

Localized Multiple Kernel Learning

M. Tech. Thesis

Submitted in partial fulfillment of the requirements
for the degree of

Master of Technology

by

Kolli Sarath

Roll No: 10305062

under the guidance of

Prof. J.Saketha Nath



Department of Computer Science and Engineering
Indian Institute of Technology, Bombay
Mumbai

Dissertation Approval Certificate

Department of Computer Science and Engineering

Indian Institute of Technology Bombay

The dissertation entitled "**Localized Multiple Kernel Learning**", submitted by **Kolli Sarath** (Roll No: **10305062**) is approved for the degree of **Master of Technology in Computer Science and Engineering** from **Indian Institute of Technology Bombay**.



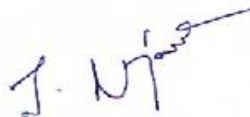
Prof. J. Saketha Nath
Dept CSE, IIT Bombay
Supervisor



Prof. Sunita Sarawagi
Dept CSE, IIT Bombay
Examiner-I



Prof. Pushpak Bhattacharya
Dept CSE, IIT Bombay
Examiner-II



Prof. J. Adinarayana
Dept CSRE, IIT Bombay
Chairperson

Place: IIT Bombay, Mumbai

Date: 13th July, 2012

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.



Kolli Sarath

Roll No. : 10305062

Date: 13th July, 2012

Acknowledgments

I thank and express my utmost gratitude to **Prof. J. Saketha Nath**, Department of Computer Science and Engineering, IIT Bombay, who has guided me and helped me throughout this project. Without his deep insight into this domain and his valuable time for this project, it would not have been possible for me to move ahead properly. He rectified my basic mistakes and explained me the things in as easy way as possible. Without him and his efforts, my understanding would have been incomplete towards the topic. He has been a constant source of inspiration for me throughout for the achievement of this task. He has been remarkable in his attempt to keep me motivated in this project and has always tried to improve me with proper feedback.

Kolli Sarath

Roll No. : 10305062

Abstract

Kernel functions are widely used in several algorithms in machine learning and statistics. In recent years instead of using a single kernel people are using combination multiple kernels. These different kernels may use information acquired from different sources or different similarity measures. Several Multiple Kernel Learning (MKL) methods are present for combining the Kernels with same weights over all the points. Localized MKL is where the weights of the Kernels will change for every point.

In the project we studied different MKL methods and Localized MKL method and did an unbiased comparison by performing experiments on different real world datasets. We also modified the current localized MKL approach in different ways to achieve better results. We changed the gating function into linear function used in the Localized MKL to linear function that ended up giving mixed results. We used Localized mkl with local SVM instead of global classifier which also ended up giving mixed results.

Contents

1	Introduction	1
2	Literature of Multiple Kernel Learning	3
3	Modifications on Localized MKL	6
3.1	Gating functions	6
3.1.1	η_m as a linear function of x	6
3.1.2	Using a gating Kernel	7
3.2	Localized MKL with local classifier	8
3.2.1	Locally linear SVM	8
3.2.2	Localized MKL with local SVM	9
4	Experiments	11
4.1	Datasets	11
4.1.1	Pendigits	11
4.1.2	Protein Fold Prediction	12
4.1.3	CAL500	12
4.2	Results	13
5	Conclusion and Future Work	14

Chapter 1

Introduction

Support Vector Machine (SVM) methods [CORTES and VAPNIK, 1995] become widely used in many classification tasks due to their success. The main advantage using SVM's is we can get linear separation by mapping the instances from input space to new feature space. But finding the map to every instance in the new feature space is costly. Given N iid instances $\{(x_i, y_i)\}_{i=1}^N$ where x_i is d -dimensional input vector and y_i is its label. SVM finds the linear discriminant with maximum margin in the the new feature space given by the mapping function $\phi : R^d \rightarrow R^t$.

$$f(x) = \langle w, x \rangle + b$$

The classifier can be trained by solving the following optimization problem is:

$$\begin{aligned} \min_{w, b, \xi_i} \quad & \frac{1}{2} \|w\|^2 + C \sum_i \xi_i \\ \text{s.t.} \quad & y_i(\langle w, x_i \rangle + b) \geq 1 - \xi_i, \xi_i \geq 0 \forall i \end{aligned}$$

where C is the regularization parameter and ξ_i 's are the slack variables. The Dual problem for the above optimization problem will be

$$\begin{aligned} \max_{\alpha_i} \quad & \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \\ \text{s.t.} \quad & \sum_{i=1}^N \alpha_i y_i = 0, \quad C \geq \alpha_i \geq 0 \forall i \end{aligned} \tag{1.1}$$

where α_i 's are the dual variables. From Equation (1.1) notice that the data is appeared only in inner products products $\langle x_i, x_j \rangle$. At this point Kernels gives as good advantage so that we don't have to find the map to any instance in the new feature space. If there is a function such that $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$ then no need for calculating $\Phi(x)$. The hypothesis function will be

$$f(x) = \sum_{i=1}^N \alpha_i y_i \langle \phi(x_i), \phi(x) \rangle + b.$$

Selecting a kernel function for the training is an important part in the training. Generally, a cross-validation procedure is used to choose the best performing kernel function among a set of kernel functions. In recent years many multiple kernel learning methods are proposed so that we can combine them to get a better kernel. Simplest way to do this is taking the unweighted sum of all kernels. This gives equal priority to every kernel but this method may not be an ideal because some kernels may not be as good as other kernels. One better method is taking the weighted sum of the given kernels. [Bach et al., 2004, Aflalo et al., 2011, Lanckriet et al., 004a] takes the convex combination of kernels. And some methods [Corinna Cortes and Rostamizadeh, 2009] took the non-linear combination of kernels. These methods gives fixed weights to the kernels over whole input space. Using different weights for different points may produce a better classifier overall. With this idea Localized MKL[Gonen and Alpaydn, 2004] method is proposed which combines the Kernels linearly by giving different weights at each data point. In the first stage of project we compared different MKL algorithms with Localized MKL where it is giving better results in most of the cases so in the second stage tried different variations with localized MKL to achieve better performance.

In chapter 2 we explained some Multiple Kernel Learning methods highlighting the similarities and differences between them. In chapter 3 we discussed some improvements for Localized MKL method. Experimental results are discussed in chapter 4 and conclusion are given in chapter 5.

Chapter 2

Literature of Multiple Kernel Learning

In the recent years lot of research was done on Multiple Kernel Learning problem. Linear combination of kernels as weighted sum is the most popular approach taken by many researchers. In this various approaches are proposed for sparse and non-sparse combination of Kernels.

$$K_{\eta}(x_i, x_j) = \sum_{m=1}^p \eta_m K_m(x_i, x_j)$$

where p is the no. of Kernels and K_m represents the m^{th} Kernel. Different versions of this approach put different restrictions on η 's:

- Linear Sum (i.e. $\eta \in R^p$)
- Conic Sum (i.e. $\eta \in R_+^p$)
- Convex Sum (i.e. $\eta \in R_+^p$ and $\sum_{m=1}^p \eta_m = 1$)

The conic and convex sums have different advantages than linear sum in terms of interpretability. First, when we have positive kernel weights, we can extract the relative importance of the combined kernels by looking at the weights of kernels. Second, when we restrict the kernel weights to be nonnegative, this corresponds to scaling the feature spaces and using the concatenation of them as the combined feature representation.

The discriminant function for the above combination looks like:

$$f(x) = \sum_{m=1}^p \sqrt{\eta_m} \langle w_m, \phi_m(x) \rangle + b.$$

And the optimization problem for MKL looks like:

$$\begin{aligned} \min_{\eta_m, w, b, \xi_i} \quad & \Omega(w) + C \sum_{i=1}^N \xi_i \\ \text{s.t.} \quad & y_i \left(\sum_{m=1}^p \sqrt{\eta_m} \langle w_m, \phi_m(x_i) \rangle + b \right) \geq 1 - \xi_i, \xi_i \geq 0 \forall i, \sum_{m=1}^p \eta_m^l \leq 1 \end{aligned} \quad (2.1)$$

$\Omega(w)$ represents some norm function on w , some people also tried mixed norms [Aflalo et al., 2011] and different norms are used to restrict η value. After finding the weights and solving the problem the discriminant function is:

$$f(x) = \sum_{i=1}^N \sum_{m=1}^p \eta_m \alpha_i y_i K_m(x_i, x) + b.$$

[Lanckriet et al., 004a] formulated this as a semidefinite programming problem which finds the combination weights and support vector coefficients together. [Bach et al., 2004, Vishwanathan et al., 2010] used SMO approach to solve the problem by using little variations of the formulation. [Bach et al., 2008] used gradient descent method to find the weights and [Aflalo et al., 2011] used the mirror descent approach to solve the problem.

All these methods used same weights kernels all over the input space. By using different weights in different localities we may get better classifier. Using this [Gonen, 2004] proposed localized multiple kernel method i.e., the weights of kernels depends on the data points.

$$K_\eta(x_i, x_j) = \sum_{m=1}^p \eta_m(x_i) K_m(x_i, x_j) \eta_m(x_j) \quad (2.2)$$

where $\eta_m(x)$ is the gating function that takes input as x . The discriminant function look like:

$$f(x) = \sum_{m=1}^p \eta_m(x) \langle w_m, \phi_m(x) \rangle + b.$$

The optimization problem will be:

$$\min_{\eta_m(x), w, b, \xi_i} \frac{1}{2} \sum_{m=1}^p \|w_m\|^2 + C \sum_{i=1}^N \xi_i \quad (2.3)$$

$$s.t. \quad y_i \left(\sum_{m=1}^p \eta_m(x_i) \langle w_m, \phi_m(x_i) \rangle + b \right) \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad \forall i$$

they used a softmax gating function for $\eta_m(x)$

$$\eta_m(x) = \frac{\exp(\langle v_m, x \rangle + v_{m0})}{\sum_{i=1}^p \exp(\langle v_i, x \rangle + v_{i0})} \quad (2.4)$$

where v_m and v_{m0} are the parameters of gating function.

They used the alternate optimization, first solving the problem with constant $\eta_m(x)$ and then gradient descent method to find the parameters of $\eta_m(x)$ i.e. v_m, v_{m0} . The final discriminant function after solving for $\eta_m(x)$ is:

$$f(x) = \sum_{i=1}^N \sum_{m=1}^p \eta_m(x_i) \alpha_i y_i K_m(x_i, x) \eta_m(x) + b.$$

Chapter 3

Modifications on Localized MKL

We tried some modifications to the localized MKL to achieve better classifier. First we tried to use different gating functions, after that we used localized MKL with local classifiers.

3.1 Gating functions

3.1.1 η_m as a linear function of x .

[Gonen and Alpaydn, 2004] used softmax gating function so that $\eta_m(x)$ is non-negative and to get a positive semidefinite kernel matrix. From 2.2 the resulting kernel matrix will be:

$$K_\eta = \sum_{m=1}^p (\eta_m \eta_m^T) \cdot * K_m \quad (3.1)$$

where η_m is a $N \times 1$ vector that gives values of all $\eta_m(x_i) \forall i \in [1, N]$. With this we can say that we don't have to select non-negative $\eta_m(x)$ we have to select the function as η_m is a Kernel matrix. So we represented $\eta(x)$ as a linear function instead of using a gating function:

$$\eta_m(x) = v_m^T x \quad (3.2)$$

By taking this linear function also the resulting kernel matrix will also be a positive semidefinite matrix. Previously because of using the gating function we will get a sparse combination of kernels. But here the weights will be non-sparse. Here we used a 2-norm regularizer on the parameter v_m to get some sparsity.

Then the primal optimization problem will be

$$\begin{aligned} \min_{\eta_m(x), w, b, \xi_i} & \frac{1}{\sum_{m=1}^p} p \|w_m\|^2 + C \sum_{i=1}^N \xi_i + C_1 \|v_m\|^2 \\ \text{s.t.} & y_i \left(\sum_{m=1}^p \eta_m(x_i) \langle w_m, \phi_m(x_i) \rangle + b \right) \geq 1 - \xi_i, \xi_i \geq 0 \forall i \end{aligned} \quad (3.3)$$

The dual for 3.3 by using the η_m as in 3.2 and at constant η_m is

$$\max_{\alpha_i} \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K_\eta(x_i, x_j) + C_1 \|v_m\|^2 \quad (3.4)$$

We will find η_m by gradient descent method same as used in [Gonen and Alpaydn, 2004]. Using 3.4 objective value $J(\eta)$ we can calculate the derivatives with respect to the parameters of $\eta_m(x)$.

$$\frac{\partial J(\eta)}{\partial v_m} = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K_m(x_i, x_j) (x_i \eta_m(x_j) + x_j \eta_m(x_i)) + 2C_1 v_m.$$

After updating $\eta_m(x)$ we will solve single kernel SVM with $K_\eta(x_i, x_j)$ at each step. The experimental results of this are discussed in next chapter.

3.1.2 Using a gating Kernel

Instead of taking gating function on each data point as in 2.2, we can use a gating function η_m such that

$$K_\eta(x_i, x_j) = \sum_{m=1}^p \eta_m(x_i, x_j) K_m(x_i, x_j)$$

Because using gating function on pairs of data points makes more naive and effective than using gating function on each data point. If the gating function is a kernel function we won't have any problem. Because the resulting kernel matrix will be a positive semi definite. So we can take any generally known kernel functions as η_m . But here we used a sigmoid kernel as η_m because it works as a good gating function. Since it is not always gives positive semi definite Kernel matrix we added ridge to make it positive semi definite. We used the same alternate minimization approach as in [Gonen and Alpaydn, 2004]. We didn't perform any experiments on this because it is converging in just one step without any change in the objective value.

3.2 Localized MKL with local classifier

Up to now MKL is applied on the global classifiers. [Ladicky and Torr, 2011] proposed local SVM that gives different classifier at different regions of the input space.

3.2.1 Locally linear SVM

The standard SVM linear discriminant is:

$$f(x) = w^T x + b$$

To encode local linearity of the SVM classifier by allowing the weight vector w and bias b to vary depending in the location of the point x in the feature space as:

$$f(x) = w(x)^T x + b(x) \tag{3.5}$$

Here they local codings methods that approximate any data point x as a linear combination of surrounding anchor points.

$$x \approx \sum_{v \in A} \gamma_v(x) v$$

where A is the set of anchor points and $\gamma_v(x)$ is the vector of coefficients called local coordinates constrained by $\sum_{v \in A} \gamma_v(x) = 1$. The anchor points can evaluated by different approaches. Using the manifold learning property any Lipschitz function $f(x)$ can be approximated as:

$$f(x) \approx \sum_{v \in A} \gamma_v(x) f(v). \tag{3.6}$$

Using 3.6, 3.5 we can be rewritten as

$$\begin{aligned} f(x) &= \sum_{i=1}^N w_i(x) x_i + b(x) \\ &= \sum_{i=1}^N \sum_{v \in A} \gamma_v(x) w_i(v) x_i + \sum_{v \in A} \gamma_v(x) b(v). \\ &= \sum_{v \in A} \gamma_v(x) \left(\sum_{i=1}^N w_i(v) x_i + b(v) \right). \end{aligned} \tag{3.7}$$

Here learning the classifier involves finding optimal $w_i(v)$ and $b(v)$ for each anchor point v . Let the number of anchor points be $m = |A|$. Let W be the $m \times n$ matrix where each row is equal to $w_i(v)$

of corresponding anchor point v and b be the vector of $b(v)$ for each anchor point. Then 3.7 can be written as:

$$f(x) = \gamma(x)^T W x + \gamma(x)^T b. \quad (3.8)$$

The optimization problem will be:

$$\min_{W, b, \xi_i} \frac{1}{2} \|W\|^2 + C \sum_{i=1}^N \xi_i \quad (3.9)$$

$$\forall i \quad y_i(\gamma(x_i)^T W x_i + \gamma(x_i)^T b) \geq 1 - \xi_i, \xi_i \geq 0$$

where $\|W\|^2 = \sum_{i=1}^m \sum_{j=1}^n W_{ij}^2$. We can solve this QP problem 3.9 in different ways by using SMO like algorithms on the dual representation. But [Ladicky and Torr, 2011] used stochastic gradient method to solve the above optimization problem.

3.2.2 Localized MKL with local SVM

By using the Localized MKL idea in LLSVM the optimization problem 3.9 will become:

$$\min_{W, b, \eta \geq 0, \xi_i} \frac{1}{2} \sum_{m=1}^p \|W_m\|^2 + C \sum_{i=1}^N \xi_i \quad (3.10)$$

$$\forall i \quad y_i \left(\sum_{m=1}^p \eta_m(x_i) \gamma(x_i)^T W_m x_i + \gamma(x_i)^T b \right) \geq 1 - \xi_i, \xi_i \geq 0$$

We will also use the same method for solving the optimization problem as in [Gonen and Alpaydn, 2004]

by solving 3.10 with respect to w_m , b and ξ_i and then updating $\eta_m(x)$ by gradient descent method.

For a fixed $\eta_m(x)$ the Lagrangian of 3.10 will be

$$L_D = \frac{1}{2} \sum_{m=1}^p \|W_m\|^2 + C \sum_{i=1}^N \xi_i + \sum_{i=1}^N \alpha_i (1 - \xi_i - y_i (\sum_{m=1}^p \eta_m(x_i) \gamma(x_i)^T W_m x_i + \gamma(x_i)^T b)) + \sum_{i=1}^N \beta_i (-\xi_i)$$

$$\frac{\partial L_D}{\partial W_m} = 0 \implies W_m = \sum_{i=1}^N \alpha_i y_i \eta_m(x_i) \gamma(x_i) x_i^T$$

$$\frac{\partial L_D}{\partial b} = 0 \implies \sum_{i=1}^N \alpha_i y_i \gamma(x_i) = 0$$

$$\frac{\partial L_D}{\partial \xi_i} = 0 \implies C = \alpha_i + \beta_i \quad (3.11)$$

From 3.10 and 3.11 the dual formulation will be

$$\begin{aligned} \max_{\alpha \geq 0} \quad & \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K_\eta^\gamma(x_i, x_j) \\ \text{s.t.} \quad & \sum_{i=1}^N \alpha_i y_i \gamma(x_i)^T = 0, \quad C \geq \alpha_i \geq 0 \forall i \end{aligned} \quad (3.12)$$

where K_η^γ is

$$K_\eta^\gamma(x_i, x_j) = \sum_{m=1}^p \gamma(x_i)^T \gamma(x_j) \eta_m(x_i) \eta_m(x_j) K_m(x_i, x_j)$$

where $\eta_m(x)$ is same as in 2.4. The gradient of $J(\eta)$ objective value obtained from 3.12 with respect to the parameters of $\eta_m(x)$. The derivatives of $J(\eta)$ with respect to v_m, v_{m0} are

$$\frac{\partial J(\eta)}{\partial v_{m0}} = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^p \alpha_i \alpha_j y_i y_j \gamma(x_i)^T \gamma(x_j) \eta_k(x_i) K_k(x_i, x_j) \eta_k(x_j) (2\delta_m^k - \eta_m(x_i) - \eta_m(x_j))$$

$$\frac{\partial J(\eta)}{\partial v_m} = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^p \alpha_i \alpha_j y_i y_j \gamma(x_i)^T \gamma(x_j) \eta_k(x_i) K_k(x_i, x_j) \eta_k(x_j) (x_i (\delta_m^k - \eta_m(x_i)) + x_j (\delta_m^k - \eta_m(x_j)))$$

where $\delta_m^k = 1$ if $m = k$ otherwise 0. After updating $\eta_m(x)$ we will solve single kernel SVM with $K_\eta(x_i, x_j)$ at each step. After determining the final $\eta_m(x)$ and SVM solution the discriminant function will be:

$$f(x) = \gamma(x)^T \left(\sum_{i=1}^n \sum_{m=1}^p \alpha_i y_i \gamma(x_i) \eta_m(x_i) K_m(x_i, x_j) \eta_m(x_j) + b \right).$$

Chapter 4

Experiments

We did experiments on the real world datasets taken from [UCS,]. We compared some MKL methods like [Aflalo et al., 2011] and [Vishwanathan et al., 2010] with Localized MKL [Gonen and Alpaydm, 2004] Then we compared our modified formulations 3.1.1 and 3.2.2 with these methods.

4.1 Datasets

4.1.1 Pendigits

Name	Dimension	Data Source
dyn	16	eight successive pen points on two-dimensional coordinate system
sta16	256	16 x 16 image bitmap representation formed by connecting the points in dyn representation with line segments
sta8	64	8 x 8 subsampled bitmap representation
sta4	16	4 x 4 subsampled bitmap representation

Table 4.1: Pendigits

The pendigits dataset 4.1 is on pen-based digit recognition (multiclass classification with 10 classes) and contains four different feature representations. The data set is split into independent training and test sets with 7494 samples for training and 3498 samples for testing.

4.1.2 Protein Fold Prediction

Name	Dimension	Data Source
Composition	D=20	amino acid Composition - Global protein characteristic
Secondary	D=21	predicted secondary structure - Global protein characteristic
Hydrophobicity	D=21	Global protein characteristic
Volume	D=21	Van der Waals volume - Global protein characteristic
Polarity	D=21	Global protein characteristic
Polarizability	D=21	Global protein characteristic
L1	D=22	PseAA pseudo-amino-acid composition at interval 1
L4	D=28	PseAA pseudo-amino-acid composition at interval 4
L14	D=48	PseAA pseudo-amino-acid composition at interval 14
L30	D=80	PseAA pseudo-amino-acid composition at interval 30

Table 4.2: Protein Fold Prediction

This dataset is on protein fold prediction 4.2(multi-class classification with 27 classes) based on a subset of the PDB-40D SCOP collection. It is an extension of the original dataset by Ding that also includes the pseudo-amino acid compositions proposed by Shen and Chou and the Smith-Waterman String kernels employed in Damoulas and Girolami. The data is split to independent train and test sets with 311 samples for training and 383 samples for testing.

4.1.3 CAL500

The CAL500 dataset 4.3 is a collection of 500 songs tagged with 174 tags by paid human labellers. 500 songs annotated using a vocab of 174 tags from 8 semantic categories that describe the genre (multiple and best), emotion, instruments, solos, vocal style, song characteristics and usage. Both binary (relevant / irrelevant) and affinity labels are included. This CAL500 multi-kernel dataset contains 6 kernels derived from various features that describe the music.

Name	Data Source
K_subsamplePPK	Probability Product Kernel (PPK) between Gaussian mixture models (GMM) of a sub-sampling of each songs Delta-MFCC feature vectors.
K_30sec_PPK	PPK between GMMs of 30 continuous seconds, starting 30 seconds into the song, of each song's Delta-MFCC feature vectors.
K_30sec_CHROMAPPK	PPK between GMMs of 30 continuous seconds, starting 30 seconds into the song, of each song's Chroma (pitch-histogram) feature vectors.
K_fpRBF	Radial basis function (RBF) kernel between the Fluctuation Pattern features.
K_lastfm	RBF kernel between document vectors derived from Last.fm's social tags
K_webdoc	RBF kernel between document vectors describing web pages returned by searching for each song on Google.

Table 4.3: CAL500

4.2 Results

For all the methods we did 5 fold Cross validation on datasets to choose the C value. For every dataset the manifold is trained with relative number of anchor points according to the size of dataset. Coefficients of the local coding were obtained using inverse Euclidean distance based weighting solved for 5-8 nearest neighbors according to the number of anchor points.

Dataset	p -norm MKL	VSKL	LMKL	LMKL-l	LMKL-LSVM
CAL500	$84.8 \pm 1.54(p = 10000)$	$86.16 \pm 1.65(q = \text{inf})$	84.83 ± 1.56	84.80 ± 1.55	86.08 ± 1.79
Protien Fold Prediction	$96.29 \pm 0.56(p = 10000)$	$94.81 \pm 1.34(q = \text{inf})$	96.94 ± 0.73	97.15 ± 0.69	96.79 ± 0.81
Pendigits	$90.00 \pm 0.44(p = 10000)$	$97.40 \pm 0.22(q = \text{inf})$	99.78 ± 0.07	98.77 ± 0.11	99.90 ± 0.05

Table 4.4: Results

Chapter 5

Conclusion and Future Work

In the project we tried to understand the MKL problem and to come up with a good solution. We modified the existing approach slightly and we also applied the existing approach on also local classifier instead of global classifier. In the end we came up with some methods that are giving comparable results as the previous methods.

We used the local SVM concept with MKL problem that gives one future approach that may be useful for further research in the MKL problem. And also using some good gating kernels in localized MKL may give some good results.

Bibliography

- [UCS,] The ucsd multiple kernel learning repository <http://mkl.ucsd.edu/>.
- [Aflalo et al., 2011] Aflalo, J., Ben-Tal, A., Bhattacharyya, C., Nath, J. S., and Raman, S. (2011). Variable sparsity kernel learning. *Journal of Machine Learning Research*.
- [Bach et al., 2004] Bach, F. R., Lanckriet, G. R. G., and Jordan, M. I. (2004). Multiple kernel learning, conic duality, and the smo algorithm. In *21st International Conference on Machine Learning*.
- [Bach et al., 2008] Bach, F. R., phane Canu, S., Rakotomamonjy, A., and Grandvalet, Y. (2008). Multiple kernel learning, conic duality, and the smo algorithm. *Journal of Machine Learning Research*.
- [Corinna Cortes and Rostamizadeh, 2009] Corinna Cortes, M. M. and Rostamizadeh, A. (2009). Learning non-linear combinations of kernels. In *Advances in Neural Information Processing Systems*.
- [CORTES and VAPNIK, 1995] CORTES, C. and VAPNIK, V. (1995). Support vector networks. *Machine Learning*.
- [Gonen, 2004] Gonen, M. (2004). *Localized multiple kernel learning for Machine Learning*. PhD thesis.
- [Gonen and Alpaydn, 2004] Gonen, M. and Alpaydn, E. (2004). Localized multiple kernel learning. In *21st International Conference on Machine Learning*.
- [Ladicky and Torr, 2011] Ladicky, L. and Torr, P. H. (2011). Locally linear support vector machines. In *21st International Conference on Machine Learning*.

- [Lanckriet et al., 004a] Lanckriet, G. R. G., Jordan, M. I., Cristianini, N., PeteBartlett, and Ghaoui, L. E. (2004a). Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5(27-72).
- [Vishwanathan et al., 2010] Vishwanathan, S. V. N., Theera-Ampornpunt, N., Sun, Z., and Varma, M. (2010). Multiple kernel learning and the smo algorithm. In *Advances in Neural Information Processing Systems*.