classifier's model were updated in each loop of their procedure. In this degree, although our classification accuracies are superior to their's, we use some label information in the dynamic CSP feature extraction procedure for sessions 1-6. TSVM-MKL

Table 2.6: Best classification accuracy (%) for the three subjects AA, BB and CC.

Algorithm	AA	BB	CC	Average
S3VM [Qin 2007]	94.52	91.84	91.51	92.62
LapSVM	96.36	93.75	94.17	94.76
TSVM	97.40	94.79	95.37	95.37
TSVM-MKL	**97.68**	**96.35**	**95.77**	**96.60**

achieves slightly better performance on such CSP feature based BCI data analysis.

Labels information is required for the extraction of CSP features. As a batch learning method, the proposed TSVM-MKL algorithm achieves better classification accuracy in the precondition of all CSP features are calculated in a batch mode. Meanwhile, TSVM-MKL is an inductive model which can handle the out-of-sample case in BCI. It could also be adopted as a self-learning mode which is similar to [Qin 2007].

2.6.2 Application on motor imagery based BCI system

Experimental data This experiment deals with three-class classification of EEG signals. We use here the data sets from BCI Competition III (data set V). These data sets contain EEG records from 3 normal subjects during 4 non-feedback sessions. For simplicity, we denote them as session 0-3. The subjects (referred to as subjects A, B and C) performed 3 tasks: imagination of repetitive self-paced left hand movements (left, class 1), imagination of repetitive self-paced right hand movements (right, class 2) and generation of words beginning with the same random letter (word, class 3) All the four sessions of a given subject were acquired on the same day, each lasting 4 minutes with 5-10 minutes breaks between them, then switched randomly to one of the other two tasks at the operator's request, and after another 15 seconds, switched to a new task again. The class labels were changed with the task at the same time. EEG data are not splitted into trials since the subjects are continuously performing all the mental tasks. Sampling rate was 512 Hz.

Signal pre-processing and feature extraction The raw EEG potentials were first spatially filtered by means of a surface Laplacian. Then, in every 62.5 ms the power spectral density (PSD) in the band 8-30 Hz was estimated over the last second of data with a frequency resolution of 2 Hz for the eight centro-parietal channels C3, Cz, C4, CP1, CP2, P3, Pz and P4. The PSD value for a time sequence $s[i]$ $(i = 0, 1, \cdots, t-1)$ can be estimated via periodogram:

$$\mathrm{PSD}(f_b) = \frac{1}{t}|S(f_b)|^2 = \frac{1}{t}S(f_b)S(f_b)^*$$

where $S(f_b)$ and $S(f_b)^*$ are respectively discrete-time Fourier transform of $s[i]$ and its conjugate:

$$S(f_b) = \sum_{i=0}^{t-1} s[i]e^{-j2\pi f_b i}$$

It is a discrete-time, continous-frequency version of PSD. Here, f_b denotes the frequency band. We select the first 12 components for each channel and attain 96 features for each sample. Each PSD sample of the EEG data is normalized (ℓ_2 normalization) to an interval of $[0, 1]$ [Liao 2007]. Since the BCI system needs a response in every 0.5 s and the EEG data are very noisy, we average the PSD data over 8 consecutive samples. The number of examples used for training and testing sets are listed in Table 2.7 (for simplicity, we denote the training session as Session 0).

Table 2.7: EEG data sets for classification with Semi-supervised algorithms.

Subject	Train(Session0)	Test(Session1)	Test(Session2)	Test(Session3)
A	438	436	434	446
B	434	434	432	434
C	436	428	428	430

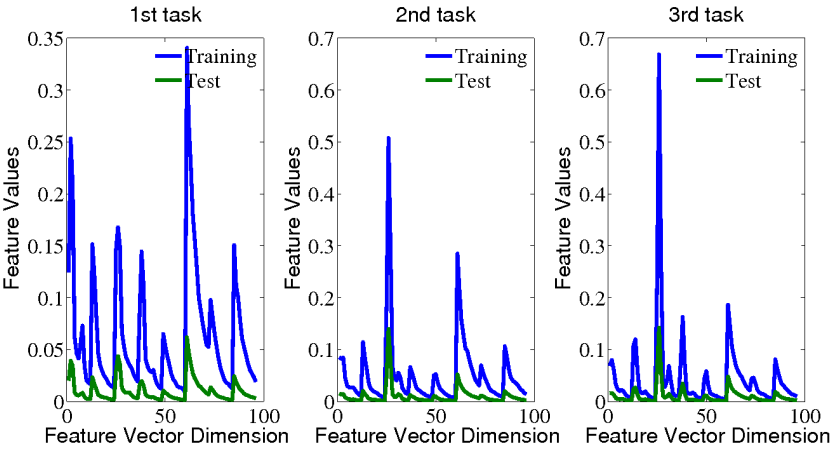
Evaluation strategy To illustrate the change of EEG patterns in different sessions of each subject, Figure 2.9 shows the feature vector values for the three mental tasks of each subject in the training session and all the three testing sessions. These features have been generated by averaging all PSD samples of a given mental task. We can see that the EEG patterns differ in quite a few aspects from subject to subject. And shift a lot from the training session to the test one on each mental task. This spontaneous variability of brain signals between sessions/subjects hinders correct online recognition with any classifier trained with the data of training sessions.

To evaluate the performance of an algorithm fairly, we fix the testing strategy as follows: training a classifier on the training set (in current experiment, means that employing the data from Session 0); keep the hyper-parameters that attained in the model selection procedure unchanged for all the three testing sessions (Session 1-3). The model of TSVM-MKL (as well as the other competitors in semi-supervised learning) is updated session by session to cope with data variability through sessions. For Session 1, labeled set are fixed as the data from the first four folds of Session 0, and the remaining one fold is selected as the unlabeled set. For Session 2 and 3, we set labeled set as Session 0, and the previous testing sessions serve as the unlabeled sets. All the experiments are implemented in the inductive setting. Our strategy is identical with that of [Liao 2007].

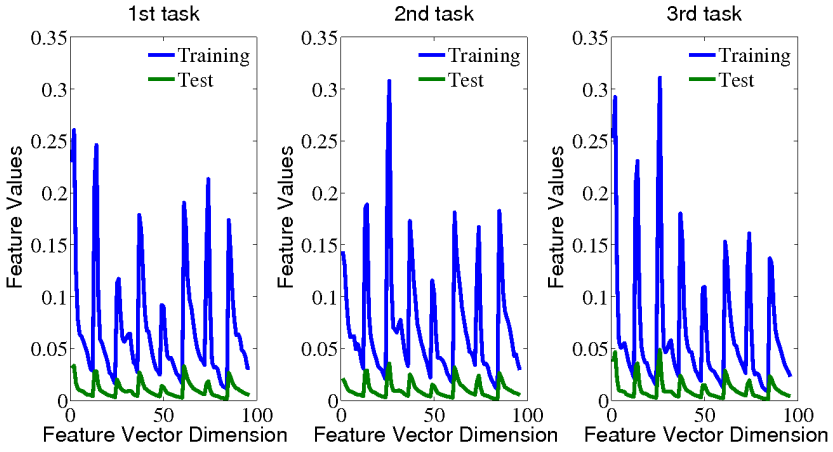
Model selection Current model selection strategies in the application of BCI are always implemented in the transductive setting: perform grid-search-based cross validation on the training set, and then select the final hyper-parameters according to the performance on the unlabeled data [Liao 2007, Zhong 2009]. There are two reasons showed that it is unsuitable to perform the model selection in this way: (1) TSVM is an inductive learner in nature, performance on the unlabeled set cannot demonstrate its generalization ability exactly. (2) Brain activities change naturally over time, thus, the EEG data can be seen as from different data distributions. Performance on the unlabeled data could be far from that on the unseen test data.

In this paper, we propose to implement the model selection of semi-supervised algorithms for BCIs in this way: divide the whole data into nF equal folds, as the EEG data are chronological distribution, the data from the first fold is selected as the labeled data. At each validation process, leave one fold out to serve as the unseen test set for validation (to distinguish from the real test set in the testing process, we denote it as "test-validation set"), and the remaining folds serve as the unlabeled data. We first train a TSVM-MKL classifier on the labeled and unlabeled data, and then evaluate it on the test-validation set. Let u be the number of unlabeled samples, N_{tv} is the number of test-validation samples, Acc_{unl} is the accuracy on unlabeled samples, and Acc_{tv} is the accuracy on the test-validation set. Final hyper-parameters are selected according to:

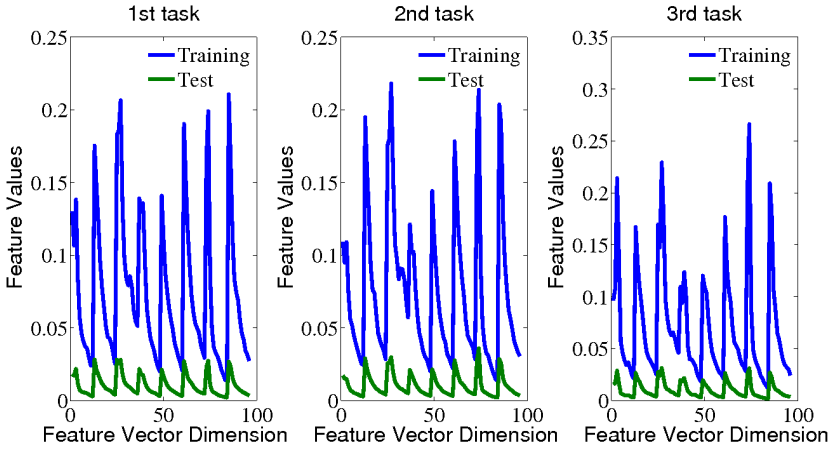
$$\frac{Acc_{unl}}{u} + \frac{Acc_{tv}}{N_{tv}} \tag{2.22}$$



(a) Subject A



(b) Subject B



(c) Subject C

Figure 2.9: Averaged PSD feature vector values for the three mental tasks on each subject (1st task: imaginary left-hand movements; 2nd task: imagery right-hand movements; 3rd task: generation of words). We select the first 12 components for each centro-parietal channel and attain 96 features for each sample.

In our experiments, we adopt "one-against-all" scheme for the multi-class problem. Following choices were made for TSVM-MKL: for the manifold kernels, we fix $p = 1$ and $N = 30$ neighbors (see equations (2.8) and (2.9) and Laplacian definition) unoptimized. For the base part, we adopt the heuristic $uC^* = \ell C$ and set $s = 0.1$ according to the experience. The remaining hyper-parameters $(C, \sigma, \gamma = \frac{\gamma_I}{\gamma_A})$ are selected by a 5-fold cross validation. Grid searches are executed over [1 10 100 500 1000], [0.1 0.32 1.0 3.2 10], and [1 10 100 1000] for C, σ, and γ, respectively. We adopt Gaussian kernel with parameter σ for the non-linear case. One base kernel and one manifold kernel compose the kernel pool of TSVM-MKL.

For the experiments on TSVM, we adopt the heuristic $uC^* = \ell C$ and we set $s = 0.1$ according to the experience. Gaussian kernel was employed for the nonlinear case. Model selection is implemented by grid search over [1 10 100 1000] for C.

Finally, for LapSVM, involved hyper-parameters include N (neighborhood size for graph construction), γ, σ_L (used for adjacency matrix weights), p and kernel parameter σ. We also fix $p = 1$ and $N = 30$ unoptimized. For the linear case, model selection is executed by grid search over [1 10 100 1000] and [0.01 0.1 1 10] over C and σ_L separately. For the non-linear case, we add the choices [0.1 1 10] for σ.

Experimental analysis The experiments with classical SVM are also implemented by employing all label information. In some degree, the results on SVM should be close to the best result achievable. In the experiments of TSVM, LapSVM, and SVM, we always employ the single-kernel case. Table 2.8 shows a comparison of relevant algorithms for the three subjects in linear (L) and non-linear case (N). From Table 2.8 we can see that, compared with single-assumption-based semi-supervised algorithms, the

Table 2.8: A comparison of SVM, TSVM, LapSVM and TSVM-MKL for the three subjects over three consecutive testing sessions. Are reported the accuracies (in percent) attained by each method. The chance level of classification accuracy is 33.3% for three tasks. Symbol L denotes linear case, and N denotes non-linear case.

Subject	Methods	Session 1	Session 2	Session 3	Average
A	SVM (L)	66.3	71.7	77.4	71.8
	SVM (N)	68.4	73.7	77.1	73.1
	TSVM (L)	68.7	70.0	76.4	71.7
	TSVM (N)	67.8	72.1	76.2	72.0
	LapSVM (L)	66.7	69.8	74.0	70.2
	LapSVM (N)	62.8	71.7	75.8	70.1
	TSVM-MKL (L)	65.6	74.4	76.7	**72.2**
	TSVM-MKL (N)	64.3	75.3	78.0	**72.5**
B	SVM (L)	59.0	59.5	66.1	61.5
	SVM (N)	59.0	59.7	67.0	61.9
	TSVM (L)	52.5	56.0	59.5	56.0
	TSVM (N)	59.5	55.6	60.4	58.5
	LapSVM (L)	53.5	57.2	58.8	56.5
	LapSVM (N)	59.5	58.1	64.5	60.7
	TSVM-MKL (L)	56.2	56.3	61.8	**58.1**
	TSVM-MKL (N)	55.4	61.6	65.9	**61.0**
C	SVM (L)	49.3	45.0	49.1	48.8
	SVM (N)	49.3	47.4	51.2	49.3
	TSVM (L)	46.5	47.7	48.1	47.4
	TSVM (N)	46.7	43.7	47.7	46.0
	LapSVM (L)	42.3	49.5	46.5	46.1
	LapSVM (N)	46.3	43.2	47.6	45.7
	TSVM-MKL (L)	46.3	49.8	48.6	**48.3**
	TSVM-MKL (N)	48.8	47.2	49.8	**48.6**

proposed TSVM-MKL can always achieve better classification accuracy in the linear and non-linear cases respectively. These results showed the improvements of TSVM-MKL in the BCI data analysis. And in many cases, the results on TSVM-MKL are close to those of SVM that employed all label information, in this degree, TSVM-MKL could be a valuable choice for the on-line BCI applications.

Adapting TSVM-MKL for channel selection As different mental tasks induce the responses in different brain regions, we believe that channel selection performed for each mental task shall lead to better performance. Hence, we embedded it into the learning process of TSVM-MKL as follows:

- Define a subpool of kernels for each channel. To ensure that the classifer has a smaller computation complexity, each subpool consists of one base and one manifold kernel. Hence, for the total 8 channels, there will be 16 kernels involved in current experiments.
- Recall that d_k acts as the selector of kernels. We constrain the kernels corresponding to the same channel to share the same d_k i.e. for each channel k, we set the kernel regularizer as $\frac{a_{k1}\|f_{k1}\|^2+a_{k2}\|f_{k2}\|^2}{d_k}$ where f_{k1} refers to the basic kernel and f_{k2} to the manifold one.
- Implement Algorithm 3, automatical channel selection is executed by assigning different values of d_k, larger values are given for those channels who have more contributions. When the weight of a channel is smaller than a certain predefined threshold ε (typically we have considered $\varepsilon = 0.01$, the channel will be discarded.
- Adopt "one-against-all" strategy for the multi-class classification task. For each mental task, channel selection and learning process are finished synchronously.

In this experiment, we only investigate the linear case to reduce the computation burden. Table 2.9 shows the performance of TSVM-MKL with/without channel selection cases. The results showed its improvements with the provided strategy. In most cases, the performance with such channel selection strategy can be improved obviously (except the 3rd session of subject C, as subject C always performs not good enough, we can take it as an exception). Such improvements could be explained by Figure 2.10, each channel makes different contribution for different mental task. Take the channel "CP2" as an example, it takes important role in the 2nd task, while gives the least contribution in the 3rd task. This figure shows the necessity of performing channel selection for each mental task, and gives the reason while TSVM-MKL achieve better performance when employ less channels.

Table 2.9: A comparison of TSVM-MKL accuracy with/without channel selection for the three subjects over three consecutive test sessions. (LN) denotes the linear case with no channel selection and (LC) denotes the linear case with channel selection. NumChan/task denotes the number of involved channels per mental task.

Subject	NumChan/task	Session 1	Session 2	Session 3	Average
A (LN)	8-8-8	65.6	74.4	76.7	72.2
A (LC)	7-8-8	67.4	74.9	78.0	**73.5**
B (LN)	8-8-8	56.2	56.3	61.8	58.1
B (LC)	7-8-5	57.8	59.0	65.4	**60.8**
C (LN)	8-8-8	46.3	49.8	48.6	48.3
C (LC)	7-7-8	50.7	54.5	45.4	**50.2**

2.7 Conclusions

In this chapter, the learning problem is regarded as a labeling process for unlabeled data in the framework of semi-supervised learning. It aims at reducing the calibration procedure in BCI applications. For this

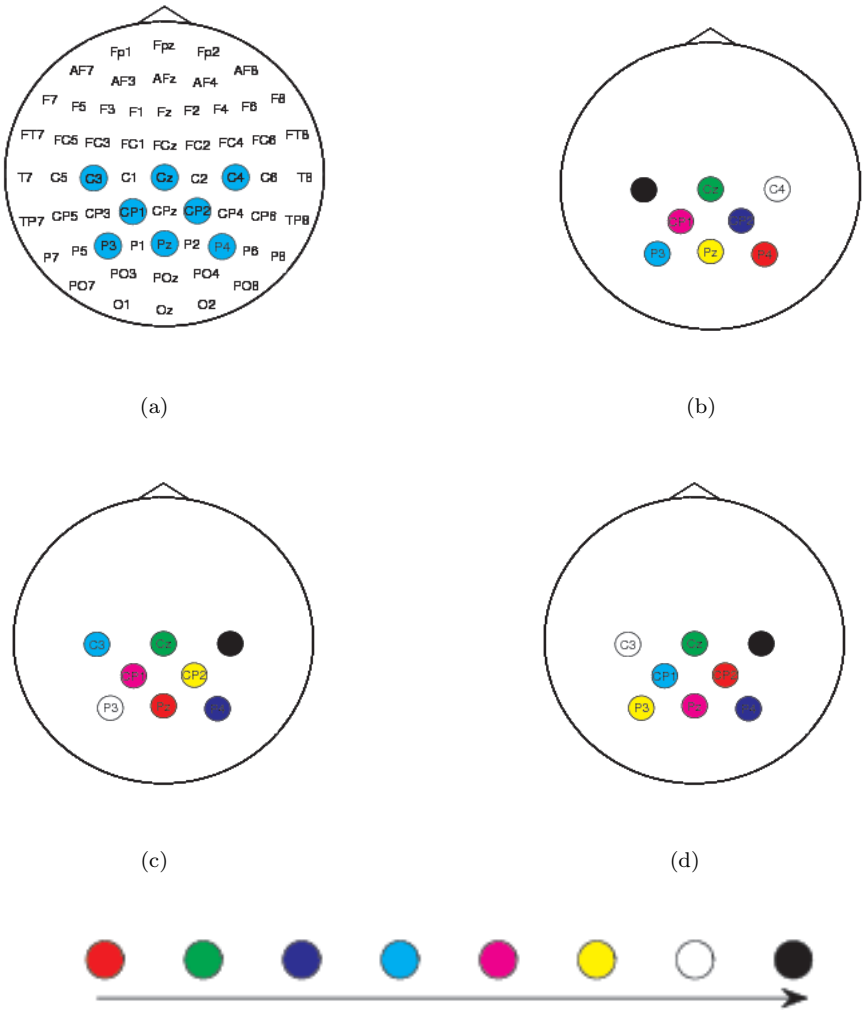


Figure 2.10: The distribution of involved channels. 2.10(a) The channels involved in the experiments without channel selection. 2.10(b) The channels for the 1st task. Below: 2.10(d) The channels for the 2nd task. 2.10(c) The channels for the 3rd task. For the experiments without channel selection, as shown in the first figure, they utilize the information from all available channels equally. For our experiments with channel selection by TSVM-MKL, each channel make different contributions to the learning task. We first sort the weights of all involved channels in descending order. And then, the "importance-degree" for each channel is obtained by its ranking. Here we use different color to denote the importance-degree: changing the color from red to black with the changes from "the most important" to "the less important".

sake, we present a kernel design algorithm and experimental results on the benchmark data sets for semi-supervised learning and BCI data sets show the effectiveness.

The common point of SSL algorithms is the fact that they always rely on some model assumptions. Cluster assumption and manifold assumption are two of the most important ones, and they have been proved to be effective in many real world applications. However, for the complex problems such as BCI data analysis, it is difficult to determine which kind of assumption is better and when one assumption should be preferred over the other. In this chapter, we propose TSVM-MKL to realize an algorithm that can self-adapt to the applications. It exploits manifold information in the framework of TSVM, and preference of assumption types can be determined automatically by means of the kernel selection in the

MKL part.

As the TSVM-MKL inherits the non-convex and non-smoothy property of TSVM, we employ the DC programming to circumvent this shortcoming. By decomposing the non-convex problem into two convex problems, the original problem turns out to solve iteratively a fully supervised multiple kernel SVM with additional balancing constraint. As TSVM-MKL does not require a tedious search of kernel parameters as in TSVM, its complexity of computation does not increase obviously compared with the supervised convex MKL.

Experimental results on benchmark data sets of semi-supervised learning showed improvements of the proposed TSVM-MKL compared with the single-assumption based SSL algorithms. We test it on two types of BCI systems. For the CSP feature based system, it achieves better classification accuracy in the precondition of calculating all the features in a batch mode. For the PSD feature based system which does not require labels information in the feature extraction procedure, it achieves obvious improvements compared with the TSVM and Laplacian SVM algorithms. As an important advantage of multiple kernel learning, it implements the channel selection synchronously in the learning process of the classifier model. Such characteristic gives the TSVM-MKL a better performance when employ less channels.

In the chapter, we solely consider MKL in the framework of kernel selection as the constraints on coefficients d_k can be seen as a ℓ_1 constraint. Future work could be exploring non-sparse regularization that is a ℓ_p constraint with $1 < p < \infty$ as in fully supervised approach. And the necessity of realizing the MKL algorithm in a online learning mode. In Chapter 3, we will explore the online strategy of MKL algorithms with $1 < p < \infty$ in supervised learning.

Chapter 3

Online multi-kernel learning: LaMKL

Contents

Kernel methods combined with large margin principle have gained a overwhelming success over the past decade. Many reasons support this success and among them one can refer to the availability of efficient solvers to handle many learning problems, and the ability to lift most of existing linear methods to non-linear cases using the famous kernel trick [Schölkopf 2002]. The kernel acts as an implicit representation of the data, and its appropriate choice will condition the learning algorithm's performance. Therefore, this raises the question of designing adequate kernel for a problem at hand.

Many attempts have been made in the past in order to address this issue and our description of existing work is far from being exhaustive. Learning the kernel was viewed as a problem of finding the right hyper-parameters of the kernels. Generally, it is implemented through the optimization, using gradient descent technique, of a proxy of generalization error [Chapelle 2002]. Another trend of methods seeks the direct learning of kernel matrix by looking for the best kernel matrix maximally aligned with the target output kernel matrix [Cristianini 2001] or the kernel matrix optimizing the dual objective function [Lanckriet 2004b] subject to constraints as positive definiteness or trace constraint. Following the latter idea, Lanckriet et al. [Lanckriet 2004a] had proposed to restrict the searching space of the kernel matrix and considered the sought kernel matrix as a linear combination of existing kernel matrices under the same regularity conditions. This formulation opens the way to multiple kernel learning (MKL) algorithms. Interestingly, learning the kernel under this framework can be seen as an extended feature selection procedure. Indeed, the different kernel matrices to be combined correspond to different types of information sources (features or groups of features, handcrafted kernels traducing the similarity between samples based on domain knowledge, ...) one desires to bind altogether in order to achieve good generalization performances. Notice that we have considered such a procedure in chapter 2 to select the appropriate data information (manifold or cluster information) for semi-supervised learning.

Existing MKL algorithms differ according to the way of interpreting kernel combination and according to the optimization method deployed to address the mathematical problem. One way to specify the type of combination is regularization over the weights of linear combination. For instance to attain a sparse feature selection, a ℓ_1-norm regularization (or constraint) was embedded into the learning problem and a flurry of methods were designed to solve such a problem [Lanckriet 2004a, Bach 2004, Sonnenburg 2006, Rakotomamonjy 2008a, Chapelle 2008a]. Those algorithms essentially alternate between solving a learning

problem (typically SVM) for the kernels weights fixed and updating those weights given the solution of the SVM until a termination criterion is met. Although sparse combination of the kernels proves efficient in practice, it reveals useless in the context where the features at hand should not be discarded but combined in a non-sparse way. The rationale behind this idea is for instance applications where different types of features are specifically tailored according to some a priori knowledge and those features have exhibited their usefulness. Therefore, some authors has considered a general ℓ_p-norm (with $p > 1$) regularization on the kernel weights with companion optimization algorithms [Vishwanathan 2010, Kloft 2011, Kloft 2009, Cortes 2009b]. These algorithms are based on Newton's optimization method or cutting planes approach [Kloft 2009], interleaving of a SVM problem and kernel weights update [Kloft 2011] or SMO[1] procedure [Vishwanathan 2010]. Finally another trend of research departs from the main streamline of linear combination of the kernels and envisions non-linear combination. To name a few, we can cite polynomial kernel combination [Cortes 2009b], localized kernel combination [Gonen 2008] or hierarchically structured kernels [Bach 2008].

Most of the described algorithms (implementing linear or non-linear combination of kernels) are all batch methods and require to handle simultaneously all the training samples. A general drawback of these MKL strategies is the high computational cost during training, which prevents their application to large scale problems. Online learning achieves significant computational advantages over batch learning algorithms, and the benefits become more evident when dealing with streaming or large scale data. Many real life machine learning problems can be more naturally viewed as online rather than batch learning problems. Take the BCI application as an example, the EEG data is often collected continuously in time. More importantly, the concepts to be learned may also evolve in time. Therefore it is desirable to have a MKL algorithm working in online fashion similarly to existing online learning procedures as kernel perceptron or variants, incremental SVM [Cauwenberghs 2001, Ma 2003], Pegasos [Shalev-Shwartz 2011] or online SVM in the dual (LASVM) [Bordes 2005]. [Jin 2010] introduced a termed online MKL that aims to learn a kernel based prediction function from a pool of predefined kernels in an online fashion. [Martins 2010] proposed a new family of online proximal algorithms for MKL (as well as for group Lasso and variants thereof) for the structured output case, which involves repeatedly solving a batch learning problem. [Orabona 2010] presented an online batch strongly convex MKL algorithm with a ℓ_p-norm regularization. All these online MKLs operate in the primal and hence induce a decision function which can lack sparsity in terms of the parameters of the SVM model. Hence, in this chapter, we consider solving a particular kind of MKL, namely ℓ_p-norm MKL, in online way based on the dual objective function similarly to LASVM. This choice is motivated by the fact that among online single kernel learning procedure, LASVM has shown efficiency both in terms of computation time and generalization property. Our new algorithm called LaMKL processes the samples on the fly using the SMO strategy: after seeing the new sample, it is added to the current set of support vectors and the parameters of the decision function along with the kernel weights are updated accordingly. This procedure is repeated until the overall data set was swept.

The remainder of the chapter is organized as follows: Section 3.1 reviews the work on batch multiple kernel learning and details the peculiarities of the most representative algorithms. Section 3.2 emphasizes on the ℓ_p-norm multiple kernel learning and we will present the SMO methodology used to solve it. From this point, we will derive our online procedure in Section 3.3. Experimental evaluation is implemented in Section 3.4 and the chapter is ended up with some conclusions and forthcoming work on unaddressed issues.

3.1 Multiple Kernel Learning Framework

Without loss of generality, let assume a binary classification problem knowing that the main features of our development can be adapted more or less straightforwardly to other learning problems as multi-class classification, regression or even unsupervised learning. Let consider a set of data $\{(\mathbf{x}_i, y_i) \in$

[1] Sequential minimal optimization

$\mathcal{X} \times \{-1, 1\}\}_{i=1}^n$ and assume we are looking for a decision function f optimizing the following problem

$$\min_f \frac{1}{2}\|f\|_{\mathcal{H}}^2 + C\sum_{i=1}^n V(f(\mathbf{x}_i), y_i).$$

Here $f(\mathbf{x}) = f_0(\mathbf{x}) + b$ and f_0 is assumed to belong to a Reproducing Kernel Hilbert Space (RKHS) induced by a kernel κ; C is the regularization parameter and V represents the hinge loss function. According to our developments in Chapter 1, we know that the decision function is given by

$$f(\mathbf{x}) = \sum_{i=1}^n \alpha_i y_i \kappa(\mathbf{x}_i, \mathbf{x}) + b$$

with the parameters α_i solution of the dual problem

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \mathbf{e}^\top \boldsymbol{\alpha} - \tfrac{1}{2}\boldsymbol{\alpha}^\top \mathbf{Y}^\top \mathbf{K}\mathbf{Y}\boldsymbol{\alpha} \\ \text{subject to} \quad & \boldsymbol{\alpha}^\top \mathbf{y} = 0 \\ & 0 \leq \alpha_i \leq C \qquad \forall i = 1, \cdots, n \end{aligned}$$

$\boldsymbol{\alpha} \in \mathbb{R}^n$ represents the vector of Lagrange parameters, $\mathbf{y} = \begin{bmatrix} y_1 & \dots & y_n \end{bmatrix}^\top$, $\mathbf{K} \in \mathbb{R}^{n \times n}$ the kernel matrix, $\mathbf{e}$ a vector with all entries equal to one and $\mathbf{Y}$ a diagonal matrix with entries y_i. Beyond the optimization of $\boldsymbol{\alpha}$, the most involved problem is the assessment of the kernel or equivalently the learning of $\mathbf{K}$. A general kernel learning problem is to search for a kernel matrix in a subset $\mathcal{K}$ of semi-definite positive kernels, that is solving the following min-max problem

$$\begin{aligned} \min_{\mathbf{K} \in \mathcal{K}} \max_{\boldsymbol{\alpha}} \quad & \mathbf{e}^\top \boldsymbol{\alpha} - \tfrac{1}{2}\boldsymbol{\alpha}^\top \mathbf{Y}^\top \mathbf{K}\mathbf{Y}\boldsymbol{\alpha} \\ \text{subject to} \quad & \boldsymbol{\alpha}^\top \mathbf{y} = 0 \\ & 0 \leq \alpha_i \leq C \qquad \forall i = 1, \cdots, n. \end{aligned} \tag{3.1}$$

Existing kernel learning methods roughly differ according to the searching space of $\mathcal{K}$. Issued from the work of Lanckriet et al. [Lanckriet 2004a], Multiple Kernel Learning algorithms consider the searching space as a combination of basic kernel matrices. Usually one can distinguish two cases: linear combination and non-linear combination of kernels. We review in the upcoming subsections these two groups of MKL algorithms.

3.1.1 Linear combination based MKL

Most popular MKL methods specify the kernel space as the non-negative combination of kernel matrices. The specification of the basis kernel matrices is user-dependent and should reflect some knowledge or assumptions on the hypothesis space to whom belongs the decision function. Specifically, consider the set $\{\mathbf{K}_k\}_{k=1}^m$ with each $\mathbf{K}_k$ a (semi)-positive definite Gram matrix with associated kernel κ_k and induced RKHS $\mathcal{H}_k$. Moreover let assume those matrices have bounded trace i.e. $trace(\mathbf{K}_k) \leq 1$. Therefore, the learned kernel matrice are searched over the space

$$\mathcal{K} = \left\{ \mathbf{K} \in \mathbb{R}^{n \times n} \,|\, \mathbf{K} = \sum_k^m d_k \mathbf{K}_k \quad \text{with} \quad d_k \geq 0 \quad \forall k = 1, \cdots, m \right\}$$

with d_k representing the weights in the mixture of kernel matrices. By definition, the resulting kernel matrix $\mathbf{K}$ is at least semi-positive definite and allows to avoid any degenerate situation while solving Problem (3.1). Solving $\mathbf{K}$ can thus turned into solving for the kernel weights d_k.

To gain in efficiency and to avoid over-fitting, it is desirable to control the complexity of the weight vector $\mathbf{d}$. As a remedy a regularization term over $\mathbf{d}$ is added to the optimization process leading to the problem

$$\min_{\mathbf{d}\geq\mathbf{0}} \max_{\boldsymbol{\alpha}} \quad \mathbf{e}^\top\boldsymbol{\alpha} - \tfrac{1}{2}\boldsymbol{\alpha}^\top\mathbf{Y}^\top\left(\sum_{k=1}^m d_k\mathbf{K}_k\right)\mathbf{Y}\boldsymbol{\alpha} \tag{3.2a}$$

$$\text{subject to} \quad \boldsymbol{\alpha}^\top\mathbf{y} = 0 \tag{3.2b}$$

$$0 \leq \alpha_i \leq C \qquad \forall i = 1, \cdots, n \tag{3.2c}$$

$$\Omega_{\mathbf{d}}(\mathbf{d}) \leq r \tag{3.2d}$$

where $r > 0$ is a user-defined constant. In most applications, the regularization term $\Omega_{\mathbf{d}}(\mathbf{d})$ is chosen as a convex norm or mixed-norm. For our concern in Chapter 2, we explored the linear kernel combination with the constraint (3.2d) taken as $\|\mathbf{d}\|_1 \leq 1$ (ℓ_1-norm regularization) in the context of semi-supervised learning. Such methods lead to a sparse $\mathbf{d}$ and benefit the corresponding advantages of sparseness if one believes that some of the basic kernels are spurious. SimpleMKL [Rakotomamonjy 2008a] is a representative algorithm in this family. However, sparse MKL may discard some important information, and thus do not always performs well in practice. Recent research to MKL explored the ℓ_p-norm ($p > 1$) regularization $\|\mathbf{d}\|_p \leq r$, which attempts to combine the kernels in a less agressive way especially when the basic kernels are all deemed useful.

An equivalent primal formulation of Problem 3.2 can be written as

$$\min_{f_k\in\mathcal{H}_k, b\in\mathbb{R}, \mathbf{d}\geq\mathbf{0}} \quad \tfrac{1}{2}\sum_{k=1}^m \frac{\|f_k\|^2_{\mathcal{H}_k}}{d_k} + C\sum_{i=1}^n V(f(\mathbf{x}_i), y_i) \tag{3.3a}$$

$$\text{subject to} \quad \Omega_{\mathbf{d}}(\mathbf{d}) \leq r. \tag{3.3b}$$

We will not elaborate on this equivalence here but defer it to Section 3.2. To obtain the dual formulation of linear MKL for convex loss functions other than the hinge loss, we invite the interested reader to consult [Kloft 2011]. The point we want to emphasize on is the convexity of both primal and dual problems for any convex regularization $\Omega_{\mathbf{d}}(\mathbf{d})$. Therefore the power of convex solvers can be deployed to solve the MKL problem. Many algorithms were proposed in literature and we propose hereafter to visit them through three categories[2] : wrapper methods that make call to SVM solver in a inner loop, methods avoiding call to SVM and SMO solution.

Wrapper methods Many efforts was made to break down the complexity of Quadratic Constraint Quadratic Programming (QCQP) method. One approach of solution is represented by wrapper methods that are generically described in Algorithm 4. They alternate between solving the SVM problem using classical solvers and update of the kernel weights. Different algorithms relying on this principle were proposed, and they were based on Semi-Infinite Linear Programming, Newton method, gradient descent method or analytical solution to cite a few.

Algorithm 4 A general wrapper framework for solving linear MKL

Initialize the weights $\mathbf{d}$
repeat
  Solve the dual of SVM for the optimal solution $\boldsymbol{\alpha}$ for $\mathbf{d}$ fixed.
  Update the kernel weights $\mathbf{d}$.
until Convergence

Semi-infinite Linear Programming (SILP) [Sonnenburg 2006] considers a sparse constraint of the form $\|\mathbf{d}\|_1 = 1$. As the coefficients $d_k \geq 0$, this writes $\sum_k d_k = 1$. It is easy to deduce that the objective function in (3.2) becomes $\sum_k^m d_k \left(\mathbf{e}^\top\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^\top\mathbf{Y}^\top\mathbf{K}_k\mathbf{Y}\boldsymbol{\alpha}\right)$ and straightforward to formulate the

[2] For a nice overview of kernel learning problems, algorithms, theoretical guarantees and publicly available softwares, one can refer to http://www.cs.nyu.edu/~mohri/icml2011-tutorial/. Another useful pointer is the review paper [Gönen 2011].

Linear Programming (LP) problem

$$\begin{aligned} \max_{t,\mathbf{d}\geq\mathbf{0},\sum_k d_k=1} \quad & t \\ \text{subject to} \quad & \sum_k^m d_k \left(\mathbf{e}^\top\boldsymbol{\alpha} - \tfrac{1}{2}\boldsymbol{\alpha}^\top\mathbf{Y}^\top\mathbf{K}_k\mathbf{Y}\boldsymbol{\alpha}\right) \leq t \\ & \boldsymbol{\alpha}^\top\mathbf{y} = 0, \quad 0 \leq \alpha_i \leq C, \ \forall i = 1, \cdots, n. \end{aligned}$$

However the LP problem holds for all parameter vectors (possibly infinitely many) $\boldsymbol{\alpha}$ satisfying the box and balancing constraints. Therefore Sonnenburg et al. designed a cutting plane algorithm that updates the kernel weights by solving a LP problem with successive linear inequality constraints based on all $\boldsymbol{\alpha}$ calculated through the iterations of corresponding wrapper optimizer. The algorithm was careful tailored in order to handle large scale data sets.

Reduced gradient mostly known as SimpleMKL [Rakotomamonjy 2008a]. This method is more efficient than solving a series of LP problems and considers a constrained optimization problem over $\mathbf{d}$ that can be solved by a simple gradient descent approach. To do so, the derivative of the objective function (3.2) w.r.t $\mathbf{d}$ is taken and depends on the optimal solution $\boldsymbol{\alpha}$. Hence each trial in direction of search involves solving a SVM which induces a certain computation cost even a warm-restart strategy can be used to speed-up the algorithm.

Newton methods [Chapelle 2008a] improve on SimpleMKL by using a second order information. It was shown that a faster convergence is obtained via Newton method. While SimpleMKl and its variant are limited to ℓ_1-norm regularization, [Kloft 2009] considered a non-sparse MKL problem through the use of ℓ_p-norm regularization, and a Newton descent approach is applied to the kernel weights with some safeguard tricks ensuring non-negativity of the weights. Also in [Kloft 2009], an extension of SILP principle to solve non-sparse MKL is devised and is based on the second order linearization of the regularization constraint on $\mathbf{d}$.

Beyond those methods, [Kloft 2011] have elaborated a wrapper where at each iteration the expression of the kernel weights is obtained analytically and does not require any specific optimization. The approach holds for non-sparse MKL knowing that similar results were described in [Rakotomamonjy 2008a] for sparse MKL.

Non-Wrapper methods To avoid the expensive cost of SVM solver in wrapper methods, [Cortes 2009a] proposed a projected gradient approach. However, we should mention their approach is limited to ℓ_2-norm regularization and is based on quadratic loss function instead of the hinge loss. Another approach is online learning. Instead of solving the dual directly, it addresses a variational formulation of the primal problem (3.3a). The derived algorithms are based on online proximal methods and prove efficient as they do not require storage of the full kernel matrices [Orabona 2010, Martins 2010]. Generally, these algorithms tend to produce a decision function with many support vectors.

SMO methods The most popular SVM solver LibSVM is based on Gauss-Seidel type optimization, namely SMO. Combined with many tricks as kernel caching or shrinking procedure, SMO methods lead to state-of-art solvers in the single kernel case. [Bach 2004] devised a SMO algorithm for sparse MKLs by smoothing the non-differentiability of ℓ_1-norm. Recently [Vishwanathan 2010] proposed a genuine SMO algorithm for non-sparse MKLs by eliminating the weights d_k from the dual (3.2). The reported results proved their ability of handling a large number of samples and kernels. The online method we will propose in the sequel heavily leans on this SMO method.

Up to now, we can summarize the characteristics of linear combination of MKL using SVMs: they can be formulated as convex optimization problems, and appropriate kernels can be selected automatically in the training procedure. In some cases, linear combination of kernels may not be rich enough to contain the optimal kernel. Therefore, we present another group of MKL algorithms, non-linear combination type MKL, in next subsection.

3.1.2 Non-linear combination MKL

Non-linear combinations of kernels have also been considered by some researchers recently [Zhuang 2011, Li 2010a, Varma 2009]. In general, non-linear combination of kernels are non-convex and difficult to solve. Among existing strategies, we present several representative algorithms as follows.

Generalized MKL extends traditional MKL formulations to handle generic kernel combinations subject to general regularization on the kernel parameters. The proposed strategy was implemented in two loops: in the outer loop, the kernel is learnt by optimizing over $\mathbf{d}$ while, in the inner loop, the kernel is held fixed and the SVM parameters are learnt. According to their results, the proposed generalized MKL can achieve the same classification accuracy with normal MKL but using far fewer features [Varma 2009].

Multi-layer MKL extends the optimization domain of κ by adopting a family of deep kernels. A ℓ-level multi-layer kernels is defined as, $\kappa^{(\ell)}(\cdot,\cdot) = g^{(\ell)}(\kappa_1^{(\ell-1)}(\cdot,\cdot),\cdots,\kappa_m^{(\ell-1)}(\cdot,\cdot))$, where $g^{(\ell)}$ is some function to combine the multiple $(\ell-1)$-level kernels that ensures the resulting combination is a valid kernel. The combinations of multiple kernels are thus implemented in a multi-layer structure [Zhuang 2011].

Localized MKL assigns different weights to the kernels in different regions. A locally combined kernel matrix is defined as $\kappa_\eta(\mathbf{x}_i,\mathbf{x}_j) = \sum_{k=1}^m \eta_k(\mathbf{x}_i)\kappa(\mathbf{x}_i,\mathbf{x}_j)\eta_k(\mathbf{x}_j)$. This algorithm is composed of two parts, namely, a gating model assigning weights to kernels for a data instance and a kernel machine with the locally combined kernel matrix. Similar accuracy results were achieved with MKL by storing fewer support vectors [Gonen 2008].

There are also some other non-linear combinations of kernels. [Cortes 2009b] presented a polynomial kernels combination for regression problem. [Li 2010a] gave several new kernel matrices by the Hadamard product of any two kernel matrices computed from original kernels. [Qin 2010] presented a non-linear combination of MKL by introducing the Gompertz model [Wheldon 1998], which is a mathematical model developed especially for time series. Satisfying results were reported, meanwhile, they suffer from the heavy computation burden. In this dissertation, we mainly involve the linear combination of MKL algorithms.

3.2 ℓ_p-norm MKL

Among divers MKL algorithms, we are particularly interested in the ℓ_p-norm MKLs which lead non-sparse combination of kernels with arbitrary norms ($p > 1$). As shown in previous section, such attributes potentially keep all kernels and have been proved that non-sparse MKLs achieve accuracies that surpass the sparse ones [Kloft 2011]. Meanwhile, the ℓ_p-norm MKLs with $p > 1 + \varepsilon$ tend to the sparse kernel combination for particular problems that prefer sparse solutions. Recently, two representative solutions of ℓ_p-norm MKLs were proposed, namely, the ℓ_p-norm multiple kernel learning by [Kloft 2011] and the ℓ_p-norm squared MKL trained by the SMO algorithm (for simplicity, we denote it as SMO-MKL). In this section, we present the SMO-MKL problem as it provides meaningful reference for deriving our online MKL algorithm, LaMKL.

3.2.1 ℓ_p-norm squared MKL formulation

3.2.1.1 Primal problem

The MKL task corresponds to learning a standard SVM with the weighted kernel expansions. Providing a kernel set $\{\kappa_k, k = 1,\cdots,m\}$ inducing RKHSs $\mathcal{H}_k$, we consider the ℓ_p-norm ($p > 1$) squared regularizer

MKL proposed by [Vishwanathan 2010] with the primal

$$\begin{array}{ll} \min\limits_{f_k, d_k \geq 0, \xi_i, b} & \frac{1}{2}\sum_{k=1}^{m} \frac{\|f_k\|^2}{d_k} + C\sum_{i=1}^{n}\xi_i + \frac{\lambda}{2}\left(\sum_{k=1}^{m} d_k^p\right)^{\frac{2}{p}} \\ \text{subject to} & y_i f(\mathbf{x}_i) \geq 1 - \xi_i \qquad i = 1, \cdots, n \\ & \xi_i \geq 0 \qquad i = 1, \cdots, n \end{array} \tag{3.4}$$

where C and λ are regularization parameters and the decision function can be formulated as

$$f(\mathbf{x}) = \sum_{k=1}^{m} f_k(\mathbf{x}) + b. \tag{3.5}$$

3.2.1.2 Optimization strategy

The SMO-MKL executes the SMO optimization sequentially on the dual of (3.4) by eliminating $\mathbf{d}$ from that. For this sake, an intermediate saddle point optimization problem is derived first by minimizing only over f_k, $\forall k$, b and ξ_i, $\forall i$. Then, the kernel weights can be recovered from the dual variables involved in the optimization. Corresponding Lagrangian of (3.4) was introduced as

$$\mathcal{L}(f_k, b, \xi_i) = \frac{1}{2}\sum_{k=1}^{m}\frac{\|f_k\|^2}{d_k} + \sum_{i=1}^{n}(C - \beta_i)\xi_i + \frac{\lambda}{2}\left(\sum_{k=1}^{m} d_k^p\right)^{\frac{2}{p}} - \sum_{i=1}^{n}\alpha_i\left[y_i f(\mathbf{x}_i) - 1 + \xi_i\right] \tag{3.6}$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ being the Lagrange multipliers. Differentiating $\mathcal{L}$ with respect to f_k, b and $\boldsymbol{\xi}$ lead to:

$$\nabla_{f_k}\mathcal{L} = 0 \quad \Rightarrow \quad f_k(\mathbf{x}) = d_k\sum_{i=1}^{n}\alpha_i y_i \kappa_k(\mathbf{x}_i, \mathbf{x}) \tag{3.7a}$$

$$\nabla_b \mathcal{L} = 0 \quad \Rightarrow \quad \sum_{i=1}^{n}\alpha_i y_i = 0 \tag{3.7b}$$

$$\nabla_{\xi_i}\mathcal{L} = 0 \quad \Rightarrow \quad \alpha_i + \beta_i = C, \quad i = 1, \cdots, n. \tag{3.7c}$$

Taking account for $\alpha_i \geq 0$ and $\beta_i \geq 0$, the box constraint can thus be attained for the Lagrangian multipliers, namely, $0 \leq \alpha_i \leq C$ and $0 \leq \beta_i \leq C$. Substituting (3.7) back to (3.6), we get

$$\mathcal{L} = -\frac{1}{2}\sum_{k=1}^{m} d_k \sum_{i,j}^{n}\alpha_i\alpha_j y_i y_j \kappa_k(\mathbf{x}_i, \mathbf{x}_j) + \frac{\lambda}{2}\left(\sum_{k=1}^{m} d_k^p\right)^{\frac{2}{p}} + \sum_{i=1}^{n}\alpha_i,$$

which corresponds to the following half way saddle point problem

$$\min_{\mathbf{d}\geq 0}\ \max_{\boldsymbol{\alpha}\in\mathcal{A}}\ \mathbf{e}^\top\boldsymbol{\alpha} - \frac{1}{2}\sum_{k=1}^{m} d_k \alpha^\top \mathbf{H}_k\boldsymbol{\alpha} + \frac{\lambda}{2}\left(\sum_{k=1}^{m} d_k^p\right)^{\frac{2}{p}} \tag{3.8}$$

where $\mathcal{A} = \{\boldsymbol{\alpha} | 0 \leq \boldsymbol{\alpha} \leq C\mathbf{e},\ \mathbf{y}^\top\boldsymbol{\alpha} = 0\}$, $\mathbf{H}_k = \mathbf{Y}\mathbf{K}_k\mathbf{Y}$ and $\mathbf{Y}$ is a diagonal matrix with labels on the diagonal.

Eliminating d A Lagrangian function is formulated to eliminate $\mathbf{d}$ from (3.8) by taking account for the non-negativity of d_k,

$$\mathcal{L}(d_k) = \mathbf{e}^\top\boldsymbol{\alpha} - \frac{1}{2}\sum_{k=1}^{m} d_k \boldsymbol{\alpha}^\top\mathbf{H}_k\boldsymbol{\alpha} + \frac{\lambda}{2}\left(\sum_{k=1}^{m} d_k^p\right)^{\frac{2}{p}} - \sum_{k=1}^{m}\gamma_k d_k \tag{3.9}$$

where $\gamma_k \geq 0$. Taking its derivatives and let $\nabla_{d_k}\mathcal{L} = 0$, we get

$$\begin{aligned}
\Rightarrow \lambda \left(\sum_{k=1}^{m} d_k^p\right)^{\frac{2}{p}-1} d_k^{p-1} &= \gamma_k + \frac{1}{2}\boldsymbol{\alpha}^\top \mathbf{H}_k \boldsymbol{\alpha} \\
\Rightarrow \lambda \left(\sum_{k=1}^{m} d_k^p\right)^{\frac{2}{p}-1} d_k^{p} &= d_k\left(\gamma_k + \frac{1}{2}\boldsymbol{\alpha}^\top \mathbf{H}_k \boldsymbol{\alpha}\right) \\
\Rightarrow \lambda \left(\sum_{k=1}^{m} d_k^p\right)^{\frac{2}{p}-1} \left(\sum_{k=1}^{m} d_k^p\right) &= \sum_{k=1}^{m} d_k\left(\gamma_k + \frac{1}{2}\boldsymbol{\alpha}^\top \mathbf{H}_k \boldsymbol{\alpha}\right) \\
\Rightarrow \lambda \left(\sum_{k=1}^{m} d_k^p\right)^{\frac{2}{p}} &= \sum_{k=1}^{m} d_k\left(\gamma_k + \frac{1}{2}\boldsymbol{\alpha}^\top \mathbf{H}_k \boldsymbol{\alpha}\right) \\
\lambda \|\mathbf{d}\|_p^2 &= \sum_{k=1}^{m} d_k\left(\gamma_k + \frac{1}{2}\boldsymbol{\alpha}^\top \mathbf{H}_k \boldsymbol{\alpha}\right)
\end{aligned}$$

Therefore (3.9) can be written as

$$\mathcal{L} = \mathbf{e}^\top \boldsymbol{\alpha} - \frac{\lambda}{2}\|\vec{\mathbf{d}}\|_p^2$$

To eliminate the vector $\vec{\mathbf{d}}$ from $\mathcal{L}$, let use Holder's inequality.[3] Recall that $d_k \geq 0$, $\gamma_k \geq 0$ and $\mathbf{H}_k$ being semi-definite positive, we have

$$\sum_{k=1}^{m} d_k\left(\gamma_k + \frac{1}{2}\boldsymbol{\alpha}^\top \mathbf{H}_k \boldsymbol{\alpha}\right) \leq \left(\sum_{k=1}^{m} d_k^p\right)^{\frac{1}{p}} \left[\sum_{k=1}^{m}\left(\gamma_k + \frac{1}{2}\boldsymbol{\alpha}^\top \mathbf{H}_k \boldsymbol{\alpha}\right)^q\right]^{\frac{1}{q}}$$

The optimal solution achieves when it reaches equality in order to minimize $\mathcal{L}$ w.r.t $\boldsymbol{\alpha}$

$$\begin{aligned}
\lambda\|\vec{\mathbf{d}}\|_p^2 = \sum_{k=1}^{m} d_k\left(\gamma_k + \frac{1}{2}\boldsymbol{\alpha}^\top \mathbf{H}_k \boldsymbol{\alpha}\right) &= \left(\sum_{k=1}^{m} d_k^p\right)^{\frac{1}{p}} \left[\sum_{k=1}^{m}\left(\gamma_k + \frac{1}{2}\boldsymbol{\alpha}^\top \mathbf{H}_k \boldsymbol{\alpha}\right)^q\right]^{\frac{1}{q}} \\
\Rightarrow \lambda\|\vec{\mathbf{d}}\|_p^2 &= \|\vec{\mathbf{d}}\|_p \cdot \left[\sum_{k=1}^{m}\left(\gamma_k + \frac{1}{2}\boldsymbol{\alpha}^\top \mathbf{H}_k \boldsymbol{\alpha}\right)^q\right]^{\frac{1}{q}}
\end{aligned}$$

Finally, we get easily from the latter relation the expression

$$\|\vec{\mathbf{d}}\|_p^2 = \frac{1}{\lambda^2}\left[\sum_{k=1}^{m}\left(\gamma_k + \frac{1}{2}\boldsymbol{\alpha}^\top \mathbf{H}_k \boldsymbol{\alpha}\right)^q\right]^{\frac{2}{q}} \tag{3.10}$$

Substituting (3.10) back to the Lagrangian we obtain

$$\mathcal{L} = \mathbf{e}^\top \boldsymbol{\alpha} - \frac{1}{2\lambda}\left[\sum_{k=1}^{m}\left(\gamma_k + \frac{1}{2}\boldsymbol{\alpha}^\top \mathbf{H}_k \boldsymbol{\alpha}\right)^q\right]^{\frac{2}{q}}$$

This dual function has to be maximized w.r.t γ_k. However, due to the semi-definite positiveness of $\mathbf{H}_k$, that is $\boldsymbol{\alpha}^\top \mathbf{H}_k \boldsymbol{\alpha} \geq 0$ and $\gamma_k \geq 0$, the optimal value achieves when $\gamma_k = 0$ and $\mathbf{d}$ was eliminated from the ℓ_p-norm squared MKL dual. For simplicity, the dual is reformulated as follows:

$$D = \mathbf{e}^\top \boldsymbol{\alpha} - \frac{1}{8\lambda}\left[\sum_{k=1}^{m}\left(\boldsymbol{\alpha}^\top \mathbf{H}_k \boldsymbol{\alpha}\right)^q\right]^{\frac{2}{q}} \tag{3.11}$$

[3] Employ the definition of ℓ_p-norm, $\|\mathbf{x}\|_p = (\sum_i \mathbf{x}_i^p)^{\frac{1}{p}}$ and Holder's inequality $\sum_i |\mathbf{x}_i \mathbf{z}_i| \leq \|\mathbf{x}\|_p \|\mathbf{z}\|_q$ where $p \geq 1$, $q < \infty$ and $\frac{1}{p} + \frac{1}{q} = 1$.

Retrieving d Keep in mind that the optimality achieves when (3.10) holds with $\gamma_k = 0$, namely,

$$\lambda \|\vec{\mathbf{d}}\|_p^{2-p} d_k^{p-1} = \frac{1}{2}\boldsymbol{\alpha}^\top \mathbf{H}_k \boldsymbol{\alpha}$$

$$\Rightarrow d_k^{p-1} = \frac{1}{2\lambda} \cdot \frac{\boldsymbol{\alpha}^\top \mathbf{H}_k \boldsymbol{\alpha}}{\|\vec{\mathbf{d}}\|_p^{2-p}} = \frac{1}{2\lambda} \cdot \frac{\boldsymbol{\alpha}^\top \mathbf{H}_k \boldsymbol{\alpha}}{\left[\frac{1}{2\lambda}\left(\sum_{k=1}^m (\boldsymbol{\alpha}^\top \mathbf{H}_k \boldsymbol{\alpha})^q\right)^{\frac{1}{q}}\right]^{2-p}}$$

$$= \frac{\boldsymbol{\alpha}^\top \mathbf{H}_k \boldsymbol{\alpha}}{2\lambda \cdot \frac{\left[\left(\sum_{k=1}^m (\boldsymbol{\alpha}^\top \mathbf{H}_k \boldsymbol{\alpha})^q\right)^{\frac{1}{q}}\right]^{2-p}}{(2\lambda)^{2-p}}} = \frac{\boldsymbol{\alpha}^\top \mathbf{H}_k \boldsymbol{\alpha}}{(2\lambda)^{p-1}\left[\sum_{k=1}^m (\boldsymbol{\alpha}^\top \mathbf{H}_k \boldsymbol{\alpha})^q\right]^{\frac{2-p}{q}}}$$

$$\Rightarrow d_k = \frac{(\boldsymbol{\alpha}^\top \mathbf{H}_k \boldsymbol{\alpha})^{\frac{1}{p-1}}}{2\lambda\left[\sum_{k=1}^m (\boldsymbol{\alpha}^\top \mathbf{H}_k \boldsymbol{\alpha})^q\right]^{\frac{2-p}{q(p-1)}}}$$

Recalling $\frac{1}{p} + \frac{1}{q} = 1$, we thus get $\frac{1}{p-1} = \frac{q}{p}$ and $\frac{2-p}{q(p-1)} = \frac{1}{p} - \frac{1}{q}$. Finally, the kernel weights can thus be recovered hereafter:

$$d_k = \frac{1}{2\lambda}\left[\sum_{k=1}^m (\boldsymbol{\alpha}^\top \mathbf{H}_k \boldsymbol{\alpha})^q\right]^{\frac{1}{q}-\frac{1}{p}} \left(\boldsymbol{\alpha}^\top \mathbf{H}_k \boldsymbol{\alpha}\right)^{\frac{q}{p}}. \tag{3.12}$$

Until now, the MKL problem has been somehow formulated as a SVM problem independent of $\vec{\mathbf{d}}$. In next subsection, we present the implementation of solving the dual (3.11) by the SMO strategy.

3.2.2 MKL solver: SMO-MKL

3.2.2.1 Sequential Minimal Optimization (SMO)

The idea of SMO-type strategy is to break down a large optimization problem into a series of smaller sub-problems. The variables involved in the sub-problem are called **working set** and denoted as $B \subset \{1, \cdots, n\}$. Similarly, we denote $N = \{1, \cdots, n\} \backslash B$ and $\boldsymbol{\alpha}_B$ and $\boldsymbol{\alpha}_N$ to be the sub-vectors of $\boldsymbol{\alpha}$ corresponding to B and N, respectively. When B is restricted to have two variables (assumed to be α_i and α_j), it is the famous SMO problem. The solution of coordinate wise optimization problem can thus be updated

$$\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha} + \triangle \tag{3.13}$$

where $\triangle = [0 \cdots \triangle_i \cdots \triangle_j \cdots 0]^\top$. Thus, the following task is to select these coordinates, namely, the working set selection problem. General approach of working set selection employs the first order or the second order information for training SVMs [Fan 2005]. A popular way to select the working set B is via the "maximal violating pair" which involves the gradient information of the dual objective function. Another manner is to select the B according to the second information to increase the dual objective value towards its optimal value. Finally, the original optimization problem can be solved repeatedly through the two-variable subproblem while holding all other variables in N constant.

3.2.2.2 Sub-optimization problem of ℓ_p-norm MKL

To apply the SMO-type solver on the MKL problem, we reformulate the dual of ℓ_p-norm MKL (3.11) as:

$$D = \begin{bmatrix}\mathbf{y}_B^\top & \mathbf{y}_N^\top\end{bmatrix}\begin{pmatrix}\boldsymbol{\alpha}_B \\ \boldsymbol{\alpha}_N\end{pmatrix} - \frac{1}{8\lambda}\left[\sum_{k=1}^m \left(\begin{bmatrix}\boldsymbol{\alpha}_B^\top & \boldsymbol{\alpha}_N^\top\end{bmatrix}\begin{pmatrix}\mathbf{H}_k^{BB} & \mathbf{H}_k^{BN} \\ \mathbf{H}_k^{NB} & \mathbf{H}_k^{NN}\end{pmatrix}\begin{pmatrix}\boldsymbol{\alpha}_B \\ \boldsymbol{\alpha}_N\end{pmatrix}\right)^q\right]^{\frac{2}{q}}$$

$$= \mathbf{y}_B^\top \boldsymbol{\alpha}_B + constant_1 - \frac{1}{8\lambda}\left[\sum_{k=1}^m \left(\boldsymbol{\alpha}^\top \mathbf{H}_k^{BB} \boldsymbol{\alpha_B} + \boldsymbol{\alpha}_N^\top \mathbf{H}_k^{BB} \boldsymbol{\alpha}_B + constant_2\right)^q\right]^{\frac{2}{q}}$$

where $B = \{i, j\}$ denotes the working set and involves only two variables. $\begin{pmatrix} \mathbf{H}_k^{BB} & \mathbf{H}_k^{BN} \\ \mathbf{H}_k^{NB} & \mathbf{H}_k^{NN} \end{pmatrix}$ is permutation of $\mathbf{H}$, and the superscript of $\mathbf{H}$ indicates the sub-matrix corresponding N and B. Considering the update strategy of $\boldsymbol{\alpha}$ (3.13) and the fact that $\boldsymbol{\alpha}_B = [\alpha_i \ \alpha_j]^\top$, we simplify the update as:

$$\begin{aligned} \alpha_i &\leftarrow \alpha_i + \triangle \\ \alpha_j &\leftarrow \alpha_j + s\triangle \end{aligned}$$

where $s = -y_i y_j$. The update rule of α_j (involving s) and α_i guarantees that the constraint $\mathbf{y}^\top \boldsymbol{\alpha} = 0$ holds after update. Finally, the dual optimization can be transformed to the following reduced optimization:

$$\max_{LB \le \triangle \le UB} \ D_r(\triangle) = (1+s)\triangle - \frac{1}{8\lambda}\left[\sum_k (a_k \triangle^2 + 2b_k \triangle + c_k)^q\right]^{\frac{2}{q}} \tag{3.14}$$

where

$$a_k = \mathbf{H}_k^{ii} + \mathbf{H}_k^{jj} + 2s\mathbf{H}_k^{ij} \tag{3.15a}$$
$$b_k = \boldsymbol{\alpha}^\top(\mathbf{H}_k^{i} + s\mathbf{H}_k^{j}) \tag{3.15b}$$
$$c_k = \boldsymbol{\alpha}^\top \mathbf{H}_k \boldsymbol{\alpha} \tag{3.15c}$$

The superscripts of $\mathbf{H}_k$ indicate specified elements in the matrix according to a matlab coding way. Recall the constraint $0 \le \alpha_i \le C$, namely, $0 \le \alpha_i + \triangle \le C$ and $0 \le \alpha_j + s\triangle \le C$. The lower bound LB and the upper bound UB for $\triangle$ can be calculated:

$$UB = \begin{cases} \min(C - \alpha_i, C - \alpha_j) & \text{when } s = +1 \\ \min(C - \alpha_i, \alpha_j) & \text{when } s = -1 \end{cases}$$

and

$$LB = \begin{cases} \max(-\alpha_i, -\alpha_j) & \text{when } s = +1 \\ \max(-\alpha_i, \alpha_j - C) & \text{when } s = -1. \end{cases}$$

Note that the sub-optimization problem is concave on $\triangle$ with the bounds LB and UB. Various solutions exist for such one dimensional problem.

3.2.2.3 Working set selection

To guide the dual objective function in an ascending direction, it is essential to select the working set B. Before detail, we first define two sets $I_{up} \subseteq \{1, \cdots, n\}$ and $I_{down} \subseteq \{1, \cdots, n\}$ as follows:

$$\begin{aligned} I_{up} \equiv \quad & \{i | \alpha_i < C, y_i = 1 \text{ or } \alpha_i > 0, y_i = -1\} \\ I_{down} \equiv \quad & \{i | \alpha_i < C, y_i = -1 \text{ or } \alpha_i > 0, y_i = 1\}. \end{aligned} \tag{3.16}$$

Following the same argument stated by [Fan 2005], $\boldsymbol{\alpha}$ is stationary solution of (3.11) if and only if there is a number b_w and two non-negative vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ such that:

$$\begin{aligned} \nabla_{\boldsymbol{\alpha}} D + b_w \mathbf{y} &= \boldsymbol{\lambda} - \boldsymbol{\mu} \\ \lambda_i \alpha_i = 0, \quad \mu_i(C - \alpha_i) = 0, \quad \lambda_i \ge 0, \ \mu_i \ge 0, \forall_i, b_w \boldsymbol{\alpha}^\top \mathbf{y} &= 0 \end{aligned} \tag{3.17}$$

where $\nabla_{\boldsymbol{\alpha}} D$ is the gradient of dual:

$$\nabla_{\boldsymbol{\alpha}} D = \mathbf{e} - \sum_k d_k \mathbf{H}_k \boldsymbol{\alpha} = \mathbf{e} - \mathbf{H}\boldsymbol{\alpha} \tag{3.18}$$

Here, $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$ and b_w are Lagrange multipliers associated to the constraints over $\boldsymbol{\alpha}$ expressed in $\mathcal{A} = \{\boldsymbol{\alpha}|\boldsymbol{\alpha}^\top \mathbf{y} = 0, 0 \leq \boldsymbol{\alpha} \leq C\mathbf{e}\}$. The condition (3.17) can be reformulated as

$$\begin{aligned} \nabla_{\boldsymbol{\alpha}} D_i + b_w y_i \geq 0, \quad \text{if} \quad \alpha_i < C \\ \nabla_{\boldsymbol{\alpha}} D_i + b_w y_i \leq 0, \quad \text{if} \quad \alpha_i > 0 \end{aligned} \tag{3.19}$$

where $\nabla_{\boldsymbol{\alpha}} D_i$ is the gradient of (3.18) taken at the index i. Considering $y_i = \pm 1$ and the definition of I_{up} and I_{down}, (3.19) can be rewritten as:

$$\begin{aligned} -y_i \nabla_{\boldsymbol{\alpha}} D_i \leq b_w, \quad \text{if} \quad i \in I_{up} \\ -y_i \nabla_{\boldsymbol{\alpha}} D_i \geq b_w, \quad \text{if} \quad i \in I_{down} \end{aligned} \tag{3.20}$$

Hence, $\boldsymbol{\alpha}$ is stationary point if and only if:

$$\max_{i \in I_{up}} y_i \nabla_{\boldsymbol{\alpha}} D_i \leq \min_{i \in I_{down}} y_i \nabla_{\boldsymbol{\alpha}} D_i \tag{3.21}$$

To the contrary, $\{i, j\}$ is considered to be the violating pair when:

$$y_i \nabla_{\boldsymbol{\alpha}} D_i \geq y_j \nabla_{\boldsymbol{\alpha}} D_j \tag{3.22}$$

where $i \in I_{up}$ and $j \in I_{down}$. When selecting the working set B to be the $\{i, j\}$ pair that most violates the stationary condition, it is the "maximal violating pair" strategy.

Now, we attain all the elements for solving ℓ_p-norm MKL optimization problem by SMO, and they are summarized in Algorithm 5.

Algorithm 5 SMO-MKL: batch learning algorithm on ℓ_p-norm MKL

Set an initial estimation $0 \leftarrow \boldsymbol{\alpha}$.
repeat
  Select the most violating pair $\{i, j\}$ by (3.22).
  Solve the reduced variable optimization problem (3.14).
  Update the $\boldsymbol{\alpha}$ by (3.13) and the weights d_k by (3.12).
until Fulfill stopping criteria.

3.2.2.4 Estimation of b and stopping criterion

At the end of each sub-optimization process, the stopping criterion shall be verified and the bias b is updated accordingly. To easy the presentation, we denote $g_i = y_i \nabla_{\boldsymbol{\alpha}} D_i$ ($i \in \mathcal{A}$), $g_{\max}$ and $g_{\min}$ as its maximum and minimum value, the new b can be obtained as follows

$$b = \frac{g_{\max} + g_{\min}}{2}.$$

Several strategies can be adopted as the stopping criterion, possibilities can be the variation of the dual problem (3.11), the duality gap and the difference between the maximum value of g_i ($i \in I_{up}$) and the minimum value of g_i ($i \in I_{down}$). We terminate the algorithm when any of them falls into a pre-specified threshold.

3.3 Online MKL: LaMKL

The solution yielded by SMO-MKL takes the following form owing to equations (3.5) and (3.7a):

$$f(\mathbf{x}) = \sum_k d_k \sum_i \alpha_i y_i \kappa_k(\mathbf{x}_i, \mathbf{x}).$$

This formula exhibits the kernel expansion form of the decision function knowing that only the coefficient α_i corresponding to support vectors are non-null and some weights d_k can be zero according to the desired level of sparsity. Our intention is to propose an algorithm which will track the evolution of kernel expansion coefficients α_i when processing samples in an online way. For this sake, we implement an online version of the dual (3.11) based on the machinery of SMO-MKL. We will require three ingredients to achieve our online SMO-MKL:

- $\mathcal{S}$: indices set of potential support vectors.
- α_i ($i \in \mathcal{S}$): Lagrange multipliers of the potential support vectors.
- g_i ($i \in \mathcal{S}$): weighted gradients for selecting the working set B.

To implement the online update of $f(\mathbf{x})$, we heavily rely on the approach developed in LASVM [Bordes 2005] for online single kernel SVMs. In this section, we employ the so-called PROCESS and REPROCESS procedures to realize an online SMO-MKL algorithm, LaMKL. For a new sample $(\mathbf{x}_h, y_h)$ available at time t, the PROCESS aims at adding the new sample into the potential support vector set $\mathcal{S}$ and performing a direction search to maximize the dual objective function. This operation involves the violating pair where one of which is the new sample $\mathbf{x}_h$ and the other is a sample already in $\mathcal{S}$. This operation can potentially leave other violating pairs in $\mathcal{S}$. To improve the results, we execute a REPROCESS to optimize the most violating pair in $\mathcal{S}$ as well as do one iteration of batch SMO-MKL.

We detail in the next subsections the elements of PROCESS and REPROCESS procedures.

3.3.1 LaMKL PROCESS

LaMKL PROCESS aims at performing a two-variable sub-optimization problem and adding the new coming sample ($\mathbf{x}_h$,y_h) into the potential support vector set. In order to select another variable from $\mathcal{S}$ to form a working set B according to the criteria of violating pair (3.22), it is necessary to estimate the gradient at current sample:

$$g_h = y_h \nabla_{\boldsymbol{\alpha}} D_h = y_h - \sum_k d_k^t \sum_{i \in \mathcal{S}} \alpha_i y_i \kappa_k(\mathbf{x}_i, \mathbf{x}_h) \tag{3.23}$$

where

$$d_k^t = \frac{1}{2\lambda} \left[\sum_k \vec{\boldsymbol{\alpha}}_{\mathcal{S}}^\top \mathbf{H}_k(\mathcal{S},\mathcal{S}) \vec{\boldsymbol{\alpha}}_{\mathcal{S}} \right]^{\frac{1}{q}-\frac{1}{p}} \cdot \left[\vec{\boldsymbol{\alpha}}_{\mathcal{S}}^\top \mathbf{H}_k(\mathcal{S},\mathcal{S}) \vec{\boldsymbol{\alpha}}_{\mathcal{S}} \right]^{\frac{q}{p}} \quad \forall k \tag{3.24}$$

Here, the kernel weight is estimated as in (3.12) but by restricting $\vec{\boldsymbol{\alpha}}$ and $\mathbf{H}_k$ to the support vector set $\mathcal{S}$ at time t. Taking account to the definition of I_{up} and I_{down} (3.16), the second variable can thus be selected:

$$\{i,j\} = \begin{cases} i \leftarrow h, \quad j \leftarrow \underset{s \in I_{down}}{\operatorname{argmin}}\ g_s, & \text{when } y_h = +1 \\ j \leftarrow h, \quad i \leftarrow \underset{s \in I_{up}}{\operatorname{argmax}}\ g_s, & \text{when } y_h = -1. \end{cases} \tag{3.25}$$

Now, we have collected all the elements for performing the two-variable optimization problem, and Algorithm 6 summarizes the implementation of LaMKL PROCESS.

3.3.2 LaMKL REPROCESS

LaMKL REPROCESS performs a two-variable optimization based on the most violating pair in $\mathcal{S}$ to improve the results produced by PROCESS. Different from LaMKL PROCESS, the working set B is selected as the one that mostly violates the stationarity condition (3.22):

$$\{i,j\} = \begin{cases} i \leftarrow \underset{s \in \mathcal{S}}{\operatorname{argmax}}\ g_s & \text{with } s \in I_{up} \\ j \leftarrow \underset{s \in \mathcal{S}}{\operatorname{argmin}}\ g_s & \text{with } s \in I_{down}. \end{cases} \tag{3.26}$$

Algorithm 6 LaMKL PROCESS

Input: sample $(\mathbf{x}_h, y_h)$
if $h \in \mathcal{S}$

- Exit the algorithm

else

- Add indice h into $\mathcal{S}$: $\mathcal{S} \leftarrow \mathcal{S} \cup \{h\}$ and set $\alpha_h \leftarrow 0$.
- Calculate the gradient g_h using (3.23).
- Select a pair $B = \{i \in \mathcal{S}, j \in \mathcal{S}\}$ using (3.25) and set $\gamma = -y_i y_j$. If such a violating pair does not exist, exit the algorithm.
- Solve the reduced variable problem (3.14) for $\triangle$.
- Update $\alpha_i \leftarrow \alpha_i + \triangle$ and $\alpha_j \leftarrow \alpha_j + \gamma\triangle$.
- Update kernel weights d_k^t.
- Update the gradients g_s, $\forall s \in \mathcal{S}$ using relation (3.23).

end

After the sub-optimization problem and update of the parameters, it can happen that some coefficients $\boldsymbol{\alpha}_s = 0$, $s \in \mathcal{S}$ without violating any stationarity condition. We shall remove the non-blatant support vectors to ameliorate the support vector set $\mathcal{S}$. Regarding the violating pair criteria (3.22), we first calculate the extreme value of g_s, namely,

$$\begin{aligned} i &\leftarrow \underset{s\in\mathcal{S}}{\operatorname{argmax}} \; g_s \quad \text{with} \quad s \in I_{up} \\ j &\leftarrow \underset{s\in\mathcal{S}}{\operatorname{argmin}} \; g_s \quad \text{with} \quad s \in I_{down}. \end{aligned}$$

For all $s \in \mathcal{S}$ with $\alpha_s = 0$, the ones that fulfill the following conditions obey the stationary condition (3.21) and will be moved out as non support vectors:

$$\mathcal{S} = \mathcal{S} - \{s\} \text{ when } \begin{cases} y_s = +1 & \text{and} \quad g_s \leq g_j, \\ y_s = -1 & \text{and} \quad g_s \geq g_i. \end{cases} \tag{3.27}$$

Finally the LaMKL REPROCESS can be summarized in algorithm 7.

3.3.3 Online LaMKL

After initialization, online LaMKL alternates PROCESS and REPROCESS to process the new coming samples. We then employ the REPROCESS repeatedly to simplify the kernel expansion. This process is called "finishing". The whole online strategy is summarized in Algorithm 8.

Finishing To improve the convergence of LaMKL to the true solution (the batch SMO-MKL), we also implement a finishing step after the online iterations of LaMKL. The finishing step calls as many times as necessary the REPROCESS procedure to compute a solution converging towards the batch optimal solution. Finishing step is nothing else than repeating the iterations of SMO-MKL; it induces a consequent computational burden and can increase sensibly the overall computation time of LaMKL.

To give a better understanding of the online MKL, let do a few analysis. At time t, if the training samples not seen yet include potential support vectors, these samples will raise a violating pair alarm during PROCESS and will be included in the kernel expansion. Otherwise, they will be skipped by

Algorithm 7 LaMKL REPROCESS

Search the most violating pair in $\mathcal{S}$ by (3.25).

if there is no violating pair

- exit the algorithm

else

- Solve the reduced variable optimization (3.14).
- Update the $\boldsymbol{\alpha}$ by and the weights d_k.
- Update the gradient of dual by (3.23).
- Remove the non-blatant support vectors by (3.27).

end

Algorithm 8 Online LaMKL Algorithm

Intialization:
Seed $\mathcal{S}$ with a few examples of each class.
Set $\vec{\boldsymbol{\alpha}}_{\mathcal{S}} \leftarrow 0$ and compute the initial gradient $\vec{g_s}$
Online iterations:
Repeat a predefined number of epochs

- Pick an example h.
- Run PROCESS.
- Run REPROCESS.
- Update the bias b.

Finishing:
Repeat REPROCESS until stopping criteria meet.

PROCESS until a potential support vector is sampled. REPROCESS adjusts the current expansion in order to satisfy the stationarity condition. If there exists violating pairs in $\mathcal{S}$, they will be found by REPROCESS and optimized accordingly. Notice that different variants of LaMKL can be derived by, for instance, delaying the REPROCESS after having process a certain number of samples. Materials for these variants can be seen in details in [Bordes 2005].

3.4 Numeric evaluation

In this section, we empirically compare the performance of our proposed online MKL algorithm against the batch MKL algorithm with SMO solution (SMO-MKL) [Vishwanathan 2010]. All experiments scripts are written in Matlab code. The two methods benefit a similar Newton-Raphson approach to solve the one-dimensional problem (3.14).

Classification accuracy In the experiments, we put LaMKL and SMO-MKL side by side for a fixed value of the hyper-parameter C ($C = 100$), as this value provides satisfying results. We employed four UCI data sets, namely, Australian, Ionosphere, Liver and Sonar. For each UCI data set we generated kernels in the same way with [Vishwanathan 2010]. Gaussian kernels with ten bandwidths [0.5 1 2 5 7 10 12 15 17 20] were generated for each individual dimension of the feature vector as well

as the full feature vector itself. We also generated polynomial kernels with three degrees [1, 2, 3]. We normalized the data set to be in the interval $[-1, 1]$ before the evaluation process and all kernel matrix were normalized to have unit trace. Regularization parameter λ is set to be 1 un-optimized. Classification accuracy and involved kernels are listed in the following tables by 5-fold cross validation where N denotes the number of points for training set, T denotes the size of testing set, D denotes the dimension of inputs and M is the number of final kernels involved in the experiments. LaMKL is implemented by 5 epochs. Table 3.1 to 3.4 present the mean and standard deviation accuracy performances and the computation time on the four data sets. In the tables, we report the results of LaMKL without finishing (denoted as "online"), LaMKL with finishing (denoted as "finishing") and the batch SMO-MKL algorithms (denoted as "batch").

Table 3.1: Liver: N=276, T=69, D=5, M=91

ℓ_p-norm	Test accuracy %			Time cost (s)		
	online	finishing	batch	online	finishing	batch
1.33	**71.6(0.5)**	71.3(5.2)	71.0(5.1)	**153.8**	622	174.1
1.66	70.2(3.4)	**70.2(2.8)**	69.9(4.7)	**154.9**	629	249.5
2.00	**72.5(3.6)**	72.2(3.6)	69.0(6.6)	**154.1**	628	168.9
2.33	**73.9(4.2)**	73.6(3.5)	67.3(4.9)	**153.6**	623	184.8
2.66	73.3(6.1)	**73.3(5.5)**	**75.9(5.9)**	**153.9**	623	313.9
3.00	69.0(5.8)	**69.0(5.7)**	65.8(3.3)	**152.9**	621	281.1

Table 3.2: Ionosphere: N=280, T=71, D=33, M=442

ℓ_p-norm	Test accuracy %			Time cost (s)		
	online	finishing	batch	online	finishing	batch
1.33	**94.6(3.4)**	94.6(3.4)	90.9(4.6)	624.3	1028	**382.9**
1.66	**94.6(1.6)**	94.6(1.6)	94.0(1.9)	658.2	1138	**141.2**
2.00	93.1(2.8)	**93.2(2.8)**	92.3(4.5)	670.0	1227	**144.1**
2.33	93.7(2.4)	**93.7(2.3)**	93.1(2.1)	688.9	1379	**166.9**
2.66	**95.4(1.6)**	95.4(1.5)	93.4(4.8)	704.1	1565	**187.4**
3.00	92.0(3.4)	92.0(3.4)	**92.9(3.5)**	700.5	1682	**567.4**

Table 3.3: Sonar: N=166, T=42, D=59, M=793

ℓ_p-norm	Test accuracy %			Time cost (s)		
	online	finishing	batch	online	finishing	batch
1.33	83.9(5.1)	83.9(5.1)	**87.8(3.9)**	**272.3**	544	863.8
1.66	**85.9(5.3)**	85.9(5.3)	85.4(6.9)	**281.9**	625	880.2
2.00	90.2(3.8)	**90.3(3.9)**	85.4(3.5)	**278.4**	700	549.4
2.33	**87.8(2.5)**	87.8(2.5)	77.1(3.3)	**283.4**	760	867.5
2.66	89.3(3.7)	**89.3(3.6)**	85.4(3.9)	**292.4**	870	856.7
3.00	**86.3(4.4)**	86.3(4.4)	83.9(6.1)	**287.2**	874	827.9

Result analysis Compared with the batch SMO-MKL algorithm, the proposed LaMKL achieves better results with less computation time in most cases on data Liver, Ionosphere and Sonar. For the data Australian, they have similar results while LaMKL cost less computation time significantly. With the

same experimental setting on the same data set, performance also differs with the values p of ℓ_p-norm regularization on weights d_k. The finishing strategy doesn't improve the performance obviously, while it cost more computation time.

Table 3.4: Australian: N=552, T=138, D=13, M=195

ℓ_p-norm	Test accuracy %			Time cost (s)		
	online	finishing	batch	online	finishing	batch
1.33	59.3(3.7)	59.3(3.6)	**60.3(1.3)**	**2848**	8388	3178
1.66	**58.5(2.4)**	58.3(2.2)	57.3(1.8)	2890	10181	**1908**
2.00	57.8(1.5)	57.6(1.7)	**59.5(2.6)**	**2978**	12108	3464
2.33	56.4(2.5)	56.7(2.4)	**59.3(2.8)**	**2888**	11856	3741
2.66	56.3(2.1)	56.1(2.3)	**59.3(5.6)**	2891	12049	**2786**
3.00	57.9(2.9)	57.7(3.1)	**58.6(3.0)**	**2894**	12090	4393

- Duality gap is defined as the difference between the primal and the dual objective function, that is,

$$\text{DualityGap} = J_{\text{Primal}} - J_{\text{Dual}}$$

  When the duality gap is zero, solution of the primal problem (3.14) is unify to that of the dual problem (3.11). Figure 3.1(a) denotes the duality gap of LaMKL, LaMKL with finishing and SMO-MKL algorithms. We can see that LaMKL can achieve competitive duality gap with SMO-MKL, and the finishing strategy accelerate the convergence obviously in the begining epoch.

- Number of support vectors: Figure 3.1(b) illustrates the final number of support vectors involved in the experiments. As the scale of the Liver being 345, the LaMKLs (with/without finishing) employ almost all the samples in the final decision function. Notice that even for SMO-MKL, almost all the training points are support vectors.

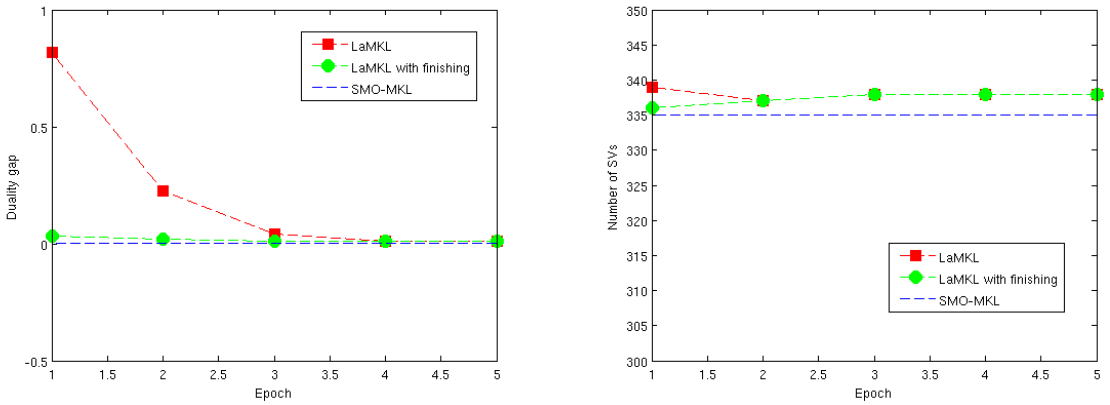
- d_k: Recall that the kernel weight d_k can be retrieved by Lagrangian multipliers $\boldsymbol{\alpha}$ by equation (3.24), we evaluate d_k instead of checking the $\boldsymbol{\alpha}$ along with the five epochs. Because we employ $p = 1.1$, many of the kernel weights will be close to zero. As shown in Figure 3.2, the kernel weight d_k keep stationary over different epochs and achieve close distribution with the batch algorithm.

Effect of different epochs We then check the mean classification accuracies with different epochs. To coincide the reported results with Table 3.1 to 3.4, we report the 5-fold validation errors in Figure 3.3. Generally, more epochs will improve the results, and the classification accuracy tends to be stationary after about three epochs.

3.5 Conclusions and discussions

In this chapter, we present an online solution of ℓ_p-norm multiple kernel learning in the dual by employing the famous SMO optimization procedure. When a new data coming, a PROCESS procedure was firstly introduced to optimize it, and then followed by a REPROCESS to improve the solution obtained by the PROCESS. Experimental results proved that the proposed method can achieve similar performance with the batch learning mode while requiring less computation time.

The LaMKL solution tends to be non-sparse as our LaMKL implementation does not employ any shrinking heuristic during the learning procedure. For the applications that prefer sparse solutions, introducing shrinking into LaMKL could be an interesting strategy to expand the application of LaMKL. In the case of online sparse MKL applications, a similar algorithm called OBSCURE was proposed by



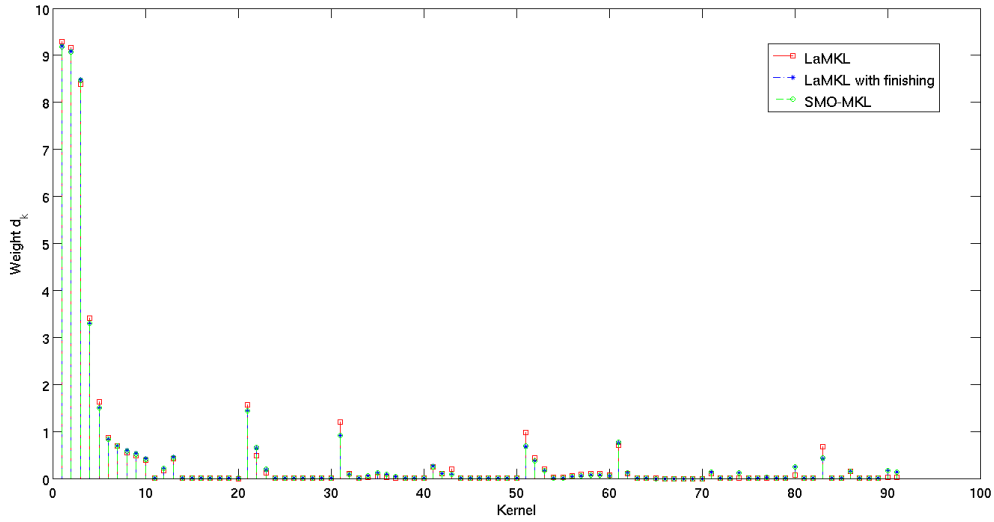
(a) Duality gap

(b) Number of support vectors

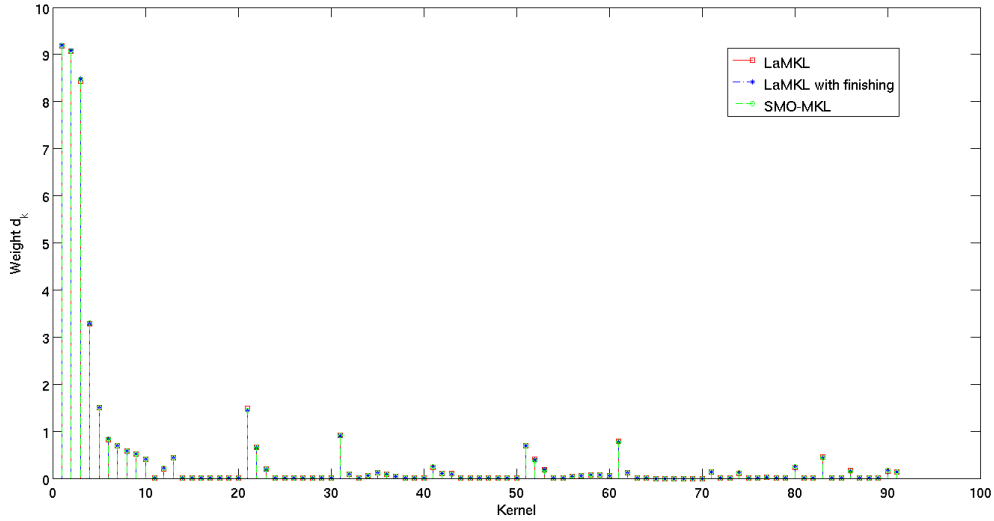
Figure 3.1: Evolution of the LaMKL convergence criteria on data set "Liver" with $p = 1.1$.

introducing a parameter that deciding the level of sparsity of the solution [Orabona 2010]. However, this algorithm was implemented in the primal and was limited to the strongly convex case.

In this chapter, we solely evaluate the ℓ_p-norm MKL on the UCI data sets due to lack of time. As such online multiple kernel learning could be a perfect choice for the time variant systems such as BCIs, the forthcoming work thus include applying LaMKL on the BCI data analysis.

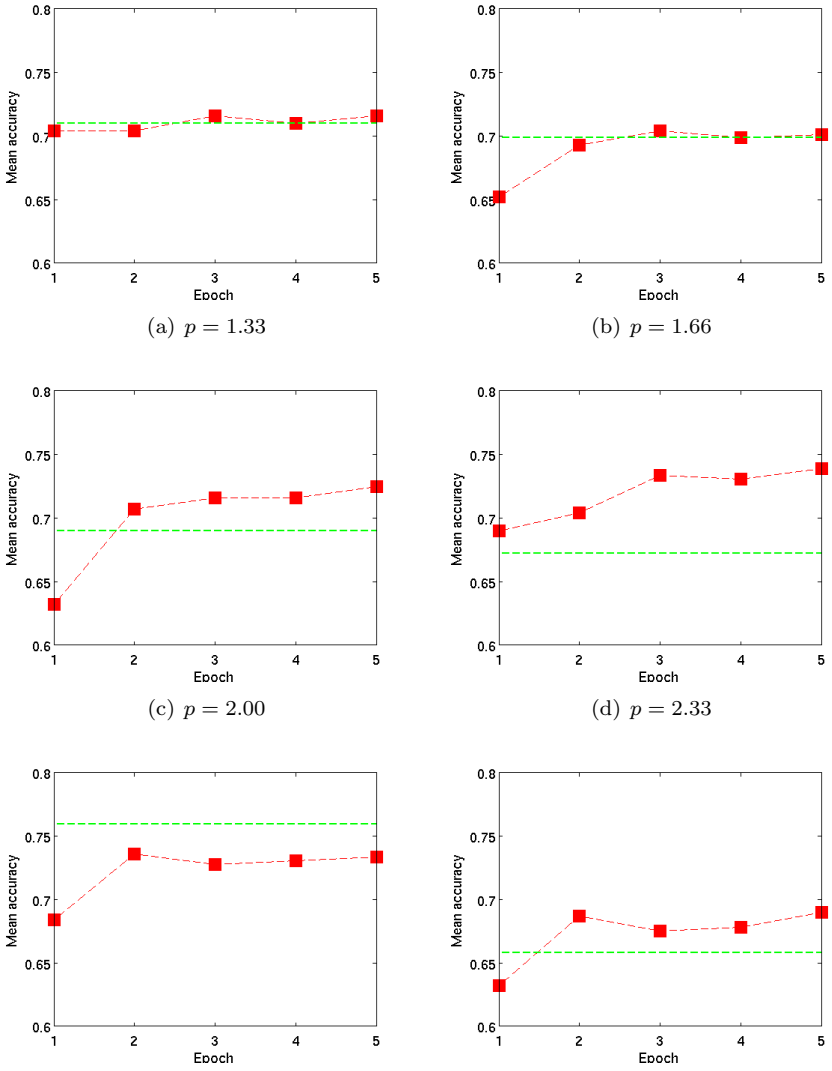


(a) Epoch 1



(b) Epoch 2

Figure 3.2: Evolution of kernel weights on "Liver" with different epoch (when $p = 1.1$). For saving space, we only present the kernel weight distribution of the first two epochs. At the first epoch, the difference between LaMKL (the red one) and SMO-MKL (the green one) is visible on some kernels. For the finishing strategy (the blue one), it achieves similar kernel weights with the batch one. For the second epoch, the online LaMKL and the batch SMO-MKL can be regarded as reaching the same kernel weight for each kernel. Regarding the relationship between d_k and $\boldsymbol{\alpha}$, the LaMKL can be roughly considered to convergence on the solution of batch SMO-MKL.



(a) $p = 1.33$ (b) $p = 1.66$

(c) $p = 2.00$ (d) $p = 2.33$

(e) $p = 2.66$ (f) $p = 3.00$

Figure 3.3: Evolution of the mean classification accuracy of five folds on "Liver" with different epochs. The red ones are the baseline batch results, and the green one are that of LaMKL (without finishing strategy). In most cases, the mean classification accuracy can be improved with the more epochs.

Chapter 4

Beyond complex classifier: how to improve signal processing in BCI?

Contents

As shown in Chapter 1, the signal processing scheme is an essential component in a successful BCI system. Experimental results in Chapter 2 have shown that effective machine learning algorithms can improve the performance of BCI system. In this chapter, we explore the signal processing scheme from two aspects:

- Which one is more important between "good features" and "perfect classifier model"?
- Does user's emotional state affect the performance of a BCI system?

The first exploration was implemented by a BCI competition data analysis, namely, "Mind reading, MLSP 2010 Competition"[1]. The goal of this competition is to select/design a classifier (and any preprocessing system, including a feature extractor) that correctly classifies EEG data into one of two classes [Kenneth 2010]. The contributor that maximizes the area under the ROC curve is considered as the winner. Instead of developing new machine learning algorithms, we aim at evaluating the counterbalance of careful feature extraction (and/or channel extraction) compared with the more complex classifier models. For this sake, we execute careful signal processing methods for feature selection and channel selection. Then, we accurately tuned all the parameters of these preprocessing stage before feeding a classifier. Without using any complex signal processing techniques nor strong neuroscience prior knowledge, we were able to build a simple and fast agnostic classifier which achieves satisfying performance [Labbé 2010]. It is a joint work with the researchers of machine learning team in INSA de Rouen.

[1] http://mlsp2010.conwiz.dk/

The second exploration was implemented by an emotional BCI (e-BCI) system. As claimed by [Calvo 2010], the inextricable link between emotions and cognition inspires researchers in BCI or human computer interface (HCI) to include emotional states in the design of new interfaces: either as evaluation indicators or as components to be inserted in the interface loop [Chanel 2009]. Utilization of the user's emotional state to adapt the BCI classification algorithms is a new trend in BCI research domain[2]. However, we don't find corresponding literature to the best of our knowledge until now. In this part of work, we design a SSVEP based BCI system which accounting for the user's emotional state. This experiment aims at verifying whether the user's emotional state affects the BCI performance or not. For this sake, we present emotional videos before BCI tasks or merge emotional images into the BCI tasks. After acquisition of EEG data with different emotional states (we categorized them to be positive, neutral and negative respectively), we performed two kinds of statistical test, namely the McNemar's test and Wilcoxon signed-rank test to assess whether emotion effect BCI performance or not. Important conclusions on the affectiveness of emotional state on BCI system are attained from perspective of statistical learning finally. The experiments were jointly finished with the Laboratory of Computing and Communication Software, University of Science and Technology of China under the Franco-Chinese Project Xu Guangqi.

4.1 Feature selection VS Classification model

We explore how important the feature (and/or model) selection is in BCI data analysis. Based on the BCI competition: "Mind reading, MLSP 2010 Competition", this section is organized as follows. In Section 4.1.1, we describe the data acquisition of the experiments and the data structure. In Section 4.1.2, signal preprocessing and feature exaction methods are detailed. Final classifier design is presented in Section 4.1.3 and we end up this section with some conclusions in Section 4.1.4.

4.1.1 Experimental setting

Experiemntal data The training data consist of EEG data collected when a subject viewed satellite images that were displayed in the center of an LCD monitor approximately 43 cm in front of them. There are 64 channels of EEG data. The total number of samples is 176378 and the sampling rate is 256 Hz. There are 75 blocks and 2775 total satellite images. Each block contains a total of 37 satellite images, each of which measures 500×500 pixels. All images within a block are displayed for 100 ms and each image is displayed as soon as the previous image is finished. Each block is initiated by the subject after a rest period, the length of which was not specified in advance.

The subject was instructed to fixate on the center of the images and to press the space bar whenever they detected an instance of a target image, where the targets are surface-to-air missile sites. Subjects also needed to press the space bar to initiate a new block and to clear feedback information that was displayed to the subject after each block. P300 was expected to occur whenever the subject detects an instance of a target picture in one of the satellite images. In addition, there is a separate neural signature associated with the pressing of the space bar. This second neural signal occurs around 500-600 ms after the stimulus containing the target is presented. The variables in the training data are defined as follows,

Machine learning task This BCI data analysis can be formulated as a machine learning problem that predicts the class of stimuli images from the raw EEG data. Without loss of generality, we denote the training set of the learning algorithm to be $\{(\mathbf{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y}\}_{i=1}^{n}$, where $\mathbf{x} \in \mathbb{R}^d$ is the EEGs from all or some selected channels corresponding to each statellite images, $y \in \{0, 1\}$ with "0" denotes that the image is non-target and "1" corresponding to target image where a peak shall appear in the P300 base BCI system. Because there are total 2775 satellite images, we thus attained 2775 samples of which 58 belongs to the positive class and 2717 belongs to the negative class.

Test data are released by the organizers after submission of competitors' algorithm. Hence, all the

[2] http://emotion-research.net/acii/acii2009/minitutorial-emotional-brain-computer-interface

eegData	The 64-channel EEG data.
t	Time (in microseconds) corresponding to each sample.
imageTrigger	Values of 1 correspond to the onset of non-target images. Values of 2 correspond to the onset of target images. Values of 0 are used elsewhere.
buttonTrigger	Values of 1 correspond to the onset of a button press. Values of 0 are used elsewhere.
eegLabel	The label for each of the 64 electrodes. The i^{th} label corresponds to the i^{th} row of eegData and the i^{th} entry of eegCoord. The letters and numbers contained in each label correspond to the location of the given electrode. For example, F, P, O, T correspond to frontal, parietal, occipital, and temporal regions of the brain, respectively.
eegCoord	The coordinates of each of the 64 electrodes. The (spherical) coordinates are measured in degrees of inclination from Cz (positive values correspond to the right hemisphere, negative values correspond to the left hemisphere) and degrees of azimuth (from T7 for the left hemisphere and from T8 for the right hemisphere, positive values correspond to anti-clockwise rotations, negative values correspond to clockwise rotations).

following data analysis is based on the training set. We will employ the SVM with Gaussian kernel as the main learning algorithm in next subsection.

4.1.2 Signal preprocessing and feature extraction

Signal preprocessing To enhance the signal-to-noise ratio and to extract features from a continuous multi-channel (64 channels) electroencephalographic recording, signal preprocessing was implemented in the following way:

- **Filtering** To remove the noise from brain signals, a preprocessing Chebyshev band-pass filter of frequency band $[f_1, f_2]$ is applied to the signal. Note that f_2 is set to $f_2 = \frac{256}{3 \cdot \mathrm{KD}}$ in order to avoid aliasing.
- **Downsampling** with a factor of KD (extract regularly $\frac{1}{\mathrm{KD}}$ sample in the signal) is employed to reduce the dimension of the data.
- **Time segmentation** Finally, after the triggering of each stimulus image, a time segmentation of size 1000 ms is extracted from all channels and concatenated so as to form a single vector that is used as feature for the classifier used for parameter tuning.

The steps we follow here is actually a refinement of the preprocessing method for P300 based BCI system. For detailed implementation, we refer the readers to the paper [Rakotomamonjy 2008b]. To extract an EEG feature, we need to fix the relevant parameters (f_1, KD). In this part, we employ a linear SVM classifier (involved parameter C is set to be 1) to maximize the average validation set with AUC by 10-fold cross validation. The final validated parameter values are: $f_1 = 1$ Hz, $\mathrm{KD} = 5$ and $f_2 = 17$ Hz.

Feature extraction The experimental setup of EEG recording suggests that discriminative information appears at various delays after image trigger. Due to the experimental protocol, we have two kinds of discriminative information: P300 related to rare events and N500 related to motor response associated with pressing a key on the keyboard. To take into account this prior knowledge, we chose different time windows after each image trigger: $[0, 625]$ ms (P300), $[343, 968]$ ms (N500), $[0, 1000]$ ms (both). We select one of these window parameters ($\mathrm{TW} \in \{\mathrm{P300}, \mathrm{N500}, \mathrm{both}\}$) through validation.

Channel selection For reducing problem dimensionality and eventually increase classifier performances [Rakotomamonjy 2008b, Lotte 2007b], we considered channel selection by evaluating the discriminative ability of each EEG channel. For this purpose, we split 10 times the data set and learned

linear SVM classifiers for the three time window sizes. For each split, we stored the hyper-plane $\mathbf{w}$ with the highest AUC validation score, and averaged them as $\mathbf{w}_m$ over the splits. Discriminative ability of each channel is estimated by summing the squared coefficients (the power) of $\mathbf{w}_m$ over time and then by sorting the channel in decreasing power. This analysis ranks on top channels with most discriminative patterns. Since the decreasing power of the channels shows no obvious threshold to select the most discriminative channels (Figure 4.1), the number of selected channels nbC $\in [10, 15, 20, 30, 40, 50]$ is chosen by validation.

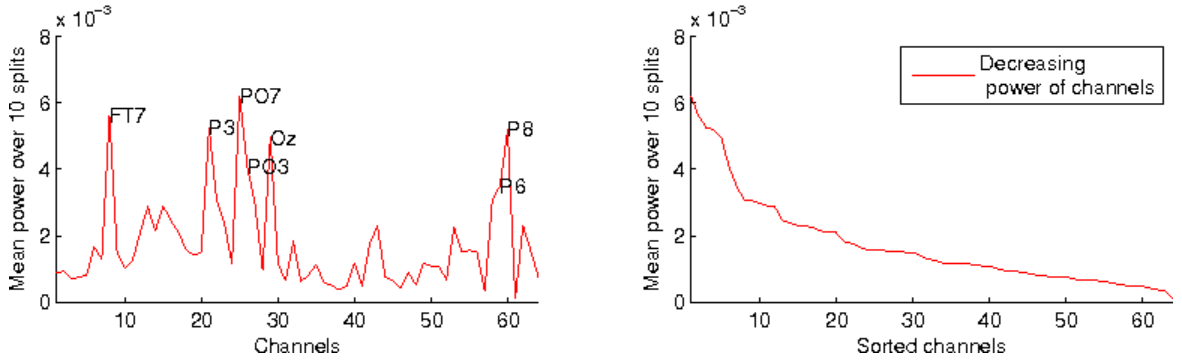


Figure 4.1: Discriminative power of channels for P300: the most important channels are indicated on the left figure, and the naming follows the 10-20 System of Electrode Placement presented in figure 1.2.

Final features The parameters TW and nbC are validated inside a double cross validation loop (figure 4.2). For each split, regularization parameter C and Gaussian kernel bandwidth σ are optimized in a second 10-fold cross validation. The mean test AUCs over the splits for every (TW,nbC) configuration are then compared. The best performance was obtained on the P300 window with 30 channels which we used for the final classifier.

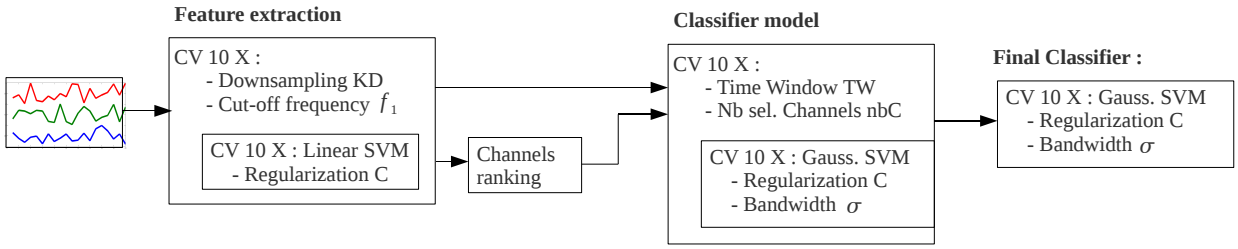


Figure 4.2: Sequential cross validations of the signal processing chain and classifier: we first employ a 10-fold cross validation to carefully select the hyperparameters for signal preprocessing; secondly, we rank the discriminative power of channels for P300 to fix the final selected channels; thirdly, another 10-fold cross validation is implemented to get the final feature vectors. For the classifier model, we decide the relevant hyperparameters, namely, the regularization parameter C and kernel bandwidth σ by a 10-fold cross validation.

4.1.3 Experimental analysis

Test data As stated in [Kenneth 2010], for the test data, there are 64 channels of EEG data and the sampling rate is 256 Hz. There is no delay between images within the same block. The subject rests between blocks for as long as they wish, and they pressed the space bar to signify that they detected a

target, to initiate a new block, and to clear feedback information that was displayed after each block. Unlike before, the test data consist of 890 blocks and 9891 satellite images and the total number of samples of EEG data is 1603334. Every other image within a block is a mask image (mask images do not contain targets) and the buttonTrigger variable is not available. The imageTrigger variable takes only values of 0 or 1, where a 1 corresponds to the onset of each prospective target image (i.e., the satellite images) and 0 is used elsewhere. Another difference is that 4 different image durations are used in the test data. The image durations, which apply to both satellite and mask images, are (approximately) 50 ms, 100 ms, 150 ms and 200 ms. All images within a given block have the same image duration and all blocks having a specified image duration are grouped together. Each block contains 22, 10, 7 and 5 prospective target images when the image duration is 50 ms, 100 ms, 150 ms and 200 ms, respectively. Keep in mind that the time difference between successive prospective target images is twice the corresponding image duration due to the presence of the mask images. Hence, successive prospective target images within a block appear every (approximately) 100 ms, 200 ms, 300 ms or 400 ms.

Evaluation strategy To ensure the fairness of competitor classifiers, we only report the results of competition results and without do any further post-processing for this data set.

Final classifier Once all preprocessing parameters have been tuned, we cross validate the regularization parameter C and the kernel parameter σ w.r.t. the whole training data set. The whole process gives us an efficient and fast decision method since the number of support vector is quite low (less than 200 compared with the whole 2775 samples). The value σ selected from the validation process seems relatively large ($\sigma = 133.3$). It must be compared to the unit variance of the normalized features, the dimensionality of the problem

$$dim = \frac{\mathrm{TW} \times F \times \mathrm{nbC}}{\mathrm{KD}}$$

where F is the sampling rate, ($dim = 0.625 \times 256/5 \times 30 = 960$, $\sqrt{dim} = 30,98$), and the median of pairwise distances between samples ($d_{med} = 35.5$). The classifier induced by a kernel with large bandwidth could be over regularized. Hence, one can expect for such a classifier to behave somehow like a linear one.

Result analysis The classifiers used in the three best entries are: a generative classifier which estimates a joint probability function with an AUC of 0.8229, a classifier based on T-weights with an AUC of 0.8217. The proposed feature (channel) selection strategy achieves the third places with an AUC of 0.8188 among a total of 35 algorithms involved in the competition. Prior knowledge on EEG signal classification problem promotes linear classifiers [Müller 2003]. Nevertheless, we tested linear and non-linear classifiers LDA, linear SVM, gaussian SVM, neural networks with their own optimized channel selection and parameters. However, the results of the competition showed that Gaussian SVM and the neural networks outperformed linear classifiers with the framework of feature selection and model selection strategy presented in this section. This result confirms the observation in [Lotte 2007a] about synchronous BCI and Gaussian SVM with high dimensionality. Regarding the classifier performance presented in Table 4.1, the SVMs achieve a mean AUC of 0.64, which, being far from that of Gaussian SVM presented here (0.8188). This suggests the importance of careful feature selection and model selection in BCI system.

Comparison with other competitors As shown in Table 4.1, various classifiers were tested in this competition. Among the whole entries, five of the methods were implemented by our machine learning team and they are very similar except for the classifier. For these 5 entries, the SVM classifier with Gaussian kernel performed slightly better than Linear Discriminative Analysis (LDA), linear SVM and a convolutional neural network. All 4 of these 5 performed much better than a one-hidden-layer neural network.

Table 4.1: Mean classifier performance for each type of classifier involved in MLSP competition (from [Kenneth 2010])

Classifier	No. of entries	Mean AUC
SVM	11	0.64
LDA	9	0.70
Neural network	5	0.66
Linear logistic	3	0.63
Other	7	0.67
Bagging	9	0.62
Non-bagging	26	0.68
All classifiers	35	0.66

4.1.4 Summary

In this section, we investigated a careful feature (and/or channel selection) with simple classifier as a counterbalance to complex classification algorithms in BCI data analysis. Precious model selection (or hyper-parameter selection) is implemented by 10-fold cross validation accurately for each component of signal processing, namely, selecting the upper and lower bound of band-pass filter, selecting the size of relevant time window, selecting the type of discriminative information, model selection of the classification algorithms and channel selection. The competition results showed that simple classifier can also achieve satisfying results with good enough features in off-line data analysis. However, the feature (and/or) channel selection implementations in this section involves heavy computation burden which hinder the applications in online context. Compared with more complex classification algorithms involved in Chapter 2 and 3, the work of current session evaluate the counterbalance of simple classifier in precondition of off-line data analysis.

Unrelatedly, in Section 4.2, we explore the effect of subject's emotional states on BCI performance when employing simple classification algorithm.

4.2 An emotional SSVEP based BCI system

Among various types of BCIs, the Steady State Visual Evoked Potential (SSVEP) based approach is advantageous with its design flexibility and little user training. In this section, we design a SSVEP based BCI system by accounting for the user's emotional states. Current section is organized as follows. In Section 4.2.1, we present preliminaries for both SSVEP based BCI system and relevant background of emotion. In Section 4.2.2, the experimental setup is detailed. In Section 4.2.3, the implementation of EEG acquisition is described, and data structure of the EEGs with different emotional states are presented in 4.2.4. Signal processing is described in Section 4.2.5. And finally, conclusions are made in Section 4.2.9.

4.2.1 Preliminaries

SSVEP based BCI system is an important Visual Evoked Potential (VEP) based BCI paradigms (as shown in Figure 4.3). The SSVEP are natural responses for visual stimulation at specific frequencies. When the retina is excited by a visual stimulus ranging from 3.5 Hz to 75 Hz, the brain generates an electrical activity at the same (or multiple of the) frequency of the visual stimulus. SSVEP based BCIs are based on detecting the target stimulus that the subject is looking at in case of the stimuli with different flashing frequencies.

Goal Current research is supported by the French-Chinese project "Programme Xu Guangqi". We precise the goal of the project from two perspectives:

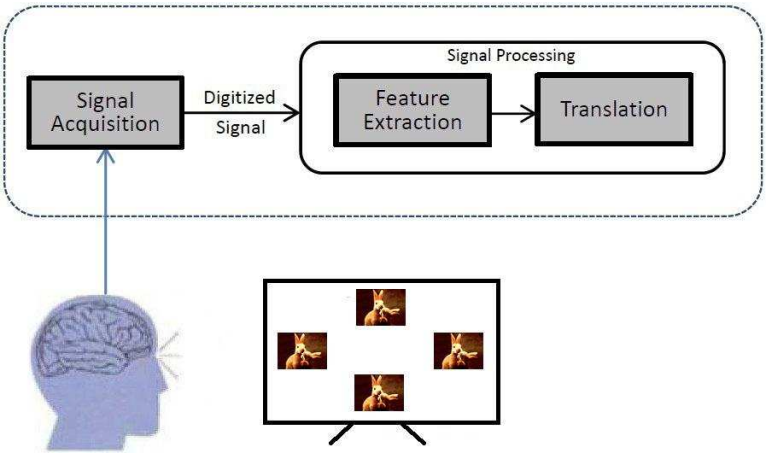


Figure 4.3: SSVEP based BCI paradigms

- **Long term goal** is to detect the emotion of subject and use that information to adapt the classifiers (online or not) to adapt the feedback with subject.

- **Short term study** presented in current section investigates whether or not the BCI classification is influenced by emotion. For that sake, we elaborate an overall experiment design with the help of University of Science and Technology of China (USTC) to acquire reliable data and testing significance of emotion on classification.

Emotion induction The work presented here involves human emotions when performing BCI tasks on short-term period. Emotions elicited by stimuli can be rated within the valence-arousal space by using the Self Assessment Manikin (SAM) (see Figure 4.4). SAM is non-verbal graphical tool on which subjects have to rate on a nine-point scale how they feel. In our experiments, three specific areas of the valence-arousal emotional spaces are defined, corresponding to negatively excited, positively excited and calm-neutral states [Chanel 2009]. To simplify the presentation, we denote them as negative, positive and neutral respectively in the remaining section. Various emotion induction methods exist in current

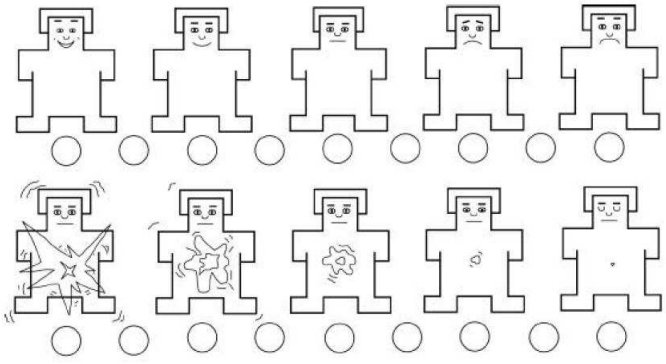


Figure 4.4: The self-Assessment Manikin (SAM). Top: valence; bottom: arousal. Valence is depicted as a smiling happy, figure transiting into a frowning, unhappy figure. For arousal SAM ranges from a sleepy figure, which eyes closed, to an excited figure, with eyes open (from [Nijboer 2009]).

research. We employ emotional video clips and the three types of images from the International Affective Picture System (IAPS) to induce predefined emotional states in this section.

Related work Existing research focuses on assessing emotion from EEG/peripheral signals [Aftanas 2004,Rozenkrants 2008] or detecting influences of different emotional states on EEGs [Aftanas 2004]. Some research on emotion related BCI systems were reported recently. [Yazdani 2009] propose a BCI based P300 evoked potential to implicit emotional tagging of multimedia content. According to their conclusion, EEGs can be assessed directly during consumption of multimedia for recognition of the induced emotion. [Lukito 2009] investigated the effect of cognitive and emotional states on the performance of a P300 based BCI system on locked-in patients. However, they didn't make explicit conclusion. [Bakardjian 2010] achieves the closest research with our experiments. In their experiments, they evaluate the affective SSVEP response to emotional face videos, and then apply such affective SSVEP responses to control a robotic arm. It has been proved that operation of a robotic arm were substantially improved by their proposed affective BCI system.

Therefore, among the existing emotion related BCI systems, we chose SSVEP based BCI which has the easiest experimental configuration and has positive reports. Experimental implementation are detailed in next section.

4.2.2 Experimental setup

Experimental environment Experiments were implemented in University of Science and Technology of China, Hefei, China. As shown in figure 4.5, we used two rooms to implement the emotional BCI experiments. One was for the experiment conductors, and the other was for the subjects. Both of them are ensured to be in a quiet and dark state. In the room of subject, the subject was seated 70 cm from a 22 inch LCD screen. In the room of experiment conductors, there are two separate computers for data collection and order sending. To ensure synchronization of the two computers, the EEGs were recorded with marks.

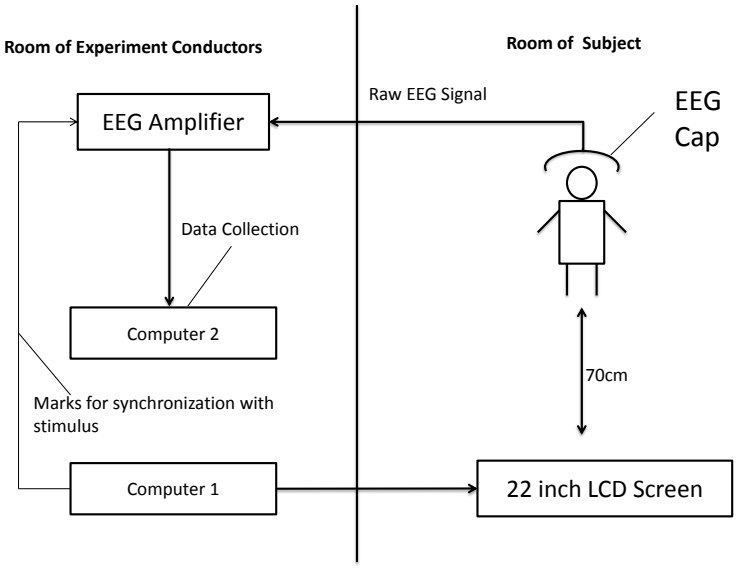


Figure 4.5: Experimental diagram of the emotional BCI system.

Hardware and software EEG signals were acquired via a Quik-cap (Neuro Inc., EI Paso, TX) with 32 Ag-AgCl electrodes arranged in an extended 10-20 system montage. Neuroscan Synamps2 bio amplifiers were used, and EEG signals were recorded using Neuroscan Scan software (v4.3.1). For presentation, E-prime (v2.0 beta) was used. Refresh rate of LCD screen is 60 Hz, and the sampling rate is 500 Hz. During the experiments, we ensure the electrode impedance is lower than 20 KΩ.

4.2.3 EEGs acquisition

Initially, we employed emotional video clips to elicit predefined emotion. As emotion decaying with time, we only select the EEGs that were reported in a satisfying degree by the subjects. We then merged emotion elicitation with BCI task by employing IAPS images. The aim of the latter is to ensure the subjects maintain the target emotion when performing the BCI task. Regarding different emotion elicitation strategies, we executed two kinds of experiments: video-drive-emotional-SSVEP based BCI and IAPS-image-drive-emotional-SSVEP based BCI systems. For the first ones, emotion elicitation and BCI tasks were implemented separately. For the second ones, we merged the two components together, namely, each IAPS image (with preselected emotional state) is flicking as a BCI task.

Before experiments, we evaluated the video clips or IAPS images by SAM method (as shown in figure 4.4). For the first kind of experiments, 36 video clips were involved in the experiments, of which 14 are positive and 22 are negative. For the neutral emotion, we let the subjects to calm themselves without seeing any video. The reason lies in that some of subjects reported that they feel delighted after seeing the neutral videos. For the second, 73 positive IAPS images were selected with valence ratings from 7.02 to 8.34 and arousal ratings from 2.67 to 5.94. 83 negative pictures were chosen with valence ratings from 1.8 to 3.47 and arousal ratings from 3.52 to 5.5. 161 neutral pictures were used with valence ratings from 4.46 to 5.46 and arousal ratings ranging from 1.55 to 4.27.

Video-driven-emotional-SSVEP based BCI As shown in Figure 4.6, we implemented three sessions of experiments for each subject. Each one corresponds one emotional state. One session is composed of several trials of BCI task. Every four trials were executed hereafter:

- Step 1: "the experiment starts" is displayed for 1s.
- Step 2: A fixation cross is displayed for 2s to concentrate the subject's sight.
- Step 3: Preselected video clip (with positive or negative emotional state) is displayed (ranging from a duration from 1 to 3 minutes) to induce a short-term emotion.
- Step 4: Highlighting the target direction in yellow out of four (left, up, right and down) to remind the subject which rectangle to be gazed at in the following BCI task.
- Step 5: Target rectangle flicking for 10s, and the four rectangles were flicked clock-wisely with frequencies of 10, 11, 12 and 15 Hz.
- Step 6: "the experiment ends" is displayed for 1s.

As a contrast, the subjects also did the BCI tasks directly without watching any video clip. This type of experiment is considered as for the neutral state, which contains 8 trials for each of the four directions. The states of subjects' current arousal, valence and emotion category were also recorded. These EEGs serve as calibration samples.

IAPS-image-driven-emotional SSVEP based BCI Data acquisition topology of IAPS-image-drive-emotional-SSVEP based BCI system is presented in Figure 4.7. Different from the video-driven ones, one session is composed of four blocks with selected frequencies and each block is composed of many trials. Every trial is implemented hereafter:

- Step 1: A fixation cross "+" is displayed for 3s to concentrate the subject's sight.
- Step 2: Highlighting the target direction in yellow out of four (left, up, right and down) to remind the subject which rectangle to be gazed at in the following BCI task.
- Step 3: Four same emotional IAPS picture flickers with frequencies of 10, 11, 12 and 15 Hz independently for 10s.
- Step 4: Rest for 10s and then returns to Step 2.

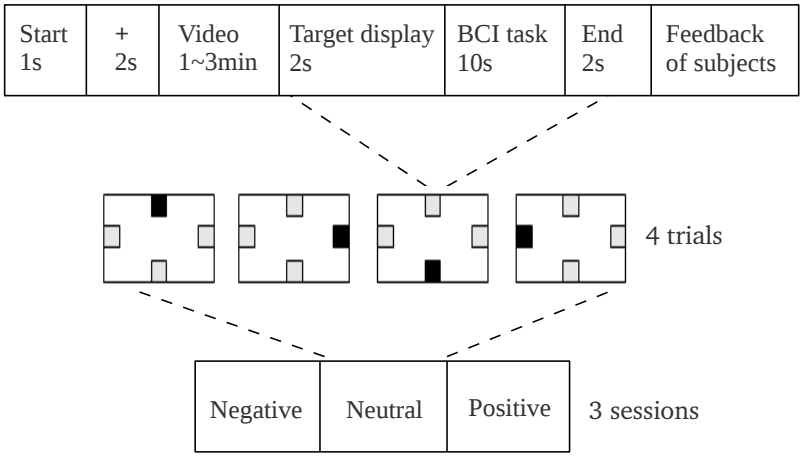


Figure 4.6: Data acquisition topology of video-driven-emotional-SSVEP based BCI system.

- Step 5: Repeat from Step 2 to 4 for 21 times, namely, there were seven positive pictures, seven neutral pictures and seven negative pictures flicked.

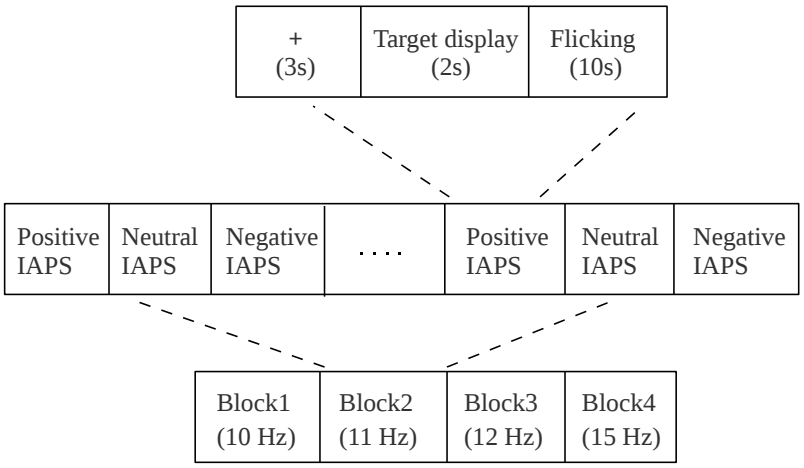


Figure 4.7: Data acquisition topology of IAPS-image-driven-emotional-SSVEP based BCI system.

At the end of each experiment (video or IAPS picture driven emotional BCI), the subjects are required to fill in a feedback form by telling the result to the experiment conductor.

Subjects Eight healthy participants (seven male and one female), aged between 21 to 25 years old participated in the video-driven experiments. Six healthy participants (three males and three females), aged from 18 to 23 years old participated in the IAPS-image-driven experiments. All subjects were ensured in normal mental status and normal corrected vision.

4.2.4 Experimental data

Raw EEG data structure EEGs are stored in ".cnt" file. Each file contains the recorded EEG data for one experiment. The EEGs were recorded with the attributes hereafter: data, mark, sampling rate,

start latency, end latency, source, subject and emotion type. Totally, there are 67 files for negative emotion, 60 files for neutral emotion and 64 files for positive emotion.

data	EEGs from 34 channels (2 of them are references channels).
mark	a real number indicating the target frequency, namely, mark $\in \{10, 11, 12, 15\}$.
sampling rate	500 Hz.
start latency	a real number indicating the start of BCI task.
end latency	a real number indicating the end of BCI task.
source	the file name.
subject	subject index.
emotion type	a string indicating the emotional state of a subject when performing BCI task. It belongs to {"negative", "positive", "neutral"}.

Extract valid data files According to the experimental design, we only extract the EEGs corresponding to the EEG tasks. The data file "without mark" or "shifting of the time window corresponding BCI task is beyond 300ms" (the length of whole BCI task is 10s) will be regarded as invalid.

Data segmentation After extracting valid data files, we denote the subjects with valid data as "s1" to "s14" to simplify the presentation. Valid EEG data files were splitted into trials. Each trial corresponds to one target rectangle (with specific frequency). For each trial, dimension of the EEGs is about $5000 = 500 \times 10$. Extracted trials were summarized in Table 4.2. Finally, 682 trials were obtained from valid data files of 14 subjects. 233 trials correspond negative emotional state, 221 trials correspond neutral state and 228 trials correspond positive state.

Table 4.2: Trial information of each subject.

Subject	s1	s2	s3	s4	s5	s6	s7	s8	s9	s10	s11	s12	s13	s14	Sum
Negative	8	12	0	8	8	12	12	12	8	7	28	28	28	28	233
Neutral	4	8	8	8	8	8	8	8	24	25	28	28	28	28	221
Positive	16	20	0	8	8	4	4	4	8	7	28	28	28	28	228
Total	28	40	8	24	24	24	24	24	74	76	84	84	84	84	682

4.2.5 Signal processing

Signal preprocessing For each trial that corresponding to 10s of EEGs $S(\tau_i)$ $(i = 1, \cdots, 5000)$, we employ a 2-order Butterworth band-pass filter of frequency band $[f_1, f_2]$. Regarding the frequencies $\{10, 11, 12, 15\}$ involved in this experiments, we use four independent filtering with the frequency band [9.75, 10.25], [10.75, 11.25], [11.75, 11.25] and [14.75, 15.25] separately. For simplicity, we denote the data after each filtering as $\{F_j(\tau_i)\}_{j=1}^4$ for each trial.

Feature exaction We employ the Power Spectral Density (PSD) feature[3] in this experiment. For each filtered signal $F_j(\tau_i)$ of one channel, the PSD value is calculated as follows.

$$\mathrm{PSD}(F_j) = \frac{1}{5000} \sum_{i=1}^{5000} F_j(\tau_i)^2$$

Finally, the PSD feature vector of a trial corresponding to one channel can be formulated as

$$\mathrm{PSD}^{\mathrm{ch}} = [\mathrm{PSD}(F_1), \mathrm{PSD}(F_2), \mathrm{PSD}(F_3), \mathrm{PSD}(F_4)].$$

[3] We are in fact using the energy of signals after filtering which corresponds to considering energy at target frequencies 10, 11, 12, 15 Hz. Abusively, we will call this feature PSD.

Let N_{ch} be the number of channels used in data analysis, the final PSD features of a trial can be attained

$$\mathbf{p} = [\mathrm{PSD}_1^{\mathrm{ch}}, \cdots, \mathrm{PSD}_{N_{ch}}^{\mathrm{ch}}]$$

where $\mathbf{p}$ has a dimension of $4 \times N_{ch}$. In the experiments, the two electrodes "F3" and "T8" were broken. Beyond another two electrodes corresponding electrooculograph (EOG), namely "Heog" and "Veog", we finally use the EEGs from 28 ($N_{ch} = 28$) channels. To sum up, we get a feature vector of a trial with the following attributes:

PSD feature	$p = [\mathrm{PSD}_1^{\mathrm{ch}}, \cdots, \mathrm{PSD}_{N_{ch}}^{\mathrm{ch}}]$.
Label	$y \in \{1, 2, 3, 4\}$ corresponding to the frequencies 10, 11, 12 and 15 Hz respectively.
Emotion	$e \in \{-1, 0, +1\}$ corresponding to the negative, neutral and positive emotional state separately.
SubjectCode	$s \in \{1, \cdots, 14\}$ denotes the index of subjects.

Evaluation strategy As shown in Algorithm 9, we employ a "leave one subject out" strategy to evaluate the BCI performance with different emotions. For a more detailed presentation, we implemented the following three groups of experiments: (1) positive emotion VS negative emotion; (2) negative emotion VS neutral emotion; (3) neutral emotion VS positive emotion in order to analyze the influence of emotion. We employ **statistical test** in this evaluation procedure. The algorithms obtained with two different emotions were denoted as "A" and "B". Hence, the null hypothesis to be tested is: "for the EEG data from the same subjects, the two learning algorithms will have the same error rate". Among all possible tests available to check for this hypothesis (see for instance [Dietterich 1998]), none of them perfectly fits our experimental framework and associated hypothesis. We decided to use two statistical tests, namely, McNemar's test and the Wilcoxon signed rank test which, although not fulfilling all requirements, allow to draw reasonable conclusions. Note that McNemar's procedure tests the difference between predictions of the two classifiers while Wilcoxon test checks for a difference between classifiers' accuracy.[4]

Algorithm 9 Leave one subject out (Emotion A VS B)

From subject s1 to s14:

repeat

Train two independent classifiers f_A and f_B under different emotions without current subject.

Calculate the model output $\hat{f_A}$ and $\hat{f_B}$ on data of the subject left out.

- McNemar's test.
- Wilcoxon signed rank test.

until the last subject

4.2.6 Result analysis through McNemar's test

We first introduce the McNemar's test [Everitt 1977] which is based on a $\mathcal{X}^2$ test for goodness-of-fit that compares the distribution of counts expected under the null hypothesis to be observed. Let $\hat{f}_A$ be the classifier output obtained by algorithm A on the testing set, and let $\hat{f}_B$ be the classifier output provided by algorithm B on the same test set. Then the null hypothesis can be formulated as

$$\begin{cases} \mathcal{H}_0 : & f_A(z) = f_B(z) \\ \mathcal{H}_1 : & f_A(z) \neq f_B(z). \end{cases} \tag{4.1}$$

[4]The two classifiers may have the same accuracy (the same error rate) but may predict different labels for the same input.

For each example z in the test set (in this section, it means the samples from the same subject), we record how it was classified and construct the following contingency table, where $n = n_{00}+n_{01}+n_{10}+n_{11}$ is the

n_{00}: number of examples misclassified by both $\hat{f}_A$ and $\hat{f}_B$	n_{01}: number of examples misclassified by $\hat{f}_A$ but not by $\hat{f}_B$
n_{10}: number of examples misclassified by $\hat{f}_B$ but not by $\hat{f}_A$	n_{11}: number of examples misclassified by neither $\hat{f}_A$ nor $\hat{f}_B$.

total number of examples in the test set. Under the null hypothesis, the two algorithms with different emotions should have the same error rate, namely, $n_{01} = n_{10}$. Then, the expected counts when the null hypothesis holds is

$$n_{01} = n_{10} = \frac{n_{01} + n_{10}}{2}$$

Hence, the $\mathcal{X}^2$ distance is

$$\frac{(n_{01} - \frac{n_{01}+n_{10}}{2})^2}{\frac{n_{01}+n_{10}}{2}} + \frac{(n_{10} - \frac{n_{01}+n_{10}}{2})^2}{\frac{n_{01}+n_{10}}{2}} = \frac{|n_{01} - n_{10}|^2}{n_{10} + n_{01}}$$

In practice, a "continuity correction" term (of -1 in the numerator) is introduced to take account for the fact that the statistic is discrete while the $\mathcal{X}^2$ distribution is continuous

$$C_{AB} = \frac{(|n_{01} - n_{10}| - 1)^2}{n_{01} + n_{10}}. \tag{4.2}$$

For a one-degree $\mathcal{X}^2$ distribution, namely $\mathcal{X}^2_{1,0.95} = 3.84$, the null hypothesis $\mathcal{H}_0$ holds with

$$\begin{aligned} \mathbb{P}(C_{AB} \geq 3.84) &= 5\% \\ \Rightarrow \mathbb{P}(C_{AB} \leq 3.84) &= 95\%. \end{aligned} \tag{4.3}$$

With a sufficiently large number of discrepancies between $\hat{f}_A$ and $\hat{f}_B$, the statistic C_{AB} can be approximate by a $\mathcal{X}^2$ distribution with one degree of freedom. In other words, when the calculated C_{AB} is less than 3.84, the null hypothesis $\mathcal{H}_0$ is accepted and the two algorithms obtained with different emotions are assumed to perform similarly on BCI classification. In the contrary, when the value of C_{AB} is larger than 3.84, $\mathcal{H}_0$ is rejected and the algorithms involved in the test are assumed to perform differently. For detail, we refer the reader to [Dietterich 1998].

Results analysis We executed three tests by employing linear SVM (with $C = 1$ un-optimized).

- **positive VS negative** For each subject, we learn a model using the data from all the other 13 subjects with positive (negative) emotion. The test set is the sample of current subject (with all emotional states available). Finally, we get the following contingency table regarding the algorithm with positive and negative emotion.

SubjectCode	1		2		3		4		5		6		7	
Confusion matrix	11	5	7	4	3	0	5	2	3	2	6	0	9	2
	3	9	2	27	1	4	0	17	1	18	1	17	8	5
SubjectCode	8		9		10		11		12		13		14	
Confusion matrix	10	5	46	5	47	6	5	2	44	4	1	5	37	11
	1	8	2	21	13	10	4	73	10	26	1	77	9	27

  Finally, we got the sum of these confusion matrix, namely, $n_{00} = 247$, $n_{01} = 40$, $n_{10} = 80$, $n_{11} = 315$ and thus $C_{AB} = 12.67 \gg 3.84$. And thus the possiblity of that algorithm A and B perform differently is quite big. According to the results, we can conclude that the positive and negative emotions have different effectiveness on BCI performance.

- **negative VS neutral** We redo the same experiment for negative and neutral emotions with results in the tables below. In this case, we got the sum of confusion matrix as $n_{00} = 250$, $n_{01} = 40$, $n_{10} = 77$, $n_{11} = 315$ and thus $C_{AB} = 11.08 \gg 3.84$. According to the results, we conclude confidently that the negative and neutral emotion have very different effectiveness on BCI performance.

SubjectCode	1		2		3		4		5		6		7	
Confusion matrix	12	2	8	1	4	0	4	1	3	1	6	1	17	0
	2	12	13	18	2	2	4	15	8	12	1	16	1	6
SubjectCode	8		9		10		11		12		13		14	
Confusion matrix	9	2	47	1	58	2	6	3	41	13	1	1	34	12
	5	8	4	22	6	10	1	74	6	24	2	80	22	16

- **neutral VS positive** We proceed as previously and set the table as follows.

SubjectCode	1		2		3		4		5		6		7	
Confusion matrix	9	6	12	2	3	0	3	6	5	2	4	4	14	6
	6	7	4	22	1	4	1	14	4	13	2	14	1	3
SubjectCode	8		9		10		11		12		13		14	
Confusion matrix	11	2	38	8	48	10	7	11	40	9	1	14	36	13
	3	8	6	22	6	12	1	65	14	21	1	68	10	25

Finally, we calculate the sum of confusion matrix of each subject, and attained $n_{00} = 234$, $n_{01} = 53$, $n_{10} = 56$ and $n_{11} = 339$. In this case, $C_{AB} = 0.037 \ll 3.84$, then we conclude that the neutral and positive emotion won't have different effectiveness on BCI performance.

4.2.7 Results analysis through Wilcoxon signed rank test

Wilcoxon signed rank test is a non-parametric statistical hypothesis test [Hollander 1999]. The Wilcoxon signed rank test for the emotional BCI can be setup as follows: assuming we collect observations with two emotional states $\{z_k^A\}_{k=1}^{14}$ and $\{z_k^B\}_{k=1}^{14}$, k denotes the particular subject that is being referred to and the first observation measured with one emotion on subject k be denoted as z_k^A and the second observation with another emotional state be z_k^B. Let $\Delta_k = z_k^A - z_k^B$ $(k = 1, \cdots, 14)$, we assume that: (1) The differences Δ_k are assumed to be independent. (2) Each Δ_i comes from the same continuous population. (3) Δ_i are ordered to make an inference about the mean difference.

The null hypothesis to be tested is

$$\begin{cases} \mathcal{H}_0: & \text{Median}(f_A(z) - f_B(z)) = 0 \\ \mathcal{H}_1: & \text{Median}(f_A(z) - f_B(z)) \neq 0. \end{cases}$$

Algorithm 10 demonstrates the relevant test procedure. The critical values of the Wilcoxon signed ranks test are listed as the following table, where S is calculated by equation (4.4). The subscript denotes confidence level and the superscript denotes the number of samples involved in the test.

12				14			
$S_{0.005}^{12}$	$S_{0.01}^{12}$	$S_{0.025}^{12}$	$S_{0.05}^{12}$	$S_{0.005}^{14}$	$S_{0.01}^{14}$	$S_{0.025}^{14}$	$S_{0.05}^{14}$
7	10	14	17	13	16	22	26

Table 4.3: Critical values corresponding $n = 12$ and $n = 14$.

Algorithm 10 Test procedure of Wilcoxon signed-rank test

Rank the absolute values $\{|\Delta_k|\}_{k=1}^{14}$ in ascending sequence. Let the rank of each non-zero $|\Delta_k|$ be R_k.
Denote the positive Δ_k values with $\phi_k = I\ (\Delta_i > 0)$, where $I(\cdot)$ is an indicator function: $\phi_k = 1$ for $\Delta_k > 0$, otherwise, $\phi_k = 0$.
The Wilcoxon signed ranked statistic W_+ is defined as

$$W_+ = \sum_{k=1}^{14} \phi_k R_k.$$

Define W_- similarly by summing ranks of the negative differences Δ_i.
Calculate S as the smaller of these two rank sums:

$$S = \min(W_+, W_-) \tag{4.4}$$

Find the critical value for the given sample size and the corresponding confidence level.
Compare S to the critical value, and reject $\mathcal{H}_0$ if S is less than or equal to the critical value.

Result analysis We perform three tests by employing linear SVM (with $C = 1$).

- **Positive VS negative** Employing the leave one subject out strategy (as shown in Algorithm 9), two independent classifiers were trained with the positive and negative emotions. z^{pos} denotes the number of misclassified samples by the algorithm with positive emotion, and z^{neg} is the number of misclassified samples by the algorithm with negative emotion. Table 4.4 shows relevant test results on the 14 subjects. We then get $W_+ = 13$, $W_- = 65$ and thus $S = \min(W_+, W_-) = 13 < S_{0.025}^{12} = 14$. We reject the hypothesis $\mathcal{H}_0$, that is, positive is better than negative for BCI operation.

Table 4.4: Wilcoxon signed-rank test among positive and negative emotion related algorithms.

SubjectCode	z^{pos}	z^{neg}	Δ	Sign	Rank	Sign*Rank
1	0.571	0.5	0.071	1	6	6
2	0.275	0.525	-0.25	-1	9.5	-9.5
3	0.375	0.75	-0.375	-1	12	-12
4	0.291	0.333	-0.042	-1	4	-4
5	0.208	0.458	-0.25	-1	9.5	-9.5
6	0.25	0.292	-0.042	-1	4	-4
7	0.458	0.75	-0.292	-1	11	-11
8	0.625	0.583	0.042	1	4	4
9	0.689	0.689	0	0	0	0
10	0.697	0.842	-0.145	-1	8	-8
11	0.083	0.083	0	0	0	0
12	0.571	0.560	0.011	1	1	1
13	0.071	0.036	0.035	1	2	2
14	0.571	0.667	-0.096	-1	7	-7

- **Negative VS neutral** As shown in Table 4.5, outputs of the algorithms based on negative and neutral emotional state are denoted as z^{neu} and z^{neg}. We get the observations $W_+ = 8$ and $W_- = 70$. Hence, $S = \min(W_+, W_-) = 8 < S_{0.025}^{12} = 14$. We reject the hypothesis $\mathcal{H}_0$, that is, neutral is better than negative emotion for BCI.
- **Neutral VS positive** Table 4.6 demonstrates the comparison observations between the algorithms with neutral and positive emotion. Outputs of the two algorithms are referred to z^{neu} and z^{pos}.

Table 4.5: Wilcoxon signed-rank test among neutral and negative emotion related algorithms.

SubjectCode	z^{neu}	z^{neg}	Δ	Sign	Rank	Sign*Rank
1	0.5	0.5	0	0	0	0
2	0.225	0.525	-0.3	-1	12	-12
3	0.5	0.75	-0.25	-1	10	-10
4	0.208	0.333	-0.125	-1	8.5	-8.5
5	0.167	0.458	-0.292	-1	11	-11
6	0.292	0.292	0	0	0	0
7	0.708	0.75	-0.042	-1	4	-4
8	0.458	0.583	-0.125	-1	8.5	-8.5
9	0.649	0.689	-0.04	-1	3	-3
10	0.789	0.842	-0.053	-1	5	-5
11	0.107	0.083	0.024	1	2	2
12	0.643	0.560	0.083	1	6	6
13	0.024	0.036	-0.012	-1	1	-1
14	0.548	0.667	-0.119	-1	7	-7

Finally, we get $W_+ = 53.5$ and $W_- = 51.5$. Hence, $S = \min(W_+, W_-) = 51.5 \gg S^{14}_{0.025} = 22$. We thus accept the hypothesis $\mathcal{H}_0$, that is, neutral and positive perform similarly for BCI operation.

Table 4.6: Wilcoxon signed-rank test among neutral and positive emotion related algorithms.

SubjectCode	z^{neu}	z^{pos}	Δ	Sign	Rank	Sign*Rank
1	0.571	0.5	0.071	1	8.5	8.5
2	0.275	0.225	0.05	1	7	7
3	0.375	0.5	-0.125	-1	12	-12
4	0.292	0.208	0.083	1	10	10
5	0.208	0.167	0.041	1	4.5	4.5
6	0.25	0.291	-0.041	-1	4.5	-4.5
7	0.458	0.708	-0.25	-1	14	-14
8	0.625	0.458	0.167	1	13	13
9	0.689	0.648	0.041	1	3	3
10	0.697	0.789	-0.092	-1	11	-11
11	0.083	0.107	-0.024	-1	1.5	-1.5
12	0.571	0.642	-0.071	-1	8.5	-8.5
13	0.071	0.024	0.047	1	6	6
14	0.571	0.547	0.024	1	1.5	1.5

4.2.8 Classification performances

During the statistical test procedure, we had to train classifiers. It seems interesting to report here their overall classification performances. Recall the PSD feature $\mathbf{p} = [\text{PSD}_1^{\text{ch}}, \cdots, \text{PSD}_{\text{N}_{\text{ch}}}^{\text{ch}}]$ and emotion e, a data set $\{(\mathbf{x}_i, y_i)\}_{i=1}^{682}$ is constructed. We consider two possibilities for the input $\mathbf{x}$: (1) Taking account for emotion e into the input, namely, $\mathbf{x}_i = [\mathbf{p}_i, e_i]$. The experiments with emotion use this type of inputs. (2) Employing $\mathbf{x}_i = \mathbf{p}_i$ for the experiments without emotion input.

We employed two kinds of classifiers, Maximum type and SVM, to evaluate the classification accuracy. For the SVM type, we employ a linear SVM classifier with $C = 1$. For the maximum one, output is

determined by the maximum sum of PSD features over each narrow filtering. That is, we extract the PSD values corresponding to each target frequency $\{10, 11, 12, 15\}$ and calculate the sum of them, the frequency corresponding the maximum value is taken as the target frequency.[5] From Table 4.7, we see that whether emotion e being one dimension of input does not affect the classification accuracy, while it tends to achieve better performance with neutral emotion.

Table 4.7: Comparison of classification accuracy in different context with leave one subject out strategy.

Emotion	Points	Dimension	Classification accuracy (%)	
			Maximum	SVM
All	682	113	44.8(7.9)	60.7(9.0)
No	682	112	44.8(7.9)	60.6(8.9)
Positive	228	113	37.6(11.7)	**59.0(8.2)**
Neutral	221	113	**46.2(10.2)**	58.5(9.2)
Negative	233	113	45.9(8.1)	49.5(9.0)

4.2.9 Summary

In this session, we implement a SSVEP based emotional BCI system. We implemented two statistical tests on the data, and confirmed that emotion does have an influence measurable through a BCI. This influence can be converted into differences between accuracy. More precisely, positive and neutral emotions are better than negative emotion for BCI systems. However, positive and neutral emotions are not statistically different.

4.3 Conclusions

In this chapter, we present independent implementations to explore two important issues in BCI system: (1) How to improve the BCI classification accuracy from perspective of signal processing strategy? (2) Is it possible to adapt the subject's emotional state to enhance the BCI operation?

The first exploration is executed by a BCI competition data analysis, "Mind reading, MLSP BCI Competition 2010". We confirmed that simple classifier with careful model/feature selection can also achieve competitive performance in off-line BCI data analysis compared with more complex models.

The second exploration is implemented by an emotional SSVEP-based BCI system. We performed careful experiments to attain reliable EEG data with three different emotional data, positive, neutral and negative. Based on these emotional EEGs, we executed two statistical tests to confirm whether emotion influence BCI performance or not. Both the two tests confirm that emotion does affect the BCI performance. More precisely, the positive and neutral emotion effect BCI performance similarly and the negative emotion performs very differently from the rest. From the view of classification accuracy, the negative emotion tends to damage the BCI operation.

Future work includes better tuning of classifiers and try other types of features. Further experiments include exploring adapting the classifier model to emotional state.

[5] The main reason of employing Maximum type classifier lies in that: EEG data of SSVEP-based BCI system are assumed to be sinusoidal signals at the same frequency as the flicking frequency. Hence, energy is assumed to be prominent in the frequency range around the target frequency.

Chapter 5

Conclusions and perspectives

Contents

5.1 Conclusions

In this thesis we explore to improve the signal processing in BCI system from two independent aspects: the first one dedicates to develop new machine learning algorithms to reduce the calibration procedure and the second one focuses on ameliorating the BCI data analysis by taking account for the user's emotional states. In what follows, conclusions drawn from this thesis are summarized separately.

Machine learning algorithms developed in this dissertation mainly involved the multiple kernel learning. Conclusions regarding this part of work are presented from the view of semi-supervised learning and the online MKL algorithms.

5.1.1 TSVM-MKL

TSVM-MKL was proposed in the context of semi-supervised learning as an efficient tool for reducing the calibration procedure for machine learning based BCI systems. Existing research with regard to SSL algorithms on BCI applications includes self-training algorithm, co-training algorithm, TSVM and graph-based methods. Most of them made strong model assumptions and thus need some a priori knowledge for particular BCI applications. The proposed TSVM-MKL realizes an automatical adaptation of model assumptions over different problems. By defining a pool of kernels for MKLs which is composed of basic kernels and manifold kernels, the two effective model assumptions are combined together in the framework of TSVM problem. Thus, the preference of model assumptions can be determined automatically in the learning procedure of the MKL algorithm. Experimental results on benchmark SSL data sets show the beneficial effect of the combination of both assumptions in terms of classification performances. And we also show that TSVM-MKL remains effective when a very few labeled samples are available. The latter phenomenon emphasizes the inductive property of TSVM-MKL which is essential to the real life applications such as BCI data analysis.

In the dissertation, we evaluate the TSVM-MKL algorithms on two kinds of BCI paradigms, namely, μ and β based BCI system that involves common spatial potential (CSP) features and motor imagery based BCI which employs power spectral density (PSD) features. Both of them demonstrate that the proposed TSVM-MKL improves the classification performance in BCI systems compared with the normal single kernel based SSL algorithms. We also propose a more elegant model selection approach for the SSL

algorithm regarding the non-stationarity of EEGs. Involved strategy can be summarized as determining the final model in an inductive way instead of a transductive manner on unlabeled data. Such model selection is proved to be more reliable.

Additionally, we benefit the advantage of multiple kernel learning, which can perform the feature selection automatically in the learning process, to implement the channel selection (or feature selection) in BCIs. As different mental tasks induce the responses in different brain regions, the channel selection for each mental task is implemented automatically in the learning process and results in a better performance compared with that without such channel selection.

5.1.2 LaMKL

Many real life machine learning problems can be more regarded as online rather than batch learning problems. In order to broaden the applications, it is important to realize an online version of the MKL algorithms. We propose an LaMKL algorithm for the ℓ_p-norm ($p > 1$) MKLs in this part of work. The motivations and contributions can thus be summarized from the following perspectives:

- ℓ_p-norm ($p > 1$) MKLs are non-sparse algorithms while sparsity can be attained by choosing p close to 1. For the problems with some prior knowledge that prefer sparsity and the fresh applications that tend to keep all solutions, the LaMKL algorithm which is based on ℓ_p-norm MKLs can achieve a self adaptation by adjusting the value of ℓ_p-norm. This is the most important motivation of adopting the ℓ_p-norm MKLs and can also be regarded as an important contribution for the community of online MKLs.

- The implementation of LaMKL adopts a similar approach developed in LASVM for online single kernel SVMs. It thus inherits almost all the advantages of LASVM, such as the small computation burden, fast convergence rate and competitive accuracies with the batch learning algorithms. Beyond these, the LaMKL can also achieve an optimal combination of relevant features with a priori knowledge benefiting the advantages of multiple kernel learning.

In the following subsection, we summarize the contributions of this dissertation beyond the development of machine learning algorithms.

5.1.3 Ameliorating BCI data analysis beyond the classifier itself

We execute two independent explorations in Chapter 4 and dedicate to ameliorate the BCI data analysis by only relying on simple classifiers. The first exploration is implemented from improving the methodology of model/feature selection in BCIs. And the second one is executed by enhancing the emotional states during the data acquisition stage. Contributions of them can be presented hereafter.

- Based on the "Mind reading, MLSP 2010 Competition" data analysis, we proved that careful feature (and/or channel selection) with simple classifier can be a counterbalance to complex classification algorithms in off-line BCI data analysis. Provided that such strategy involves heavy computation burden which hinder its online applications, we investigate another strategy that taking account for the user's emotional state in the same chapter.

- An affective SSVEP based BCI system is designed to investigate whether or not the BCI classification is influenced by emotion. The main reason lies on the close link between BCI paradigms and the human cognition procedure. Based on the reliable EEG data affected by different emotional states (positive, negative and neutral states), we confirmed that the emotional state does affect the BCI performance according to serious statistical tests.

The work involved in this dissertation retain some open questions as some future works presented in the subsequent section.

5.2 Perspectives

We mainly present the perspectives of this thesis from three aspects as follows: (1) extend the TSVM-MKL algorithm to be a more generalized semi-supervised MKL method. (2) improve the LaMKL approach and apply it in the BCI data analysis; and (3) realize an affective BCI system that can self adapt the classifier model to the user's emotional states.

5.2.1 Multiple kernel version of semi-supervised algorithms

In Chapter 2, we solely consider the sparse MKL restricted to be ℓ_1 constraint. Exploring non-sparse regularization with ℓ_p constraint ($p > 1$) in the semi-supervised learning context is also meaningful in real life machine learning problems. To this end, two important future works are proposed hereafter: (1) Employ the ℓ_p-norm ($p > 1$) on the combination of kernel weights in the framework of Transductive SVMs. Such strategy retains the advantages of multiple kernel learning which realizes the fusion of different model assumptions automatically, and achieves an adaptation of sparsity degree with specific applications. (2) Extend other types of semi-supervised learning algorithms such as Laplacian SVMs to the multiple kernel fashion. In this way, the tedious kernel parameter selection procedure can be avoided.

5.2.2 LaMKL

Experimental results have shown the effectiveness of LaMKL algorithms including the small computation burden and the fast convergence rate. Future works are dedicated to several directions, such as adopting shrinking strategy in the training process to achieve a sparse solution, speeding up the algorithm by more accurate working set selection and exploring simple heuristics to guide the reduced optimization problem. Benefit from the advantages of online learning algorithms, future works also include applying them in the BCI applications and some large scale problems.

5.2.3 Affective BCI system

The affective SSVEP based BCI system presented in Chapter 4 provides an accordance that the emotion does influence the BCI performance. This conclusion confirms the feasibility of affective BCI system, and the perspectives of them can be stated from two aspects: (1) for the short term study, we shall employ a more accurate model selection strategy to evaluate the influence of different emotional states on BCI performance; (2) for a long term study, we will investigate how to adapt the classifier to the user according to the detected emotional state.

Bibliography

[Aftanas 2004] L. Aftanas, N. Reva, A. Varlamov, S. Pavlov et V. Makhnev. *Analysis of evoked EEG synchronization and desynchronization in conditions of emotional activation in humans.* Neuroscience and behavioral physiology, vol. 34, no. 8, pages 859–867, 2004. (Cited on page 96.)

[Allison 2003] B. Allison. *P3 or not P3: toward a better P300 BCI.* PhD thesis, University of California, San Diego, 2003. (Cited on page 25.)

[Alpaydin 2004] E. Alpaydin. Introduction to machine learning. MIT Press, 2004. (Cited on page 28.)

[Bach 2004] F. Bach, G. Lanckriet et M. Jordan. *Multiple kernel learning, conic duality, and the SMO algorithm.* In Proceedings of the 21st International Conference on Machine learning (ICML 04), pages 41–48, Alberta, Canada, 2004. (Cited on pages 39, 69 and 73.)

[Bach 2008] F. Bach. *Exploring large feature spaces with hierarchical multiple kernel learning.* In Proceedings of Advances in Neural Information Processing Systems (NIPS 2008), Vancouver, Canada, 2008. (Cited on page 70.)

[Bakardjian 2010] H. Bakardjian, T. Tanaka et A. Cichocki. *Brain control of robotic arm using affective steady-state visual evoked potentials.* In Proceedings of the 5th IASTED Inter- national Conference Human-Computer Interaction, Hawaii, USA, 2010. (Cited on page 96.)

[Bashashati 2007] A. Bashashati, M. Fatourechi, R. Ward, G. Birch*et al.* *A survey of signal processing algorithms in Brain–computer interfaces based on electrical brain signals.* Journal of Neural engineering, vol. 4, pages 32–57, 2007. (Cited on pages 17, 22 and 24.)

[Bauer 1999] E. Bauer et R. Kohavi. *An empirical comparison of voting classification algorithms: Bagging, boosting, and variants.* Machine Learning, vol. 36, no. 1, pages 105–139, 1999. (Cited on page 19.)

[Belkin 2006] M. Belkin, P. Niyogi et V. Sindhwani. *Manifold regularization: a geometric framework for learning from label and unlabeled examples.* Journal of Machine Learning Research, vol. 7, pages 2399–2434, 2006. (Cited on pages 36, 41, 43, 53 and 54.)

[Bengio 2009] Y. Bengio. *Learning deep architectures for AI.* Foundations and Trends in Machine Learning, vol. 2, no. 1, pages 153–160, 2009. (Cited on page 35.)

[Bennett 1998] K. Bennett et A. Demiriz. Semi-supervised support vector machines. MIT Press, 1998. (Cited on page 36.)

[Berger 2007] T. Berger, J. Chapin, G. Gerhardt, D.J. McFarland, J. Principe et P. Tresco. *International assessment of research and development in Brain-computer interfaces.* Rapport technique, World Technology Evaluation Center, 2007. (Cited on pages 16 and 37.)

[Birbaumer 1999a] N. Birbaumer. *Slow cortical potentials: plasticity, operant control, and behavioral effects.* The Neuroscientist, no. 5, pages 74–78, 1999. (Cited on page 25.)

[Birbaumer 1999b] N. Birbaumer, N. Ghanayim, T. Hinterberger, I. Iversen, B. Kotchoubey, J. Perelmouter, E. Taub et H. Flor. *A spelling device for the paralyzed.* Nature, vol. 398, pages 297–298, 1999. (Cited on page 25.)

[Birbaumer 2000] N. Birbaumer, A. Kübler, N. Ghanayim, T. Hinterberger, J. Perelmouter, J. Kaiser, I. Iversen, B. Kotchoubey, N. Neumann et H. Flor. *The thought translation device (TTD) for completely paralyzed patients.* IEEE Transactions on Rehabilitation Engineering, vol. 8, no. 2, pages 190–192, 2000. (Cited on page 25.)

[Birbaumer 2007] N. Birbaumer et L. Cohen. *Brain–computer interfaces: communication and restoration of movement in paralysis.* Journal of Physiol, vol. 579, pages 621–636, 2007. (Cited on pages 16, 20 and 25.)

[Bishop 1996] CM Bishop. Neural networks for pattern recognition. Oxford University Press, 1996. (Cited on page 32.)

[Blanchard 2004] G. Blanchard et B. Blankertz. *BCI competition 2003–data set IIa: spatial patterns of self-controlled brain rhythm modulations.* IEEE Transactions on Biomedical Engineering, vol. 51, no. 6, pages 1062–1066, 2004. (Cited on pages 18 and 19.)

[Blankertz 2008] B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe et K.R. Müller. *Optimizing spatial filters for robust EEG single-trial analysis.* IEEE Signal Processing Magazine, vol. 25, no. 1, pages 41–56, 2008. (Cited on page 18.)

[Blankertz 2009] B. Blankertz, C. Sanelli, S. Halder, E. Hammer, A. Kübler, K.R. Müller, G. Curio et T. Dickhaus. *Predicting BCI performance to study BCI illiteracy.* BMC Neuroscience, vol. 10, page 84, 2009. (Cited on page 37.)

[Blumberg 2007] J. Blumberg, J. Rickert, S. Waldert, A. Schulze-Bonhage, A. Aertsen et C. Mehring. *Adaptive classification for brain computer interfaces.* In Proceedings of the 29th Annual International Conference of the IEEE in Engineering in Medicine and Biology Society, pages 2536–2539, 2007. (Cited on page 38.)

[Bordes 2005] A. Bordes, S. Ertekin, J. Weston et L. Bottou. *Fast kernel classifiers with online and active learning.* Journal of Machine Learning Research, vol. 6, pages 1579–1619, 2005. (Cited on pages 39, 70, 80 and 82.)

[Bos 2011] D. Bos, H. Gürkök, B. Van de Laar, F. Nijboer et A. Nijholt. *User experience evaluation in BCI: mind the gap!* International Journal of Bioelectromagnetism, vol. 13, no. 1, pages 48–49, 2011. (Cited on page 20.)

[Bottou 2004] L. Bottou. *Stochastic learning.* Advanced Lectures on Machine Learning, vol. 3176/2004, pages 146–168, 2004. (Cited on page 34.)

[Bottou 2007] L. Bottou, O. Chapelle, D. DeCoste et J. Weston. Large scale kernel machines. MIT Press, 2007. (Cited on page 32.)

[Brunner 2007] C. Brunner, M. Naeem, R. Leeb, B. Graimann et G. Pfurtscheller. *Spatial filtering and selection of optimized components in four class motor imagery EEG data using independent components analysis.* Pattern Recognition Letters, vol. 28, no. 8, pages 957–964, 2007. (Cited on page 17.)

[Buttfield 2006] A. Buttfield, P. Ferrez et J.R. Millan. *Towards a robust BCI : error potentials and online learning.* IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 14, no. 2, pages 164–168, 2006. (Cited on page 38.)

[Calvo 2010] R. Calvo et S. Mello. *Affect detection: an interdisciplinary review of models, methods, and their applications.* IEEE Transactions on Affective Computing, vol. 1, no. 1, pages 18–37, 2010. (Cited on page 90.)

[Cauwenberghs 2001] G. Cauwenberghs et T. Poggio. *Incremental and decremental support vector machine learning.* In Proceedings of Neural Information Processing System (NIPS 2001), Denver, CO, USA, 2001. (Cited on page 70.)

[Chanel 2009] G. Chanel, J. Kierkels, M. Soleymani, D. Grandjean et T. Pun. *Short-term emotion assessment in a recall paradigm.* International Journal of Human-Computer Studies, vol. 67, pages 607–627, 2009. (Cited on pages 90 and 95.)

[Chapelle 2002] Olivier Chapelle, Vladimir Vapnik, Olivier Bousquet et Sayan Mukherjee. *Choosing Multiple Parameters for Support Vector Machines*. Machine Learning, vol. 46, no. 1-3, pages 131–159, 2002. (Cited on page 69.)

[Chapelle 2005] O. Chapelle et A. Zien. *Semi-supervised classification by low density separation*. In Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, 2005. (Cited on page 41.)

[Chapelle 2006] O. Chapelle, B. Schölkopf et A. Zien, editeurs. Semi-supervised learning. MIT Press, Cambridge, MA, 2006. (Cited on pages 20, 28, 42 and 47.)

[Chapelle 2008a] O. Chapelle et A. Rakotomamonjy. *Second order optimization of kernel parameters*. In NIPS Workshop on Automatic Selection of Optimal Kernels, Vancouver, Canada, 2008. (Cited on pages 69 and 73.)

[Chapelle 2008b] O. Chapelle, V. Sindhwani et S. Keerthi. *Optimization techniques for semi-supervised support vector machines*. Journal of Machine Learning Research, vol. 9, pages 203–233, 2008. (Cited on pages 36, 42, 43, 44 and 56.)

[Chapin 1999] J.K. Chapin, K. Moxon, R. Markowitz et M. Nicolelis. *Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex*. Nature Neuroscience, vol. 2, pages 664–670, 1999. (Cited on page 18.)

[Cheng 2002] M. Cheng, X. Gao, S. Gao et D. Xu. *Design and implementation of a Brain-computer interface with high transfer rates*. IEEE Transactions on Biomedical Engineering, vol. 49, no. 10, pages 1181–1186, 2002. (Cited on page 24.)

[Cheng 2004] M. Cheng, W. Jia, X. Gao, S. Gao et F. Yang. *Mu rhythm-based cursor control: an offline analysis*. Clinical Neurophysiology, vol. 115, no. 4, pages 745–751, 2004. (Cited on page 18.)

[Chiappa 2004] S. Chiappa et S. Bengio. *HMM and IOHMM modeling of eeg rhythms for asynchronous bci systems*. In ESANN'2004 proceedings - European Symposium on Artificial Neural Networks, pages 199–204, Bruges, Belgium, 2004. (Cited on page 24.)

[Cincotti 2008] F. Cincotti, D. Mattia, F. Aloise, S. Bufalari, G. Schalk, G. Oriolo, A. Cherubini, M. Marciani et F. Babiloni. *Non-invasive brain-computer interface system: towards its application as assistive technology*. Brain Research Bulletin, vol. 75, pages 796–803, 2008. (Cited on page 20.)

[Collobert 2006] R. Collobert, F. Sinz, J. Weston et L. Bottou. *Large scale transductive SVMs*. Journal of Machine Learning Research, vol. 7, pages 1687–1712, 2006. (Cited on pages 45, 47, 48, 50, 51, 52 and 54.)

[Congedo 2006] M. Congedo, F. Lotte et A. Lecuyer. *Classsification of movement intention by spatially filtered electromagnetic inverse solutions*. Physics in Medicine and Biology, vol. 51, no. 8, pages 1971–1989, 2006. (Cited on page 24.)

[Cortes 2009a] Corinna Cortes, Mehryar Mohri et Afshin Rostamizadeh. *L2 regularization for learning kernels*. In Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI '09, pages 109–116, Arlington, Virginia, United States, 2009. AUAI Press. (Cited on page 73.)

[Cortes 2009b] Corinna Cortes, Mehryar Mohri et Afshin Rostamizadeh. *Learning Non-Linear Combinations of Kernels*. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams et A. Culotta, editeurs, Advances in Neural Information Processing Systems 22, pages 396–404. 2009. (Cited on pages 70 and 74.)

[Coyle 2005] D. Coyle, G. Prasad et T. McGinnity. *A Time-Frequency Approach to Feature Extraction for a Brain-computer interface with a comparative analysis of performance measures*. EURASIP Journal on Advances in Signal Processing, 2005. (Cited on page 18.)

[Coyle 2007] S. Coyle, T. Ward et C. Markham. *Brain–computer interface using a simplified functional near-infrared spectroscopy system*. Journal of Neural Engineering, vol. 4, no. 3, 2007. (Cited on page 16.)

[Coyle 2010] D. Coyle, A. Satti et T. McGinnity. *Predictive spectral spatial preprocessing for a multiclass Brain-computer interface*. In Proceedings of the 2010 International Joint Conference on Neural Networks (IJCNN), Barcelona, Spain, 2010. (Cited on page 17.)

[Cristianini 2001] Nello Cristianini, John Shawe-Taylor, André Elisseeff et Jaz S. Kandola. *On Kernel-Target Alignment*. In Thomas G. Dietterich, Suzanna Becker et Zoubin Ghahramani, editeurs, NIPS, pages 367–373. MIT Press, 2001. (Cited on page 69.)

[D. 2007] Krusienski D., S. Gerwin, McFarland D.J. et J.R. Wolpaw. *A mu-rhythm matched filter for continuous control of a brain-computer interface*. IEEE Transactions on Biomedical Engineering, vol. 54, no. 2, pages 273–280, 2007. (Cited on page 24.)

[Dai 2007] G. Dai et D. Yeung. *Kernel selection for semi-supervised kernel machines*. In Proceedings of the 24th International Conference on Machine Learning (ICML 07), 2007. (Cited on pages 42 and 53.)

[Darvas 2010] F. Darvas, R. Scherer, J. Ojemann, R. Rao, K. Miller et L. Sorensen. *High gamma mapping using EEG*. Neuroimage, vol. 49, no. 1, pages 930–938, 2010. (Cited on page 18.)

[Devlaminck 2009] D. Devlaminck, B. Wyns, L. Boullart, P. Santens et G. Otte. *Brain-Computer Interfaces: from theory to practice*. In ESANN'2009 proceedings, European Symposium on Artificial Neural Networks - Advances in Computational Intelligence and Learning., Bruges, Belgium., 2009. (Cited on page 24.)

[Dias 2005] N. Dias, P. Mendes et J. Correia. *Subject age in P300 BCI*. In Proceedings of the 2nd Internation IEEE EMBS Conference on Neural Engineering, pages 579–582, 2005. (Cited on page 25.)

[Dietterich 1998] T. Dietterich. *Approximate statistical tests for comparing supervised classification learning algorithms*. Neural Computation, vol. 10, no. 7, pages 1895–1923, 1998. (Cited on pages 100 and 101.)

[Duda 2000] R. Duda, P. Hart et D. Stork. Pattern classification. 2000. (Cited on page 19.)

[Everitt 1977] B. Everitt. The analysis of contingency tables. Chapman and Hall, London, 1977. (Cited on page 100.)

[Fabiani 2004] G. Fabiani, D. McFarland, Wolpaw J.R et G. Pfurtscheller. *Conversion of EEG activity into cursor movement by a brain–computer interface (BCI)*. IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 12, no. 3, pages 331–338, 2004. (Cited on page 18.)

[Fan 2005] R. Fan, P. Chen et C. Lin. *Working set selection using second order information for training support vector machines*. Journal of Machine Learning Research, vol. 2005, no. 6, pages 1889–1918, 2005. (Cited on pages 77 and 78.)

[Farina 2007] D. Farina, O. Nascimento, M. Lucas et C. Doncarli. *Optimization of wavelets for classification of movement-related cortical potentials generated by variation of force-related parameters*. Journal of Neuroscience Methods, vol. 162, no. 1-2, pages 357–363, 2007. (Cited on page 17.)

[Farwell 1998] L. Farwell et E. Donchin. *Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials*. Electroencephalogr Clin Neurophysiol, vol. 70, pages 510–23, 1998. (Cited on page 25.)

[Flamary 2011] R. Flamary, D. Tuis, B. Labbé, G. Camps-valls et A. Rakotomamonjy. *Large margin filtering*. IEEE Transactions Signal Processing, 2011. (Cited on page 17.)

[Freund 2003] Y. Freund, R. Iyer, R. Schapire et Y. Singer. *An efficient boosting algorithm for combining preferences*. Journal of Machine Learning Research, vol. 4, pages 933–969, 2003. (Cited on page 19.)

[Friedman 1997] J. Friedman. *On bias, variance, 0/1-loss, and the curse-of-dimensionality*. Data Mining and Knowledge Discovery, vol. 1, no. 1, pages 55–77, 1997. (Cited on page 38.)

[Friedman 2010] J. Friedman, T. Hastie et R. Tibshirani. *A note on the group lasso and a sparse group lasso*. Arxiv preprint arXiv:1001.0736, 2010. (Cited on page 30.)

[Friman 2007] O. Friman, T. Luth, I. Volosyak et A. Graser. *Spelling with Steady-state visual evoked potentials*. In Proceedings of 3rd International IEEE/EMBS Conference on Neural Engineering, pages 354–357, 2007. (Cited on page 20.)

[Gao 2003] X. Gao, D. Xu, M. Cheng et S. Gao. *A BCI-based environmental controller for the motion-disabled*. IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 11, no. 2, pages 137–140, 2003. (Cited on page 24.)

[Garcia 2003] G. Garcia, T. Ebrahimi et J. Vesin. *Support vector EEG classification in the Fourier and time-frequency correlation domains*. In Proceedings of 1st International IEEE EMBS Conference on Neural Engineering, pages 591–594, 2003. (Cited on page 20.)

[Geddes 2003] L. Geddes et R. Roeder. *Criteria for the selection of materials for implanted electrodes*. Annals of Biomedical Engineering, vol. 31, no. 7, pages 879–890, 2003. (Cited on page 37.)

[Georgopoulos 1986] A. Georgopoulos, A. Schwartz et R. Keriner. *Neuronal population coding of movement direction*. Science, vol. 233, 1986. (Cited on page 16.)

[Gerven 2009] V.M. Gerven, Farquhar J., R. Schaefer, R. Vlek, J. Geuze, A. Nijholt, N. Ramsey, P. Haselager, L. Vuurpijl, S. Gielen et P. Desain. *Topical review: the brain-computer interface cycle*. Journal of Neural Engineering, vol. 6, no. 4, page 041001, 2009. (Cited on pages 18 and 20.)

[Goldberg 2009] A. Goldberg, X. Zhu, A. Singh, Z. Xu et R. Nowak. *Multi-manifold semi-supervised learning*. Journal of Machine Learning Research, vol. 5, 2009. (Cited on pages 42, 53 and 54.)

[Gonen 2008] M. Gonen et E. Alpaydin. *Localized multiple kernel learning*. In Proceedings of the 25th International Conference on Machine Learning (ICML 2008), Helsinki, Finland, 2008. (Cited on pages 70 and 74.)

[Gönen 2011] Mehmet Gönen et Ethem Alpaydin. *Multiple Kernel Learning Algorithms*. Journal of Machine Learning Research, vol. 12, pages 2211–2268, 2011. (Cited on page 72.)

[Gu 2009] Y. Gu. *Decoding of movement characteristics for Brain computer interfaces application*. PhD thesis, Aalborg University, Denmark, 2009. (Cited on page 24.)

[Guan 2004] C. Guan, M. Thulasidas et J. Wu. *High performance P300 speller for brain–computer interface*. In Processing IEEE International Workshop on Biomedical Circuits and Systems (Singapore), pages S3–5, 2004. (Cited on page 18.)

[Guger 2001] C. Guger, A. Schlögl, C. Neuper, D. Walterspacher, T. Strein et G. Pfurtscheller. *Rapid prototyping of an EEG-based Brain-computer interface (BCI)*. IEEE Transactions on Rehabilitation Engineering, vol. 9, no. 1, pages 49–58, 2001. (Cited on page 21.)

[Guyon 2010] I. Guyon, A. Saffari, G. Dror et G. Cawley. *Model selection: beyond the Bayesian/Frequentist divide*. Journal of Machine Learning Research, vol. 11, pages 61–87, 2010. (Cited on page 28.)

[Hammon 2007] P. Hammon et R. Virginia. *Preprocessing and Meta-classification for Brain-computer interfaces.* IEEE Transactions on Biomedical Engineering, vol. 54, no. 3, pages 518–525, 2007. (Cited on page 18.)

[Haselsteiner 2000] E. Haselsteiner et G. Pfurtscheller. *Using time-dependent neural networks for EEG classification.* IEEE Transactions on Rehabilitation Engineering, vol. 8, no. 4, pages 457–463, 2000. (Cited on page 19.)

[Hashimoto 2010] Y. Hashimoto, J. Ushiba, A. Kimura, M. Liu et Y. Tomita. *Change in brain activity through virtual reality- based brain-machine communication in a chronic tetraplegic subject with muscular dystrophy.* BMC Neuroscience, vol. 11, 2010. (Cited on page 24.)

[Hastie 2009] T. Hastie, R. Tibshirani et J. Friedman. The elements of statistical learning: data mining, inference, and prediction. Springer, 2009. (Cited on pages 19, 28 and 30.)

[He 2010] L. He, Z. Gu, Y. Li et Z. Yu. *Feature extraction with multiscale autoregression of multichannel time series for P300 speller BCI.* In Proceedings of 2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), pages 610–613, 2010. (Cited on page 25.)

[Hinterberger 2004a] T. Hinterberger, S. Schmidt, N. Neumann, J. Mellinger, B. Blankertz, G. Curio et N. Birbaumer. *Brain-computer communication and slow cortical potentials.* IEEE Transactions on Biomedical Engineering, vol. 51, no. 6, pages 1011–1018, 2004. (Cited on page 25.)

[Hinterberger 2004b] T. Hinterberger, N. Weiskopf, R. Veit, B. Wilhelm, E. Betta et N. Birbaumer. *An EEG-driven Brain-computer interface combined with functional magnetic resonance imaging (fMRI).* IEEE Transactions on Biomedical Engineering, vol. 51, no. 6, pages 971–974, 2004. (Cited on page 25.)

[Hinton 1999] G. Hinton et T. Sejnowski. Unsupervised learning: foundations of neural computation. MIT Press, 1999. (Cited on page 28.)

[Hinton 2006] G.E. Hinton, S. Osindero et Y.W. Teh. *Learning multiple layers of representation.* Neural Computation, vol. 18, pages 1527–1554, 2006. (Cited on pages 34 and 35.)

[Hjorth 1975] B Hjorth. *An on-line transformation of EEG scalp potentials into orthogonal source derivations.* Clinical Neurophysiology, vol. 39, pages 526–30, 1975. (Cited on page 18.)

[Hoffmann 2008] U. Hoffmann, J.M. Vesin T. Ebrahimi et K. Diserens. *An efficient P300-based brain–computer interface for disabled subjects.* Journal of Neuroscience Methods, vol. 167, pages 115–125, 2008. (Cited on page 17.)

[Hollander 1999] M. Hollander et D.A. Wolfe. Nonparametric statistical methods. J. Wiley, New York, USA, 1999. (Cited on page 102.)

[Holzner 2009] C. Holzner, C. Guger, G. Edinger, C. Gronegress et M. Slater. *Virtual smart home controlled by thoughts.* In 18th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises, pages 236–239, Groningen, 2009. (Cited on page 20.)

[Hoya 2003] T. Hoya, G. Hori, H. Bakardjian, T. Nishimura, T. Suzuki, Y. Miyawaki, A. Funase et J. Cao. *Classification of single trial EEG signals by a combined principal + independent component analysis and probabilistic neural network approach.* In Proceedings of ICA2003, pages 197–202, 2003. (Cited on page 19.)

[Huan 2004] N. Huan et R. Palaniappan. *Neural network classification of autoregressive features from electroencephalogram signals for brain–computer interface design.* Journal of Neural Engineering, vol. 1, pages 142–150, 2004. (Cited on page 19.)

[Huang 2009] J. Huang, T. Zhang et D. Metaxas. *Learning with structured sparsity.* In ICML '09 Proceedings of the 26th Annual International Conference on Machine Learning, Montreal, Canada, 2009. (Cited on page 30.)

[Jin 2010] R. Jin, S. Hoi et T. Yang. *Online multiple kernel learning: algorithms and mistake bounds.* In Proceedings of the 21st International Conference on Algorithmic Learning Theory, pages 390–404, Canberra, Australia, 2010. (Cited on page 70.)

[Joachims 1999] T. Joachims. *Transductive inference for text classification using support vector machines.* In Proceedings of the 16th International Conference on Machine Learning, pages 200–209, San Francisco, USA, 1999. (Cited on pages 41 and 43.)

[Joachims 2003] T. Joachims. *Transductive learning via spectral graph partitioning.* In Proceedings of the 20th International Conference on Machine Learning, volume 20, page 290, Washington, USA, 2003. (Cited on page 41.)

[Kachenoura 2008] Amar Kachenoura, Laurent Albera, Lotfi Senhadji et Pierre Comon. *ICA: a potential tool for BCI systems.* IEEE Signal Processing Magazine, vol. 25, no. 1, pages 57–68, 2008. (Cited on page 18.)

[Kaper 2004] M. Kaper, P. Meinicke, U. Grossekathoefer, T. Lingner et H. Ritter. *BCI competition 2003-data set IIb: Support vector machines for the P300 speller paradigm.* IEEE Transactions on Biomedical Engineering, vol. 51, no. 6, pages 1073–1076, 2004. (Cited on page 19.)

[Kauhanen 2006] L. Kauhanen, T. Palomäki, P. Jylänki, F. Aloise, M. Nuttin et J.R. Millan. *Haptic feedback compared with visual feedback for BCI.* In Proceedings of the 3rd International Brain-Computer Interface Workshop & Training Course 2006, pages 66–67, Graz, Austria, 2006. (Cited on page 21.)

[Kennedy 1998] P. Kennedy et R. Bakay. *Restoration of neural output from a paralyzed patient by a direct brain connection.* NeuroReport, no. 9, pages 1707–1711, 1998. (Cited on page 16.)

[Kenneth 2010] E. Kenneth, K. Mikko et D. Calhoun. *The sixth annual MLSP competition, 2010.* In 2010 IEEE International Workshop on Machine Learning for Signal Processing (MLSP), Kittila, Finland, 2010. (Cited on pages 89, 92 and 94.)

[Kloft 2009] M. Kloft, U. Brefeld, S. Sonnenburg, P. Laskov, K.R. Müller et A. Zien. *Efficient and accurate lp-norm multiple kernel learning.* In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams et A. Culotta, editeurs, Advances in Neural Information Processing Systems 22, pages 997–1005. MIT Press, 2009. (Cited on pages 70 and 73.)

[Kloft 2011] M. Kloft, B. Ulf, S Sonnenburg et A. Zien. *Lp-norm multiple kernel learning.* Journal of Machine Learning Research, vol. 12, pages 953–997, 2011. (Cited on pages 46, 70, 72, 73 and 74.)

[Kotchoubey 2001] B. Kotchoubey, U. Strehl, C. Uhlmann, S. Holzafei, M. König, W. Fröscher, V. Blankenhorn et N. Birbaumer. *Modification of slow cortical potentials in patients with refractory epilepsy: a controlled outcome study.* Epilepsia, vol. 42, no. 3, pages 406–416, 2001. (Cited on page 25.)

[Krusienski 2011] D.J. Krusienski, M.G. Wentrup, F. Galan, D. Coyle, K.J. Miller, E. Forney et C.W. Anderson. *Critical issues in state-of-the-art Brain-computer interface signal processing.* Journal of Neural Engineering, vol. 8, page 025002, 2011. (Cited on page 38.)

[Kübler 2001] A. Kübler, B. Kotchoubey, J.R. Wolpaw et N. Birbaumer. *Brain–computer communication: unlocking the locked Psychol.* Psychological Bulletin, vol. 127, no. 3, pages 358–375, 2001. (Cited on page 24.)

[Labbé 2010] B. Labbé, X. Tian et A. Rakotomamonjy. *MLSP Competition, 2010: Description of third place method*. In Proceedings of IEEE International Workshop on Machine Learning for Signal Processing, 2010. (Cited on pages 7, 19 and 89.)

[Lalor 2004] E. Lalor, S. P. Kelly, C. Finucane, R. Burke, R. B. Reilly et G. McDarby. *Brain computer interface based on the steady-state VEP for immersive gaming control*. Biomed Tech, vol. 49, no. 1, pages 63–64, 2004. (Cited on pages 19 and 24.)

[Lanckriet 2004a] G. Lanckriet, N. Cristianini, P. Bartlett, L. Ghaoui et M. Jordan. *Learning the kernel matrix with semidefinite programming*. Journal of Machine Learning Research, vol. 5, pages 27–72, 2004. (Cited on pages 69 and 71.)

[Lanckriet 2004b] G. Lanckriet, N. Cristianini, P. Bartlett, L.E. Ghaoui et M.I. Jordan. *Learning the kernel matrix with semi-definite programming*. Journal of Machine Learning Research, vol. 5, pages 27–72, 2004. (Cited on page 69.)

[Laubach 2000] M. Laubach, J. Wessberg et M. Nicolelis. *Cortical ensemble activity increasingly predicts behaviour outcomes during learning of a motor task*. Nature, vol. 405, no. 6786, pages 567–570, 2000. (Cited on page 16.)

[Lee 2003] H. Lee et S. Choi. *PCA+HMM+SVM for EEG pattern classification*. In Proceedings of the 7th International Symposium on Signal Processing and its Applications, volume 1, pages 541–544, 2003. (Cited on page 19.)

[Lee 2010] P. Lee, J. Sie, Y. Liu, C. Wu, M. Lee, C. Shu, P. Li, C. Sun et Shyu K. *An SSVEP-actuated brain computer interface using phase-tagged flickering sequences: a cursor system*. Annals of Biomedical Engineering (2010), vol. 38, no. 7, pages 2383–2397, 2010. (Cited on page 24.)

[Leuthardt 2004] E.C. Leuthardt, G. Schalk, J.R. Wolpaw, J.G. Ojemann et D.W. Moran. *A brain-computer interface using electrocorticographic signals in humans*. Journal of Neural Engineering, vol. 1, pages 63–71, 2004. (Cited on page 16.)

[Li 2008] Y. Li, C. Guan, H. Li et Z. Chin. *A self-training semi-supervised SVM algorithm and its application in an EEG-based brain computer interface speller system*. Pattern Recognition Letters, vol. 29, pages 1285–1294, 2008. (Cited on pages 19, 20, 41 and 46.)

[Li 2010a] J. Li et S. Sun. *Nonlinear combination of multiple kernels for support vector machines*. In Proceedings of 2010 Inernational Conference on Pattern Recognition, 2010. (Cited on page 74.)

[Li 2010b] Y Li, J Long, T Yu, Z Yu, C Wang, H Zhang et C Guan. *An EEG-based BCI system for 2-D cursor control by combining Mu/Beta rhythm and P300 potential*. IEEE Trans Biomed Eng., vol. 57, no. 10, pages 495–505, 2010. (Cited on page 20.)

[Liao 2007] X. Liao, D. Yao et C. Li. *Transductive SVM for reducing the training effort in BCI*. Journal of Neural Engineering, no. 4, pages 246–254, 2007. (Cited on pages 20, 41, 46 and 62.)

[Liaw 2002] A. Liaw et M. Wiener. *Classification and regression by random forest*. R news, vol. 2, no. 3, pages 18–22, 2002. (Cited on page 19.)

[Liefhold 2007] C. Liefhold, G. Moritz, G. Klaus et B. Martin. *Comparison of adaptive spatial filters with heuristic and optimized region of interest for EEG-based Brain-computer-interfaces*. Lecture Notes in Computer Science, pages 274–283, 2007. (Cited on page 18.)

[Lin 2011] C. Lin, C. Euler, A. Mekhtarian, A. Gil, L. Hern, D. Prince, Y. Shen et J. Horvath. *A brain-computer interface for intelligent wheelchair mobility*. In Health Care Exchanges (PAHCE), 2011 Pan American, 2011. (Cited on page 20.)

[Liu 2010] T. Liu, L. Goldberg, S. Gao et B. Hong. *An online brain–computer interface using non-flashing visual evoked potentials*. Journal of Neural Engineering, vol. 7, no. 3, page 036003, 2010. (Cited on page 24.)

[Lotte 2007a] F. Lotte, M. Congedo, A. Lecuyer, F. Lamarche et B. Arnaldi. *A review of classification algorithms for EEG-based brain–computer interfaces*. Journal of Neural Engineering, vol. 4, no. 2, page R1, 2007. (Cited on pages 15, 19, 30 and 93.)

[Lotte 2007b] F Lotte, M Congedo, A Lecuyer, F Lamarche et B Arnaldi. *A review of classification algorithms for EEG-based Brain-computer interfaces*. Journal of Neural Engineering, vol. 4, 2007. (Cited on pages 25 and 91.)

[Lotte 2008] F. Lotte. *Study of electroencephalographic signal processing and classification techniques towards the use of Brain-computer interfaces in virtual reality applications*. PhD thesis, Institut National des Sciences Appliquées de Rennes, France, 2008. (Cited on page 21.)

[Lotte 2009] F. Lotte et C. Guan. *An efficient P300-based Brain-computer interface with minimal calibration time*. In Assistive Machine Learning for People with Disabilities symposium (NIPS'09 Symposium), 2009. (Cited on page 25.)

[Lukito 2009] S. Lukito, S. Halder, P. Bretherton, C. Vogele et A. Kübler. *The effect of emotions on P300 brain-computer interface (BCI) performance*. In Proceedings of COST Neuromath Workshop, 2009. (Cited on page 96.)

[Ma 2003] J. Ma, J. Theier et S. Perkins. *Accurate on-line support vector regression*. Neural Computation, vol. 15, no. 11, pages 2683–2703, 2003. (Cited on page 70.)

[MacDonald 2006] P. MacDonald et P. Rorsman. *Oscillations, intercellular coupling, and insulin secretion in pancreatic β cells*. PLoS Biol, vol. 4, no. 2, page e49, 2006. (Cited on page 16.)

[Mak 2009] J. Mak et J.R. Wolpaw. *Clinical applications of Brain-computer interfaces: current state and future prospects*. IEEE Reviews in Biomed Engineering, vol. 2, pages 187–199, 2009. (Cited on pages 19 and 20.)

[Malechka 2011] T. Malechka et Żygierewiczc J. *ERD/ERS BCI training on the basis of a labyrinth application*. Journal of Bioelectromagnetism, vol. 13, 2011. (Cited on page 24.)

[Mallapragada 2009] P. Mallapragada, R. Jin, A. Jain et Y. Liu. *SemiBoost: boosting for semi-supervised learning*. IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI), vol. 31, no. 11, pages 2000–2014, 2009. (Cited on pages 42, 53 and 54.)

[Manoilov 2006] P. Manoilov. *EEG power spectrum analysis during mental task performance*. In Proceedings of International Conference on Computer Systems and Technologies, 2006. (Cited on page 17.)

[Martinez 2007] P. Martinez, H. Bakardjian et A. Cichocki. *Fully-online, multi-command brain computer interface with visual neurofeedback using SSVEP paradigm*. Computational Intelligence and Neuroscience, vol. 2007, 2007. (Cited on page 24.)

[Martins 2010] A. Martins, M. Figueiredo, P. Agular, N. Smith et E. Xing. *Online multiple kernel learning for structured prediction*. In Proceedings of NIPS 2010 Workshop, 2010. (Cited on pages 70 and 73.)

[McFarland 1997] D.J. McFarland, L. McCane, S. David et J.R. Wolpaw. *Spatial filter selection for EEG-based communication*. Electroencephalogr Clin Neurophysiol, vol. 103, pages 386–94, 1997. (Cited on page 18.)

[McFarland 2006] D.J. McFarland, D. Krusienski et J.R. Wolpaw. *Brain-computer interface signal processing at the Wadsworth center: mu and sensorimotor beta rhythms.* Progress in Brain Research, vol. 159, pages 411–419, 2006. (Cited on pages 24 and 59.)

[McFarland 2008] D.J. McFarland et J.R. Wolpaw. *Sensorimotor rhythm-based brain-computer interface (BCI): model order selection for autoregressive spectral analysis.* Journal of Neural Engineering, vol. 5, no. 2, pages 155–162, 2008. (Cited on page 19.)

[Middendorf 2000a] M. Middendorf, G. McMillan, G. Calhoun et K. Jones. *Brain–computer interfaces based on the steady-state visual-evoked response.* IEEE Transactions on Rehabilitation Engineering, vol. 8, no. 2, pages 211–214, 2000. (Cited on page 24.)

[Middendorf 2000b] M. Middendorf, G. McMillan, G. Calhoun et K. Jones. *Brain-computer interfaces based on the steady-state visual-evoked response.* IEEE Transactions on Rehabilitation Engineering, vol. 8, no. 2, pages 211–214, 2000. (Cited on page 20.)

[Millan 2007] J.R. Millan, A. Buttfield, C. Vidaurre, M. Krauledat, A. Schogl, P. Shenoy, B. Blankertz, R. Rao, R. Cabeza et G. Pfurtscheller. *Adaptation in Brain-computer interfaces.* Toward Brain-Computer Interfacing, pages 303–326, 2007. (Cited on page 38.)

[Mourino 2002] J. Mourino, S. Chiappa, R. Jane et J.R. Millan. *Evolution of the mental states operating a brain-computer interface.* In Proceedings of the International Federation for Medical and Biological Engineering, volume 3, pages 600–601, 2002. (Cited on page 37.)

[Mühl 2010] C. Mühl, H. Gürkök, D. Bos, M. Thurlings*et al. Bacteria hunt: a multimodal, multiparadigm BCI game.* In Workshop Report for the Enterface Workshop, Genova, Italy, 2010. (Cited on page 20.)

[Müller 2003] K.R. Müller, C.W. Anderson et G.E. Birch. *Linear and non-linear methods for brain-computer interfaces.* IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 11, no. 2, pages 165–169, 2003. (Cited on page 93.)

[Müller 2005] P.R. Müller, R. Scherer, G. Pfurtscheller et R. Rupp. *EEG-based neuroprosthesis control: a step towards clinical practice.* Neuroscience Letters, vol. 382, no. 1-2, pages 169–174, 2005. (Cited on page 20.)

[Naeem 2006] M. Naeem, C. Brunner, R. Leeb, B. Graimann et G. Pfurtscheller. *Seperability of four-class motor imagery data using independent components analysis.* Journal of Neural Engineering, vol. 3, no. 3, pages 208–16, 2006. (Cited on page 18.)

[Nasihatkon 2009] B. Nasihatkon, R. Boostani et M.Z. Jahromi. *An efficient hybrid linear and kernel CSP approach for EEG feature extraction.* Neurocomputing, vol. 73, no. 1-3, pages 432–437, 2009. (Cited on page 19.)

[Neuper 1999] C. Neuper, A. Schlögl et G. Pfurtscheller. *Enhancement of left-right sensorimotor EEG differences during feedback-regulated motor imagery.* Journal of Clinical Neurophysiology, vol. 16, no. 4, page 373, 1999. (Cited on page 21.)

[Neuper 2005] C. Neuper, Grabner R., F. Andreas et Neubauer A. *Long-term stability and consistency of EEG event-related (de-)synchronization across different cognitive task.* Clinical Neurophysiology, vol. 116, no. 7, pages 1681–1694, 2005. (Cited on page 24.)

[Nicolelis 2003] M. Nicolelis. *Brain–machine interfaces to restore motor function and probe neural circuits.* Nature Reviews Neuroscience, vol. 4, no. 5, pages 417–422, 2003. (Cited on page 16.)

[Nijboer 2009] F. Nijboer, S.P. Carmien, E. Leon, F.O. Morin, R. Koene et U. Hoffmann. *Affective Brain-computer interfaces: psychophysiological markers of emotion in healthy persons and in persons with amyotrophic lateral sclerosis.* In Proceedings of 3rd International Conference on Affective

Computing and Intelligent Interaction and Workshops, 2009, ACII 2009, 2009. (Cited on pages 37 and 95.)

[Nocedal 2006] J. Nocedal et S.J. Wright. Numerical optimization. Springer, 2006. (Cited on page 43.)

[Obermaier 2001] B. Obermaier, C. Neuper, C. Guger et G. Pfurtscheller. *Information transfer rate in a five-classes brain-computer interface*. IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 9, no. 3, 2001. (Cited on page 20.)

[Ochoa 2002] J.B. Ochoa. *EEG signal classification for brain computer interface applications*. Rapport technique, Ecole Polytechnique Federale De Lausanne, 2002. (Cited on page 17.)

[Odom 2004] J Vernon Odom, Michael Bach, Colin Barber, Mitchell Brigell, Michael F. Marmor, Alma Patrizia Tormene, Graham E. Holder et Vaegan. *Visual evoked potentials standard (2004)*. Documenta Ophthalmologica, vol. 108, pages 115–123, 2004. (Cited on page 23.)

[Omar 2011] C. Omar, A. Akce, M. Johnson, T. Bretl, R. Ma, E. Maclin, M. McCormick et T. Coleman. *A feedback information-theoretic approach to the design of brain-computer interfaces*. International Journal of Human-computer Interaction, vol. 27, no. 1, pages 5–23, 2011. (Cited on page 21.)

[Orabona 2010] F. Orabona, J. Luo et B. Caputo. *Online-batch strongly convex multi kernel learning*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 787–794, San Francisco, CA, USA, 2010. (Cited on pages 70, 73 and 85.)

[Panicker 2010] R.C. Panicker, S. Puthusserypady et Y. Sun. *Adaptation in P300 Brain–computer interfaces: a two–classifier cotraining approach*. IEEE Transactions on Biomedical Engineering, vol. 57, no. 12, pages 2927–2935, 2010. (Cited on pages 20 and 41.)

[Pfurtscheller 1997] G. Pfurtscheller, C. Neuper, D. Flotzinger et M. Pregenzer. *EEG-based discrimination between imagination of right and left hand movement.* Electroencephalography and clinical Neurophysiology, vol. 103, no. 6, pages 642–51, 1997. (Cited on pages 18, 21 and 24.)

[Pfurtscheller 1998] G. Pfurtscheller, C. Neuper, A. Schlogl et K. Lugger. *Separability of EEG Signals Recorded During right and left motor imagery using adaptive autoregressive parameters*. IEEE Transactions on Rehabilitation Engineering, vol. 6, no. 3, pages 316–325, 1998. (Cited on page 24.)

[Pfurtscheller 2003a] G. Pfurtscheller, G.R. Müller, J. Pfurtscheller, H. Gerner et R. Rupp. *Thought control of functional electrical stimulation to restore hand grasp in a patient with tetraplegia*. Neuroscience Letters, vol. 351, no. 1, pages 33–36, 2003. (Cited on page 20.)

[Pfurtscheller 2003b] G. Pfurtscheller, C. Neuper, G.R. Müller, B. Obermaier, G. Krausz, A. Schlögl, R. Scherer, B. Graimann, C. Keinrath, D. Skliris, M. Wortz, G. Supp et C. Schrank. *Graz-BCI: state of the art and clinical applications*. IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 11, no. 2, pages 77–80, 2003. (Cited on page 24.)

[Pfurtscheller 2004] G. Pfurtscheller. *Brain-computer interface : State of the art and future prospects*. In Proceedings of European Signal Processing Conference, pages 509–510, Vienna , Autriche, 2004. (Cited on page 15.)

[Pfurtscheller 2006a] G. Pfurtscheller, G.R. Müller, A. Schlogl, B. Graimann, R. Scherer, R. Leeb, C. Brunner, C. Keinrath, F. Lee, G. Townsend, C. Vidaurre et C. Neuper. *5 years of BCI research at Graz University of Technology: Current projects*. IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 14, no. 2, pages 205–210, 2006. (Cited on page 20.)

[Pfurtscheller 2006b] G. Pfurtscheller et C. Neuper. *Future prospects of ERD/ERS in the context of brain–computer interface (BCI) developments*. Progress in Brain Research, vol. 159, pages 433–437, 2006. (Cited on page 24.)

[Pfurtscheller 2010] G. Pfurtscheller, B. Allison, C. Brunner, G. Bauernfeind, T. Solis-Escalante, R. Scherer, T. O. Zander, G. Mueller-Putz, C. Neuper et N. Birbaumer. *The hybrid BCI.* Frontiers in Neuroscience, vol. 4, 2010. (Cited on page 16.)

[Piccione 2006] F. Piccione, K. Priftis, P. Tonin, D. Vidale, R. Furlan, M. Cavinato, A. Merico et L. Piron. *Task and stimulation paradigm effects in a P300 brain computer interface exploitable in a virtual environment: a pilot study.* PsychNology Journal, vol. 6, no. 1, pages 99–108, 2006. (Cited on page 25.)

[Popdscu 2007] F. Popdscu, S. Fazli, Y. Badower, B. Blankertz et K.R. Müller. *Single trial classification of motor imagination using 6 dry EEG electrodes.* PLoS ONE, vol. 2, no. 7, page e637, 2007. (Cited on page 37.)

[Qin 2007] J. Qin, Y. Li et W. Sun. *A semisupervised support vector machines algorithm for BCI systems.* Computational Intelligence and Neuroscience, vol. 2007, pages 1687–5265, 2007. (Cited on pages 20, 24, 39, 41, 46, 59, 60 and 61.)

[Qin 2010] H. Qin, D. Dou et Y. Fang. *Financial forecasting with gompertz multiple kernel learning.* In Proceedings of the 10th IEEE International Conference on Data Mining, Sydney, Australia, 2010. (Cited on page 74.)

[Rakotomamonjy 2008a] A. Rakotomamonjy, F. Bach, S. Canu et Y. Grandvalet. *SimpleMKL.* Journal of Machine Learning Research, vol. 9, pages 2491–2521, 2008. (Cited on pages 39, 42, 46, 50, 52, 69, 72 and 73.)

[Rakotomamonjy 2008b] A. Rakotomamonjy et V. Guigue. *BCI competition III: dataset II - ensemble of SVMs for BCI P300 speller.* IEEE Transactions on Biomedical Engineering, vol. 55, no. 3, pages 1147–1154, 2008. (Cited on pages 19 and 91.)

[Ramoser 2000] H. Ramoser, J. Muller-Gerking et G. Pfurtscheller. *Optimal spatial filtering of single trial EEG during imagined hand movement.* IEEE Transactions on Rehabilitation Engineering, vol. 8, no. 4, pages 441–446, 2000. (Cited on page 60.)

[Rebsamen 2007a] B. Rebsamen, E. Burdet, C. Guan, Teo. C., Zeng Q., Ang M. et Laugier C. *Controlling a wheelchair using a BCI with low information transfer rate.* In Proceedings of the 2007 IEEE 10th International Conference on Rehabilitation Robotics, pages 1003–1008, 2007. (Cited on page 20.)

[Rebsamen 2007b] B. Rebsamen, E. Burdet, C. Guan, H. Zhang, C. Teo, Q. Zeng, C. Laugier et Marcelo H. *Controlling a wheelchair indoors using thought.* Intelligent Systems, vol. 22, no. 2, pages 18–24, 2007. (Cited on pages 20 and 25.)

[Reuderink 2008] B. Reuderink. *Games and Brain-computer interfaces: the state of the art.* Rapport technique TR-CTI, Centre for Telematics and Information Technology, University of Twente, 2008. (Cited on page 20.)

[Rockafellar 1996] R.T Rockafellar. Convex analysis. Princeton University Press, 1996. (Cited on page 49.)

[Rothman 1970] H. Rothman, H. Davis et I. Hay. *Slow evoked cortical potentials and temporal features of stimulation.* Electroencephalography and Clinical Neurophysiology, vol. 29, no. 3, pages 225–232, 1970. (Cited on page 19.)

[Rothman 2010] L. Rothman. *Ensemble-based classifiers.* Artificial Intelligence, vol. 33, no. 1, pages 1–39, 2010. (Cited on page 19.)

[Rozenkrants 2008] B. Rozenkrants et J. Polich. *Affective ERP processing in a visual oddball task: arousal, valence, and gender.* Clinical Neurophysiology, vol. 119, no. 10, pages 2260–2265, 2008. (Cited on page 96.)

[Schalk 2004] G. Schalk, D.J. McFarland, T. Hinterberger, N. Birbaumer et J.R. Wolpaw. *BCI2000: a general-purpose brain-computer interface (BCI) system*. IEEE Transactions on Biomedical Engineering, vol. 51, no. 6, pages 1034–1043, 2004. (Cited on pages 21 and 26.)

[Schölkopf 1997] B. Schölkopf, A. Smola et K.R. Müller. *Kernel principal component analysis*. In Proceedings of the 7th International Conference, pages 583–588, Lausanne, Switzerland, 1997. (Cited on page 32.)

[Schölkopf 2002] B. Schölkopf et A.J. Smola. Learning with kernels support vector machines, regularization, optimization, and beyond. MIT Press, 2002. (Cited on pages 19, 31, 32, 33 and 69.)

[Seeger 2002] M. Seeger. *Learning with labeled and unlabeled data*. Rapport technique, Institute for ANC, Edinburgh, UK, 2002. (Cited on page 43.)

[Shalev-Shwartz 2011] Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro et Andrew Cotter. *Pegasos: primal estimated sub-gradient solver for SVM*. Math. Program., vol. 127, no. 1, pages 3–30, 2011. (Cited on page 70.)

[Shawe-Taylor 2004] J. Shawe-Taylor et N. Cristianini. Kernel methods for pattern analysis. Cambridge University Press, 2004. (Cited on page 31.)

[Shen 2009] H. Shen, L. Zhao, Y. Bian et L. Xiao. *Research on SSVEP-based controlling system of multi-DoF manipulator*. Lecture Notes in Computer Science, vol. 5553/2009, pages 171–177, 2009. (Cited on page 24.)

[Sindhwani 2005] V. Sindhwani, P. Niyogi et M. Belkin. *Beyond the point cloud: from transductive to semi-supervised learning*. In Proceedings of the 22nd International Conference on Machine Learning, pages 824–831, Bonn, Germany, 2005. ACM Press. (Cited on pages 45, 54, 55 and 56.)

[Sindhwani 2006] V. Sindhwani, S. Keerthi et O. Chapelle. *Deterministic annealing for semi-supervised kernel machines*. In Proceedings of the 23rd international conference on Machine learning, pages 841–848, New York, USA, 2006. ACM Press. (Cited on page 44.)

[Sitaram 2007] R. Sitaram, H. Zhang, C. Guan, M. Thulasidas, Y. Hoshi, A. Ishikawa, K. Shimizu et N. Birbaumer. *Temporal classification of multichannel near-infrared spectroscopy signals of motor imagery for developing a brain–computer interface*. Neuroimage, vol. 34, no. 4, pages 1416–1427, 2007. (Cited on pages 16 and 19.)

[Smola 2005] A. Smola, S. Vishwanathan et T. Hofmann. *Kernel methods for missing variables*. In Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, pages 325–332, 2005. (Cited on page 49.)

[Soekadar 2008] S. Soekadar, K. Haagen et N. Birbaumer. Brain-computer interfaces (bci): restoration of movement and thought from neuroelectric and metabolic brain activity, pages 229–252. Coordination: Neural, Behavioral and Social Dynamics. 2008. (Cited on pages 16, 20, 22 and 25.)

[Sonnenburg 2006] S. Sonnenburg, G. Rätsch, C. Schäfer et B. Schölkopf. *Large scale multiple kernel learning*. Journal of Machine Learning Research, vol. 7, no. 1, pages 1531–1565, 2006. (Cited on pages 69 and 72.)

[Subasi 2010] A. Subasi et M. Gursoy. *EEG signal classification using PCA, ICA, LDA and support vector machines*. Expert Systems with Applications, vol. 37, no. 12, pages 8659–8666, 2010. (Cited on page 19.)

[Szafranski 2008] M. Szafranski et A. Rakotomamonjy. *Composite kernel learning*. In Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 2008. (Cited on page 30.)

[Tao 1998] P. D. Tao et L. T. Hoai An. *DC optimization algorithms for solving the trust region subproblem*. SIAM Journal of Optimization, vol. 8, no. 2, pages 476–505, 1998. (Cited on pages 42, 48 and 49.)

[Tatum 2008] W. Tatum, A. Husain, S. Benbadis et P. Kaplan. Handbook of eeg interpretation. Demos Medical Publishing, 2008. (Cited on page 16.)

[Taylor 2003] D. Taylor, S. Tillery et A. Schwartz. *Information conveyed through brain-control: cursor versus robot.* IEEE Trans. on Neural Systems and Rehabilitation Engineering, vol. 11, no. 2, pages 195–199, 2003. (Cited on page 20.)

[Tian 2010] X. Tian, R. Hérault, G. Gasso et S. Canu. *Pré-apprentissage supervisé pour les réseaux profonds.* In Proceedings of Rfia 2010, 2010. (Cited on page 36.)

[Tian 2011] X. Tian, G. Gasso et S. Canu. *An inductive semi-supervised algorithm for BCIs.* International Journal of Bioelectromagnetism, vol. 13, no. 3, pages 117–118, 2011. (Cited on page 7.)

[Tian 2012] X. Tian, G. Gasso et S. Canu. *A multiple kernel framework for inductive semi-supervised SVM learning.* Neurocomputing, 2012. (Cited on pages 7, 20 and 36.)

[Tibshirani 1996] R. Tibshirani. *Regression shrinkage and selection via the Lasso.* Journal of the Royal Statistical Society. Series B (Methodological), vol. 58, no. 1, pages 267–288, 1996. (Cited on page 30.)

[Trejo 2006] L.J. Trejo, R. Rosipal et B. Matthews. *Brain-computer interfaces for 1-D and 2-D cursor control: designs using volitional control of the EEG spectrum or steady-state visual evoked potentials.* IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 14, no. 2, pages 225–229, 2006. (Cited on pages 20 and 24.)

[Tsui 2009] C. Tsui, J. Gan et S. Roberts. *A self-paced brain-computer interface for controlling a robot simulator: an online event labelling paradigm and an extended Kalman filter based algorithm for online training.* Medical and Biological Engineering and Computing, vol. 47, no. 3, pages 257–265, 2009. (Cited on page 37.)

[Ungureanu 2004] M. Ungureanu, C. Bigan, R. Strungaru et V. Lazarescu. *Independent component analysis applied in biomedical signal processing.* Measurement Science Review, vol. 4, no. 2, 2004. (Cited on page 18.)

[Vapnik 1977] V. Vapnik et A. Sterin. *On structural risk minimization or overall risk in a problem of pattern recognition.* Automation and Remote Control, vol. 10, pages 1495–1503, 1977. (Cited on pages 36, 41 and 43.)

[Vapnik 1995] V. Vapnik. The nature of statistical learning theory. Springer, 1995. (Cited on pages 27, 28, 30 and 33.)

[Varma 2009] M. Varma et B. Badu. *More generality in efficient multiple kernel learning.* In Proceedings of the 26th Annual International Conference on Machine Learning, pages 1065–1072, Montreal, Canada, 2009. (Cited on page 74.)

[Vaughan 2003] T. Vaughan, W. Heetderks, L.J. Trejo, W. Rymer, M. Weinrich, M. Moore, A. Kubler, B. Dobkin, N. Birbaumer, E. Donchin, E. Wolpaw et J.R. Wolpaw. *Brain-computer interface technology: a review of the second international meeting.* IEEE Transactions on Neural Systems and Rehabilitation Engeneering, vol. 11, no. 2, pages 94–109, 2003. (Cited on page 18.)

[Vishwanathan 2010] S. Vishwanathan, Z. Sun et N. Theera-Ampornpunt. *Multiple kernel learning and the SMO algorithm.* In Proceedings of 24th Annual Conference on Neural Information Processing Systems, Vancouver, Canada, 2010. (Cited on pages 39, 70, 73, 75 and 82.)

[von Luxburg 2011] U. von Luxburg et B. Schölkopf. Statistical learning theory: models, concepts, and results, pages 751–706. Elsevier North Holland, Amsterdam, Netherlands, 2011. (Cited on page 27.)

[Vuckovic 2008] A. Vuckovic et F. Sepulveda. *Quantification and visualisation of differences between two motor tasks based on energy density maps for brain–computer interface applications.* Clinical Neurophysiology, vol. 119, pages 446–458, 2008. (Cited on page 24.)

[Wang 2004a] T. Wang, J. Deng et B. He. *Classifying EEG-based motor imagery tasks by means of time-frequency synthesized spatial patterns.* Clinical Neurophysiology, vol. 115, no. 12, pages 2744–2753, 2004. (Cited on page 24.)

[Wang 2004b] Y. Wang, Z. Zhang, X. Gao et S. Gao. *Lead selection for SSVEP-based brain–computer interface.* In Proceedings of the 26th Annual International Conference of the IEEE EMB, San Francisco, USA, 2004. (Cited on page 18.)

[Wang 2005] C. Wang, C. Guan et H. Zhang. *P300 Brain-computer interface design for communication and control applications.* In Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference, Shanghai, China, 2005. (Cited on page 25.)

[Wang 2008] Y. Wang, X. Gao, B. Hong, C. Jia et S. Gao. *Brain–computer interfaces based on visual evoked potentials: feasibility of practical system designs.* Biomedical Engineering in China, pages 64–71, 2008. (Cited on page 24.)

[Wang 2009] J. Wang, X. Shen et W. Pan. *On efficient large margin semisupervised learning: method and theory.* Journal of Machine Learning Research, vol. 10, pages 719–742, 2009. (Cited on pages 44 and 45.)

[Wang 2010a] P. Wang, J. Shen et J. Shi. *P300 detection algorithm based on Fisher distance.* International Journal of Modern Education and Computer Science, no. 2, pages 9–17, 2010. (Cited on page 25.)

[Wang 2010b] Y. Wang, Y. Wang et T. Jung. *Visual stimulus design for high-rate SSVEP BCI.* Electronics Letters, vol. 46, no. 15, pages 1057–1058, 2010. (Cited on page 24.)

[Wasserman 1989] P.D. Wasserman. Neural computing: theory and practice. Van Nostrand Reinhold Co., 1989. (Cited on page 19.)

[Wheldon 1998] T. Wheldon. Mathematical models in cancer research. Bristol: Adam hilger, 1998. (Cited on page 74.)

[Wolpaw 2002] J.R. Wolpaw, N. Birbaumer, D.J. McFarland, G. Pfurtscheller et T. Vaughan. *Brain-computer interfaces for communication and control.* Clinical Neurophysiology, vol. 113, pages 767–791, 2002. (Cited on pages 22, 23, 24 and 25.)

[Xu 2009] Z. Xu, R. Jin, J. Zhu, I. King, M. Lyu et Z. Yang. *Adaptive regularization for transductive support vector machine.* In Proceedings of the 23th Annual Conference on Neural Information Processing Systems, Vancouver, Canada, 2009. (Cited on page 53.)

[Xu 2010] Z. Xu, R. Jin, H. Yang, I. King et M. Lyu. *Simple and efficient multiple kernel learning by group lasso.* In Proceedings of the 27th International Conference on Machine Learning, Haifa, Israel, 2010. (Cited on pages 46 and 50.)

[Yang 2005] J. Yang, A. Frangi, J. Yang, D. Zhang et Z. Jin. *KPCA plus LDA: a complete kernel Fisher discriminant framework for feature extraction and recognition.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 2, pages 230–244, 2005. (Cited on page 32.)

[Yang 2007] L. Yang, J. Li et Y. Yao. *P300 detection algorithm based on wavelet decomposition and support vector machine.* Chinese Journal of Biomedical Engineering, vol. 26, no. 6, pages 804–809, 2007. (Cited on page 25.)

[Yazdani 2009] A. Yazdani, J. Lee et T. Ebrahimi. *Implicit emotional tagging of multimedia using EEG signals and brain computer interface.* In Proceedings of the first SIGMM workshop on Social media, New York, USA, 2009. (Cited on page 96.)

[Yuille 2001] A. L. Yuille et A. Rangarajan. *The Concave-convexe procedure*. In Proceedings of Advances in Neural Information Processing Systems, 2001. (Cited on pages 48 and 49.)

[Zander 2011] T.O. Zander, M. Lehne, K. Ihme, S. Jatzev, J. Correia, C. Kothe, B. Picht et F. Nijboer. *A dry EEG-system for scientific research and Brain–computer interfaces*. Frontiers in Neuroscience, 2011. (Cited on page 37.)

[Zhao 2008] B. Zhao, F. Wang et C. Zhang. *Cuts3vm: a fast semi-supervised svm algorithm*. In Proceedings Of The Acm Sigkdd International Conference On Knowledge Discovery And Data Mining, pages 830–838, New York, USA, 2008. (Cited on page 45.)

[Zhong 2008] M. Zhong, F. Lotte, M. Girolami et A. Lecuyer. *Classifying EEG for brain computer interfaces using Gaussian processes*. Pattern Recognition Letters, vol. 29, no. 3, 2008. (Cited on page 19.)

[Zhong 2009] J. Zhong, X. Lei et D. Yao. *Semi-supervised learning based on manifold in BCI*. Journal of Electronics Science and Technology of China, vol. 7, pages 22–26, 2009. (Cited on pages 20, 41, 46 and 62.)

[Zhu 2002] X. Zhu et Z. Ghahramani. *Learning from labeled and unlabeled data with label propagation*. Rapport technique CMU-CALD-02-107, Carnegie Mellon University, 2002. (Cited on page 41.)

[Zhuang 2011] J. Zhuang, I. Tsang et S. Hoi. *Two-layer multiple kernel learning*. In Proceedings of Interational Conference on Artificial Intelligence and Statistics, 2011. (Cited on page 74.)