

# Semi-supervised Mixture of Kernels via LPBoost Methods

Jinbo Bi Glenn Fung Murat Dundar Bharat Rao  
Computer Aided Diagnosis and Therapy Solutions  
Siemens Medical Solutions, Malvern, PA 19355  
jinbo.bi, glenn.fung, murat.dundar, bharat.rao@siemens.com

## Abstract

*We propose an algorithm to construct classification models with a mixture of kernels from labeled and unlabeled data. The derived classifier is a mixture of models, each based on one kernel choice from a library of kernels. The sparse-favoring 1-norm regularization method is employed to restrict the complexity of mixture models and to achieve the sparsity of solutions. By modifying the column generation boosting algorithm LPBoost to a more general linear programming formulation, we are able to efficiently solve mixture-of-kernel problems and automatically select kernel basis functions centered at labeled data as well as unlabeled data. The effectiveness of the proposed approach is proved by experimental results on benchmark datasets.*

## 1 Introduction

Recent years have seen considerable interests in learning with labeled and unlabeled data since experiments for labeling data are often expensive while unlabeled data is easily available. Boosting algorithms are well-studied learning methodologies that construct classifiers in an incremental fashion by using a weighted “vote” of various simple models obtained from a weak learner. In this paper, we propose a boosting algorithm to construct classification models from labeled and unlabeled data based on kernel methods.

Kernel methods construct nonlinear models using linear learning algorithms by introducing a positive semidefinite kernel  $K$ . The choice of kernel is usually determined by predefining the type of kernel (e.g, RBF or polynomial kernels), and tuning the kernel parameters using cross-validation performance. Cross-validation is expensive and the resulting kernel is not guaranteed to be a good choice.

Recent work has attempted to design kernels that adapt to a particular task to be solved. For example, Lanckriet et al. [5] proposed the use of a linear combination of kernels  $K = \sum_p \mu_p K_p$  from a family of various kernel functions  $K_p$ . To ensure the positive semidefiniteness, the combina-

tion coefficients  $\mu_p$  are either simply required to be nonnegative or determined in the way such that the composite kernel is positive semidefinite, for instance, by solving a semidefinite program. Instead of forming new kernels, mixture-of-kernel models construct classifiers that are a mixture of models, each based on one kernel choice from a library of kernels [3, 2]. This paper extends the inductive learning model presented in [3] to semi-supervised learning problems with unlabeled data. The proposed approach makes use of various geometry of the feature spaces introduced by a family of kernels, and automatically determines the kernel basis functions to be used in the mixture model. Moreover, the unlabeled data is used jointly with the labeled data as possible centers for the kernel basis functions. From the boosting stand, this corresponds to incorporating the basis functions constructed by unlabeled data into the hypothesis space, which tends to produce better predictive models, especially when very few labels can be obtained. The 1-norm regularization method is employed to restrict the capacity of mixture models and to achieve sparsity. We modify LPBoost, a column generation (CG) boosting algorithm [4] to a more general linear programming formulation. The LPBoost modification can thus be used to efficiently optimize the proposed semi-supervised mixture-of-kernel models and further enhance the sparsity.

## 2 Semi-supervised Mixture of Kernels

We consider a problem that is slightly different from traditional transductive learning [7] where the task is to predict the labels of a working set of unlabeled examples that is given together with labeled examples. No inference model needs to be explicitly derived. In our problem, an inference model is required to predict labels of the test examples which are unseen in the training process. The validation of the algorithm performance is conducted only on an independent test set.

In a semi-supervised setting, besides a set of labeled data,  $L = \{(\mathbf{x}_i, y_i), i = 1, \dots, \ell\}$ , a set of unlabeled data  $U = \{\mathbf{x}_{\ell+j}, j = 1, \dots, \ell_u\}$  is also given. The kernel ma-

trix calculated on both labeled and unlabeled data is

$$\mathbf{K}^p = \begin{pmatrix} \mathbf{K}_{L,L}^p & \mathbf{K}_{L,U}^p \\ \mathbf{K}_{L,U}^{pT} & \mathbf{K}_{U,U}^p \end{pmatrix} \quad (1)$$

where  $\mathbf{K}_{i,j}^p = K_p(\mathbf{x}_i, \mathbf{x}_j)$ ,  $i, j = 1, \dots, \ell, \ell + 1, \dots, \ell + \ell_u$ . The classification model based on a mixture of these kernel matrices takes form of

$$f(\mathbf{x}) = \sum_p \sum_{j=1}^{\ell+\ell_u} \alpha_j^p K_p(\mathbf{x}, \mathbf{x}_j), \quad (2)$$

and the classifier is defined by  $\text{sgn}(f(\mathbf{x}))$ . Models of form (2) allow basis functions to locate at examples with no labels. The estimates of the centers or support vectors can still be automatically optimized by a SVM-like algorithm. Furthermore, a lot more choices for basis centers come for free from unlabelled examples without any need of unsupervised techniques.

Similar to SVMs, we optimize models by minimizing the margin-based 1-norm error metric  $\sum_{i=1}^{\ell} \xi_i$  where  $\xi_i$  satisfies  $y_i f(\mathbf{x}_i) \geq 1 - \xi_i, \forall i = 1, \dots, \ell$ . Without loss of generality, we include an explicit offset  $b$  in the model  $f(\mathbf{x})$ . Note that the offset can be incorporated into the kernel matrix. We write out  $b$  explicitly since we do not require the regularization to be taken on  $b$ . To achieve good generalization, it is important to apply appropriate regularization conditions to the model class. A commonly used regularization condition by single-kernel methods is the reproducing kernel Hilbert space (RKHS) regularization  $R(f)$ .

$$R(f) = \left\| \sum_i \alpha_i K(\mathbf{x}, \mathbf{x}_i) \right\|_{\mathcal{H}}^2 = \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha}. \quad (3)$$

where  $\mathcal{H}$  is the RKHS induced by the kernel  $K$ . The natural extension of the RKHS regularization condition to the semi-supervised mixture-of-kernel model is the following  $R(f)$

$$R(f) = \sum_p \boldsymbol{\alpha}^{pT} \mathbf{K}^p \boldsymbol{\alpha}^p. \quad (4)$$

The objective of the learning problem with RKHS regularization is to minimize  $\sum_p \boldsymbol{\alpha}^{pT} \mathbf{K}^p \boldsymbol{\alpha}^p + C \sum_{i=1}^{\ell} \xi_i$  where the error term is defined only using the top  $\ell$  rows of the kernel matrix  $\mathbf{K}$  as illustrated in (1), and the regularization term uses the entire kernel matrix  $\mathbf{K}$ .

The RKHS regularization condition requires positive semi-definiteness (PSD) of each of the kernel matrices  $K_p$ , and solving the resulting optimization problem can be computationally expensive. To remove the PSD requirement and achieve computational efficiency, we can apply other regularization conditions, such as penalizing the 1-norm or 2-norm of  $\boldsymbol{\alpha}$ , that are equally suitable for capacity control. In particular, we are in favor of the 1-norm regularization

$\|\boldsymbol{\alpha}\|_1 = \sum |\alpha_j|$  since it is well known that the 1-norm regularization leads to sparse solutions, which in the mixture-of-kernel model, is very desirable. Furthermore, we prove that the RKHS norm can be bounded using the 1-norm  $\sum_i |\alpha_i^p|$ .

$$\begin{aligned} \left\| \sum_i \alpha_i^p K_p(\mathbf{x}, \mathbf{x}_i) \right\|_{\mathcal{H}_p}^2 &= \sum_i \alpha_i^p \sum_j \alpha_j^p K_p(\mathbf{x}_i, \mathbf{x}_j) \\ &\leq (\sum_i |\alpha_i^p|) \sup_{i,j} \left| \sum_j \alpha_j^p K_p(\mathbf{x}_i, \mathbf{x}_j) \right| \\ &\leq \sum_i |\alpha_i^p| (\sup_i K_p(\mathbf{x}_i, \mathbf{x}_i))^{1/2} \left\| \sum_i \alpha_i^p K_p(\mathbf{x}_i, \mathbf{x}) \right\|_{\mathcal{H}_p}. \end{aligned}$$

Thus,

$$\left\| \sum_i \alpha_i^p K_p(\mathbf{x}, \mathbf{x}_i) \right\|_{\mathcal{H}_p} \leq \sum_i |\alpha_i^p| \left( \sup_i K_p(\mathbf{x}_i, \mathbf{x}_i) \right)^{1/2}.$$

Consequently, the 1-norm regularization is less restrictive than RKHS regularization, and hence gives a more expressive model.

For notational convenience, let us line up all kernel matrices together  $\mathbf{K} = [\mathbf{K}^1 \ \mathbf{K}^2 \ \dots \ \mathbf{K}^p]$ , and re-index the columns in  $\mathbf{K}$ . Let index  $j$  run through the columns and index  $i$  run along the rows. Hence  $\mathbf{K}_i$  denotes the  $i^{\text{th}}$  row of  $\mathbf{K}$ , and  $\mathbf{K}_{.j}$  denotes the  $j^{\text{th}}$  column. There are  $d = (\ell + \ell_u) \times p$  columns in total. We thus formulate the learning problem as

$$\begin{aligned} \min_{\boldsymbol{\alpha}, \boldsymbol{\xi}} \quad & \sum_{j=1}^d |\alpha_j| + C \sum_{i=1}^{\ell} \xi_i \\ \text{s.t.} \quad & y_i \left( \sum_j \mathbf{K}_{ij} \alpha_j + b \right) + \xi_i \geq 1, \\ & \xi_i \geq 0, \quad i = 1, \dots, \ell, \end{aligned} \quad (5)$$

where the  $j^{\text{th}}$  column now only consists of the first  $\ell$  elements of that in  $\mathbf{K}$ .

### 3 Generalized LPBoost

The CG techniques have been widely used for solving large-scale linear programs (LPs) or difficult integer programs since 1950s [6]. In the primal space, the CG method solves LPs on a subset of variables  $\boldsymbol{\alpha}$ , which means not all columns of the kernel matrix are generated at once and used to construct the function  $f$ . Columns are generated iteratively and added to the problem to achieve optimality. In the dual space, a column in the primal problem corresponds to a constraint in the dual problem. When a column is not included in the primal, the corresponding constraint does not appear in the dual. If a constraint absent from the dual problem is violated by the solution to the restricted problem, this constraint (a cutting plane) needs to be included in the dual problem to further restrict its feasible region. Thus these techniques are also referred to as cutting plane methods [1]. We first briefly review the existing LPBoost with

1-norm regularization. Then we propose our modification that allows us to consider kernels that do not necessarily comply with the PSD requirement.

If the hypothesis  $\mathbf{K}_{\cdot j}\alpha_j$  based on a single column of the matrix  $\mathbf{K}$  is regarded as a weak model or base classifier, we can rewrite LPBoost using our notation and following the statement in [4]:

$$\begin{aligned} \min_{\alpha, \xi} \quad & \sum_{j=1}^d \alpha_j + C \sum_{i=1}^{\ell} \xi_i \\ \text{s.t.} \quad & y_i \sum_j \mathbf{K}_{ij} \alpha_j + \xi_i \geq 1, \quad \xi_i \geq 0, \quad i = 1, \dots, \ell \\ & \alpha_j \geq 0, \quad j = 1, \dots, d, \end{aligned} \quad (6)$$

where  $C > 0$  is the regularization factor. The dual of (6) is

$$\begin{aligned} \max_{\beta} \quad & \sum_{i=1}^{\ell} \beta_i \\ \text{s.t.} \quad & \sum_{i=1}^{\ell} \beta_i y_i \mathbf{K}_{ij} \leq 1, \quad j = 1, \dots, d, \\ & 0 \leq \beta_i \leq C, \quad i = 1, \dots, \ell. \end{aligned} \quad (7)$$

These problems are referred to as the master problems. The CG method solves LPs by incrementally selecting a subset of columns from the simplex tableau and optimizing the tableau restricted on the subset of variables (each corresponding to a selected column). After a primal-dual solution  $(\hat{\alpha}, \hat{\xi}, \hat{\beta})$  to the restricted problem is obtained, we solve

$$\tau = \max_j \sum_i \hat{\beta}_i y_i \mathbf{K}_{ij}, \quad (8)$$

where  $j$  runs over all columns of  $\mathbf{K}$ . If  $\tau \leq 1$ , the solution for the restricted problem is optimal to the master problems. If  $\tau > 1$ , then the solution to (8) provides a column to be included in the restricted problem.

As illustrated in problem (5), kernel methods in general do not require the model coefficients  $\alpha_i$  to be non-negative. Moreover, an explicit offset term  $b$  can be included in the function  $f$ . To form a LP from problem (5), we rewrite  $\alpha_j = u_j - v_j$  where  $u_j, v_j \geq 0$ . Then  $|\alpha_j| = u_j + v_j$  if either  $u_j$  or  $v_j$  has to be 0. The LP is then formulated in variables  $\mathbf{u}, \mathbf{v}, b, \xi$  as

$$\begin{aligned} \min_{\mathbf{u}, \mathbf{v}, \xi} \quad & \sum_{j=1}^d (u_j + v_j) + C \sum_{i=1}^{\ell} \xi_i \\ \text{s.t.} \quad & y_i \left( \sum_j \mathbf{K}_{ij} (u_j - v_j) + b \right) + \xi_i \geq 1, \quad (9) \\ & \xi_i \geq 0, \quad i = 1, \dots, \ell, \\ & u_j, v_j \geq 0, \quad j = 1, \dots, d. \end{aligned}$$

Solving the above LP yields solutions equivalent to those obtained by (5) because in the optimal solution, at least one of the two variables  $u_j$  and  $v_j$  will be zero for all  $j = 1, \dots, d$ . Otherwise, assume  $u_j > v_j > 0$  without loss

of generality, and we can find a better solution by setting another feasible solution  $\hat{u}_j = u_j - v_j$  and  $\hat{v}_j = 0$ . Then  $\hat{u}_j + \hat{v}_j = u_j - v_j < u_j + v_j$  contradicting the optimality of  $(\mathbf{u}, \mathbf{v})$ .

Two variables  $u_j, v_j$  correspond to a column  $\mathbf{K}_{\cdot j}$  of the kernel matrix in problem (9). Correspondingly the Lagrangian dual problem has two constraints for the column  $\mathbf{K}_{\cdot j}$ , i.e.,  $\sum_{i=1}^{\ell} \beta_i y_i \mathbf{K}_{ij} \leq 1$  and  $-\sum_{i=1}^{\ell} \beta_i y_i \mathbf{K}_{ij} \leq 1$ . Combining both constraints, we have  $-1 \leq \sum_{i=1}^{\ell} \beta_i y_i \mathbf{K}_{ij} \leq 1$ . Hence the dual problem becomes:

$$\begin{aligned} \max_{\beta} \quad & \sum_{i=1}^{\ell} \beta_i \\ \text{s.t.} \quad & -1 \leq \sum_{i=1}^{\ell} \beta_i y_i \mathbf{K}_{ij} \leq 1, \quad j = 1, \dots, d, \\ & \sum_{i=1}^{\ell} \beta_i y_i = 0, \\ & 0 \leq \beta_i \leq C, \quad i = 1, \dots, \ell. \end{aligned} \quad (10)$$

The CG method partitions the variables  $\alpha_j$  into two sets, the working set  $W$  used to build the model and the remaining set denoted as  $N$  that is eliminated from the model as the corresponding columns are not generated. Each CG step optimizes a subproblem over the working set  $W$  of variables and then selects a column from  $N$  to add to  $W$ . At each iteration,  $\alpha_j$  in  $N$  can be interpreted as  $\alpha_j = 0$ , or accordingly,  $u_j, v_j = 0$ . Hence once a solution  $\alpha^W = \mathbf{u}^W - \mathbf{v}^W$  to the restricted problem is obtained,  $\hat{\alpha} = (\alpha^W \alpha^N = 0)$  is feasible to the master LP (9). The following statement examines when an optimal solution for the master problem is obtained in the CG procedure.

**Proposition 1 (Optimality of LP CG)** *Let  $(\hat{\mathbf{u}}, \hat{\mathbf{v}}, \hat{\xi}, \hat{\beta})$  be the primal-dual solution to the current restricted problems with variable  $b$  included in  $W$ . The solution is optimal to LP (9) if and only if for all  $j \in N$ ,  $\left| \sum_i \hat{\beta}_i y_i \mathbf{K}_{ij} \right| \leq 1$ .*

To show the optimality is achieved, we need to confirm primal feasibility, dual feasibility and the equality of primal and dual objectives. Recall how we define  $\hat{\mathbf{u}} = (\mathbf{u}^W \mathbf{u}^N = 0)$  and  $\hat{\mathbf{v}} = (\mathbf{v}^W \mathbf{v}^N = 0)$ , so  $(\hat{\mathbf{u}}, \hat{\mathbf{v}}, \hat{\xi})$  is feasible for LP (9). Since the solution is optimal to the restricted problems, the primal objective is equal to the dual objective. Now the key issue to evaluate is the dual feasibility. Since  $\hat{\beta}$  is optimal for the restricted problem, it satisfies all constraints of the restricted dual. Hence the dual feasibility is validated if  $\left| \sum_i \hat{\beta}_i y_i \mathbf{K}_{ij} \right| \leq 1, j \in N$ .

Any column that violates dual feasibility can be added. For LPs, a common heuristic is to choose the column  $\mathbf{K}_{\cdot j}$  that maximizes  $\left| \sum_i \hat{\beta}_i y_i \mathbf{K}_{ij} \right|$  over all  $j \in N$ . In other words, the column  $\mathbf{K}_{\cdot j}$  that solves

$$\tau = \max_{j \in N} \left| \sum_i \hat{\beta}_i y_i \mathbf{K}_{ij} \right| \quad (11)$$

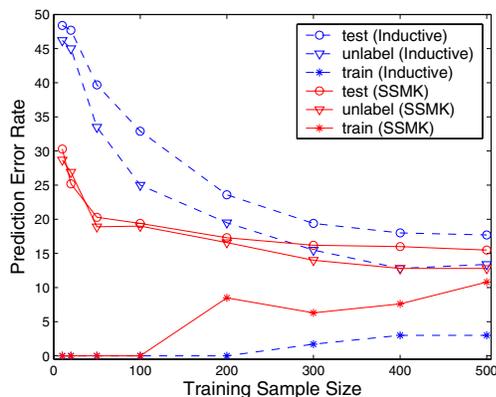
will be included in the restricted problem. Compared with original LPBoost, our method is in general equivalent to enclosing negations of weak models  $\mathbf{K}_{.j}$  in the hypothesis set. We describe the modified LPBoost for semi-supervised learning with a mixture of kernels (SSMK) in Algorithm 1 where  $\mathbf{e}$  is a vector of ones of appropriate dimension corresponding to the bias term  $b$ .

**Algorithm 1 SSMK: CG Boosting for LP (9)**

1. Initialize the first column  $\mathbf{K}_0 = \mathbf{e}$ , specify the tolerance  $tol$
2. For  $t = 1$  to  $T$ , do
3. Solve problem (9) with  $\mathbf{K}_{t-1}$ , obtain solution  $(\mathbf{u}^t, \mathbf{v}^t, \xi^t, \beta^t)$
4. Solve problem (11) to obtain  $\tau$ , and let  $\mathbf{z}$  be the solution
5. If  $\tau \leq 1 + tol$ , optimal, break from loop, otherwise,  $\mathbf{K}_t = [\mathbf{K}_{t-1} \ \mathbf{z}]$ , continue
6. End of loop
7.  $\hat{\alpha} = \mathbf{u}^t - \mathbf{v}^t$ .

**4 Experimental Study**

We validate the proposed approach on NIST handwritten digit database and compare it to the inductive mixture-of-kernel (IMK) approach. More thorough comparison between IMK and other kernel methods can be found in [3]. We generated datasets in the following way: preserve  $\ell_t = 2000$  examples randomly drawn from the databases as test sets (T); randomly take  $\ell_u = 500$  examples from the remaining data as unlabeled data (U) used in model construction; then randomly take  $\ell = 10, 20, 50, 100, 200, 300, 400, 500$  examples as training data (L) with labels. We performed 10 trials, and present error rates averaged on these trials in Figure 1. The linear kernel and RBF kernels were considered.



**Figure 1. Plot of error versus training size  $\ell$ .**

Clearly, from Figure 1, the use of unlabeled data in LP(9)

helps improve the prediction performance both on the unlabeled data which is used in training with no labels and the test data which is completely blinded to the training process. With more and more labeled data available, the difference between IMK and SSMK is generally reduced. By applying the generalized LPBoost, we dramatically reduced the size of optimization problems to be solved and thus gained computational efficiency. For example, when  $\ell = 100$  at training, we solved a problem of very large size  $100 \times 1200$  in IMK, and 47 linear kernel basis and 2 RBF basis were used in the final model. In SSMK with the CG procedure, we need to solve problems of much smaller size (the largest size =  $100 \times 34$  during the 34 CG iterations). The resulting model used only 26 linear kernel columns. Hence SSMK had potentials to find more sparse solutions although both IMK and SSMK obtained sparse solutions due to the use of 1-norm regularization.

**5 Conclusions**

We have explored the mixture-of-kernel model in semi-supervised learning settings. By using kernels with different geometric properties and allowing the basis functions to locate at unlabeled data points, we enhance the capability of kernel methods to adapt to tasks with various target functions. The optimization problems involved in model construction can be solved in a very efficient way through the exploitation of column generation techniques.

**References**

- [1] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty. *Nonlinear Programming: Theory and Algorithms*. John Wiley & Sons, Inc., New York, NY, 1993.
- [2] K. Bennett, M. Momma, and M. Embrechts. MARK: A boosting algorithm for heterogeneous kernel models. In *Proceedings of SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 24–31, 2002.
- [3] J. Bi, T. Zhang, and K. P. Bennett. Column-generation boosting methods for mixture of kernels. In *Proceedings of SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 521–526, 2004.
- [4] A. Demiriz, K. P. Bennett, and J. Shawe-Taylor. Linear programming boosting via column generation. *Machine Learning*, 46(1–3):225–254, 2002.
- [5] G. Lanckriet, N. Cristianini, P. Bartlett, L. Ghaoui, and M. I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2003.
- [6] S. G. Nash and A. Sofer. *Linear and Nonlinear Programming*. McGraw-Hill, New York, NY, 1996.
- [7] V. N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, Inc., New York, 1998.