Localized Multiple Kernel Learning - A Convex Approach

Team 13

Introduction

- Kernel-based methods such as support vector machines have found diverse applications.

- The performance of such algorithms, however, crucially depends on the involved kernel function as it intrinsically specifies the feature space where the learning process is implemented, and thus provides a similarity measure on the input space.

- Initially, for a range of data, the kernel function was specified by the user. But here we consider the fact that the data may be diverse and may need different kernel functions based upon the data. Hence we need to learn the kernel function which is the most appropriate for the given input data.

How Is Kernel Learnt ?

- So to choose the best kernel function, we basically choose a set of Kernels (Multiple Kernels) and we give weights so as to choose the best kernel for the data.

Clearly if we were to use a color kernel, it would work fine for the first image, but for the second image it would not work properly as the horse coincides with the background

Previous Approaches

- Existing approaches to localized MKL optimize non-convex objective functions. This puts their generalization ability into doubt.

- Indeed the generalization performance of localized MKL algorithms (as measured through large-deviation bounds) is poorly understood, which potentially could make these algorithms prone to overfitting

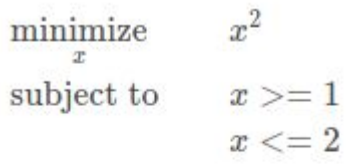
- Further potential disadvantages of non-convex localized MKL approaches include
    - Computational Difficulty in finding good local minima
    - The Induced lack of reproducibility of results (due to varying local optima).

Present Approach

- This paper presents a convex formulation of localized multiple kernel learning, which is formulated as a single convex optimization problem over a precomputed cluster structure.
- We derive an efficient optimization algorithm based on Dual Problem.
- Computational experiments on data from the domains of computational biology and computer vision show that the proposed convex approach can achieve higher prediction accuracies than its global and non-convex local counterparts.
- The Approaches to localized MKL are formulated in terms of non-convex optimization problems, and deep theoretical foundations in the form of generalization error or excess risk bounds are unknown.

Advantages of Convex Curves

- The main advantage of Convex Curves over others is that it has at most one Maxima or Minima, hence making it a global one and thus our Minimization or Maximization Problem would be easier when compared to the case where there are many local minimas or maximas.



$$\begin{array}{ll} \underset{x}{\text{minimize}} & x^2 \\ \text{subject to} & x >= 1 \\ & x <= 2 \end{array}$$

- Another advantage is that, in a constrained problem, a convex feasible region makes it easier to ensure that you do not generate infeasible solutions while searching for an optimum. If you have two feasible solutions, any solution within the line segment connecting them is feasible.

Multiple Kernel Learning

- Multiple kernel learning refers to a set of machine learning methods that use a predefined set of kernels and learn an optimal linear or non-linear combination of kernels as part of the algorithm.
- Reasons to use multiple kernel learning include
  - The ability to select for an optimal kernel and parameters from a larger set of kernels, reducing bias due to kernel selection while allowing for more automated machine learning methods
  - Combining data from different sources (e.g. sound and images from a video) that have different notions of similarity and thus require different kernels.
- Instead of creating a new kernel, multiple kernel algorithms can be used to combine kernels already established for each individual data source.

Multiple Kernel Learning (Contd.)

- Existing approaches to localized MKL optimize non-convex objective functions, whereas here we introduce a Convex Optimization Approach

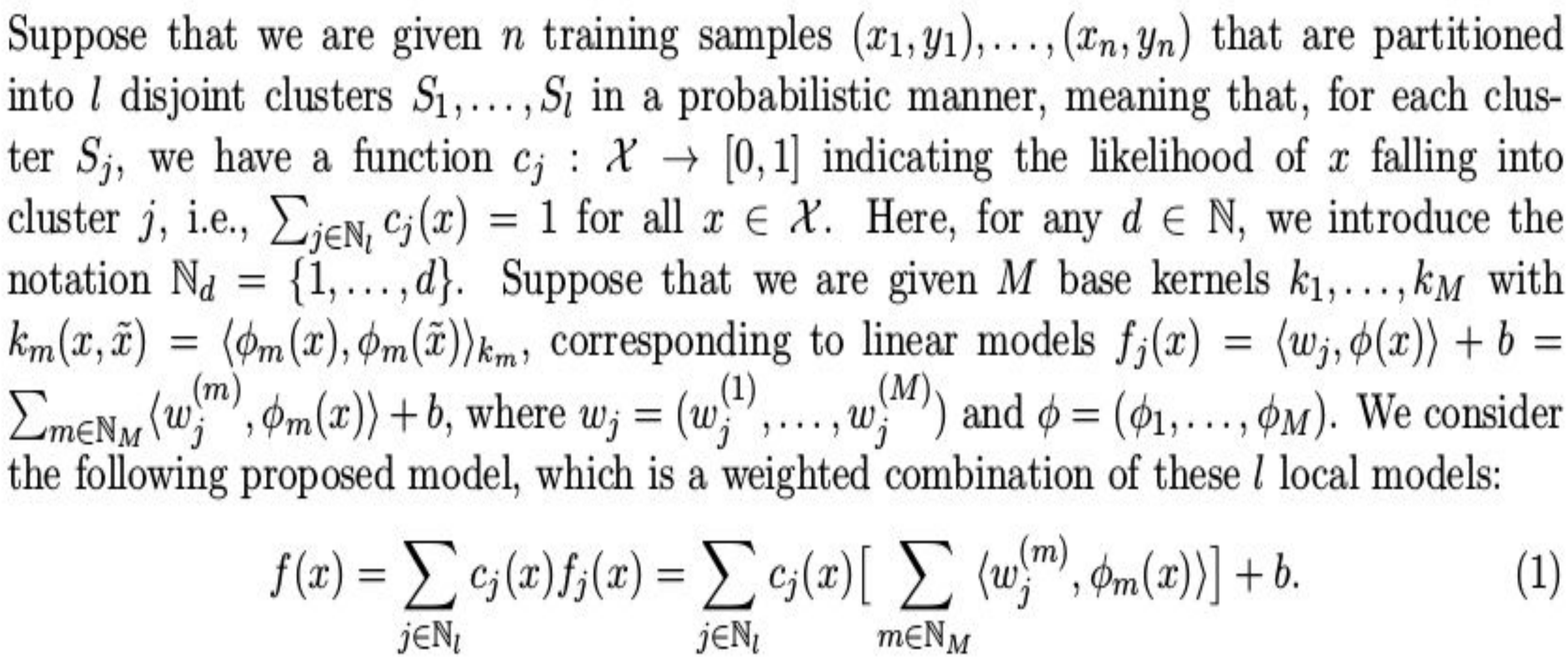
- G¨onen and Alpaydin initiated the work on localized MKL by introducing

$$f(x) = \sum_{m=1} \eta_m(x;v)\langle w_m, \phi_m(x)\rangle + b, \quad \eta_m(x;v) \propto \exp(\langle v_m, x\rangle + v_{m0})$$

- Here in the above equation, “eta” is a weight function for the Kernel chosen, giving a brief idea of how apt that specific kernel works for the given input data.

- Later other approaches involving clustering of the data and choosing the best kernels for a cluster were introduced.

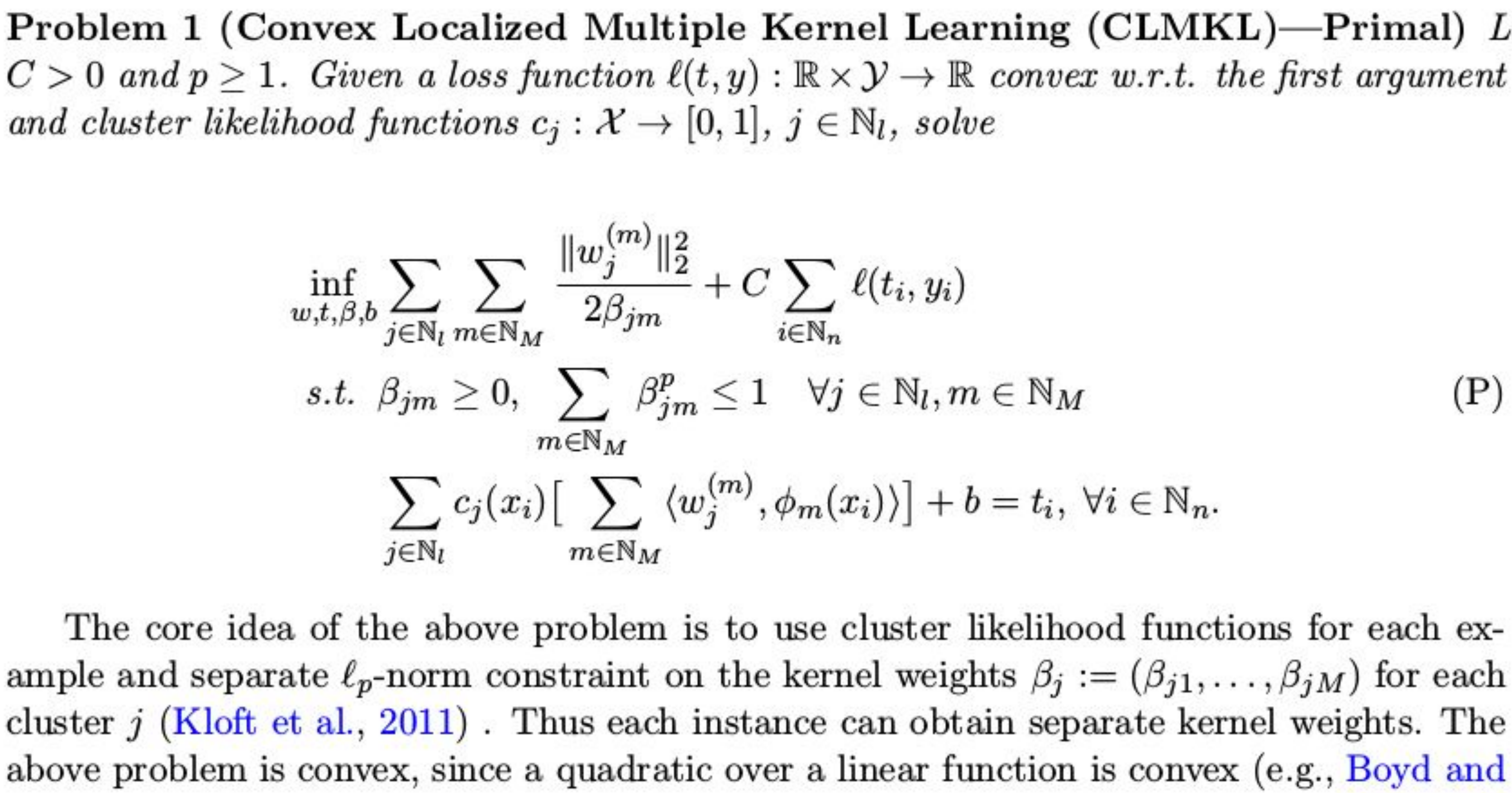
Convex Localized Multiple Kernel Learning



Suppose that we are given n training samples $(x_1, y_1), \ldots, (x_n, y_n)$ that are partitioned into l disjoint clusters $S_1, \ldots, S_l$ in a probabilistic manner, meaning that, for each cluster S_j, we have a function $c_j : \mathcal{X} \to [0, 1]$ indicating the likelihood of x falling into cluster j, i.e., $\sum_{j \in \mathbb{N}_l} c_j(x) = 1$ for all $x \in \mathcal{X}$. Here, for any $d \in \mathbb{N}$, we introduce the notation $\mathbb{N}_d = \{1, \ldots, d\}$. Suppose that we are given M base kernels $k_1, \ldots, k_M$ with $k_m(x, \tilde{x}) = \langle \phi_m(x), \phi_m(\tilde{x}) \rangle_{k_m}$, corresponding to linear models $f_j(x) = \langle w_j, \phi(x) \rangle + b = \sum_{m \in \mathbb{N}_M} \langle w_j^{(m)}, \phi_m(x) \rangle + b$, where $w_j = (w_j^{(1)}, \ldots, w_j^{(M)})$ and $\phi = (\phi_1, \ldots, \phi_M)$. We consider the following proposed model, which is a weighted combination of these l local models:

$$f(x) = \sum_{j \in \mathbb{N}_l} c_j(x) f_j(x) = \sum_{j \in \mathbb{N}_l} c_j(x) \Big[\sum_{m \in \mathbb{N}_M} \langle w_j^{(m)}, \phi_m(x) \rangle \Big] + b. \qquad (1)$$

Primal Form of the Problem



Problem 1 (Convex Localized Multiple Kernel Learning (CLMKL)—Primal) *Let $C > 0$ and $p \geq 1$. Given a loss function $\ell(t, y) : \mathbb{R} \times \mathcal{Y} \to \mathbb{R}$ convex w.r.t. the first argument and cluster likelihood functions $c_j : \mathcal{X} \to [0, 1]$, $j \in \mathbb{N}_l$, solve*

$$\begin{aligned} &\inf_{w,t,\beta,b} \sum_{j\in\mathbb{N}_l} \sum_{m\in\mathbb{N}_M} \frac{\|w_j^{(m)}\|_2^2}{2\beta_{jm}} + C \sum_{i\in\mathbb{N}_n} \ell(t_i, y_i) \\ &\text{s.t.}\ \ \beta_{jm} \geq 0, \sum_{m\in\mathbb{N}_M} \beta_{jm}^p \leq 1 \quad \forall j \in \mathbb{N}_l, m \in \mathbb{N}_M \\ &\qquad \sum_{j\in\mathbb{N}_l} c_j(x_i) \Big[\sum_{m\in\mathbb{N}_M} \langle w_j^{(m)}, \phi_m(x_i) \rangle \Big] + b = t_i,\ \forall i \in \mathbb{N}_n. \end{aligned} \tag{P}$$

The core idea of the above problem is to use cluster likelihood functions for each example and separate ℓ_p-norm constraint on the kernel weights $\beta_j := (\beta_{j1}, \ldots, \beta_{jM})$ for each cluster j (Kloft et al., 2011) . Thus each instance can obtain separate kernel weights. The above problem is convex, since a quadratic over a linear function is convex (e.g., Boyd and

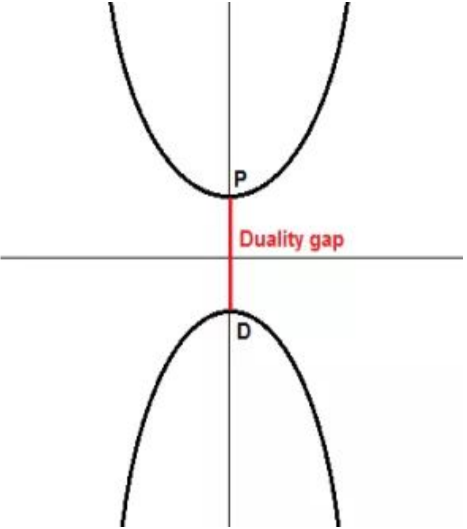
Understanding Duality

- In mathematical optimization theory, Duality means that optimization problems may be viewed from either of two perspectives, the primal problem or the dual problem (the duality principle). The solution to the dual problem provides a lower bound to the solution of the primal (minimization) problem.

- This means that if you have a minimization problem, you can also see it as a maximization problem. And when you find the maximum of this problem, it will be a lower bound to the solution of the minimization problem, i.e. it will always be less than or equal to the minimum of the minimization problem.

Why do we care about Duality?

- It turns out that sometimes, solving the dual problem is simpler than solving the primal problem.
- Let us see an example for better understanding
  Here imagine that in our primal problem, we are trying to minimize the function at the top of the graph. Its minimum is the point P.
  If we search for a dual function, we could end up with the one at the bottom of the graph, whose maximum is the Point D. Here we can clearly see that D is a lower bound.
- We define the distance between P and D as the **Duality Gap.** In this example, P−D>0 and we say that weak duality holds.
- If there is no duality gap, then we say that strong duality Holds. In this case, the maxima of Dual Problems is same as The minima of the Primal Problem



Understanding Lagrange Multipliers

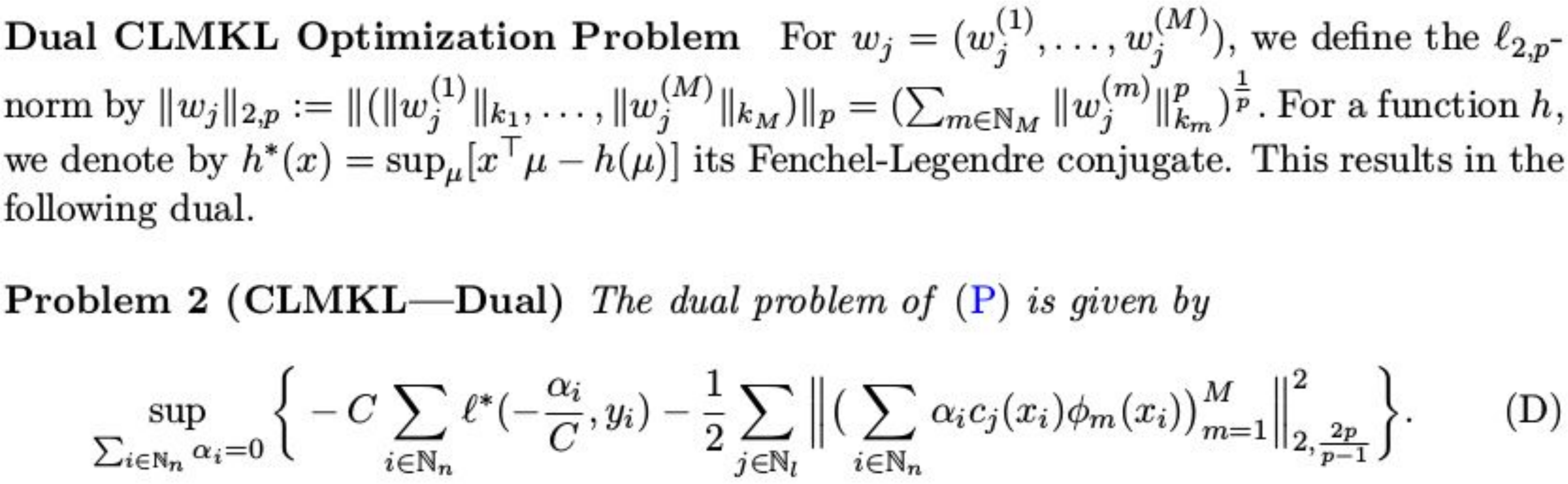
- In mathematical optimization, the method of Lagrange multipliers is a strategy for finding the local maxima and minima of a function subject to equality constraints.
- For the case of only one constraint and only two choice variables consider the optimization problem

  maximize f(x, y)
  subject to g(x, y) = 0
- We introduce a new variable (λ) called a Lagrange multiplier and study the Lagrange function (or Lagrangian or Lagrangian expression) defined by

$$\mathcal{L}(x, y, \lambda) = f(x, y) - \lambda \cdot g(x, y),$$

- We then derivate the above with respect to x,y,lambda to get the optimized value.

Dual CLMKL Optimization Problem



Dual CLMKL Optimization Problem For $w_j = (w_j^{(1)}, \ldots, w_j^{(M)})$, we define the $\ell_{2,p}$-norm by $\|w_j\|_{2,p} := \|(\|w_j^{(1)}\|_{k_1}, \ldots, \|w_j^{(M)}\|_{k_M})\|_p = (\sum_{m\in\mathbb{N}_M} \|w_j^{(m)}\|_{k_m}^p)^{\frac{1}{p}}$. For a function h, we denote by $h^*(x) = \sup_\mu [x^\top \mu - h(\mu)]$ its Fenchel-Legendre conjugate. This results in the following dual.

Problem 2 (CLMKL—Dual) *The dual problem of* (P) *is given by*

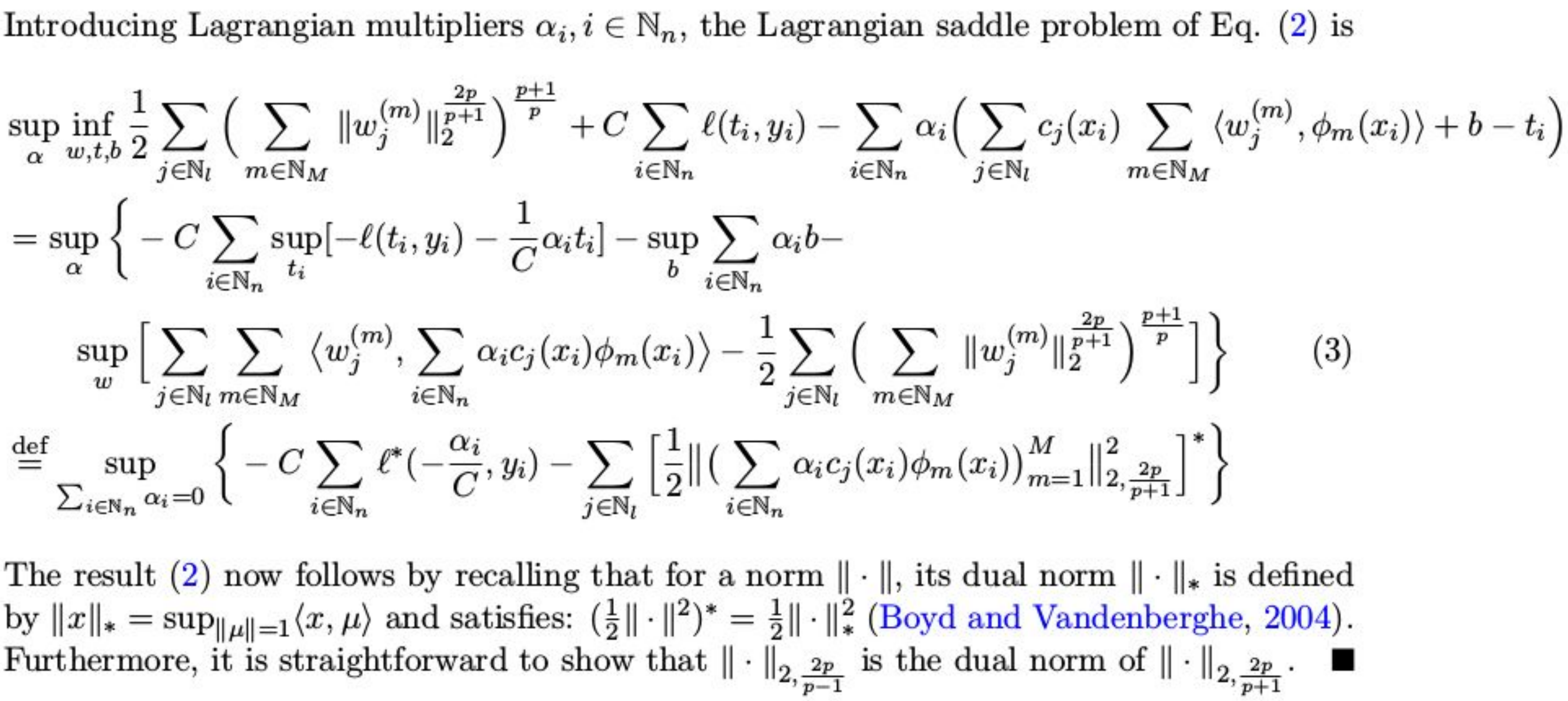
$$\sup_{\sum_{i\in\mathbb{N}_n} \alpha_i = 0} \Big\{ -C \sum_{i\in\mathbb{N}_n} \ell^*(-\frac{\alpha_i}{C}, y_i) - \frac{1}{2} \sum_{j\in\mathbb{N}_l} \Big\| \big(\sum_{i\in\mathbb{N}_n} \alpha_i c_j(x_i) \phi_m(x_i) \big)_{m=1}^M \Big\|_{2,\frac{2p}{p-1}}^2 \Big\}. \tag{D}$$

Proof of Dual Form of the Problem

The Problem P is equivalent to

$$\begin{aligned}
&\inf_{w,t,b} \frac{1}{2}\sum_{j\in\mathbb{N}_l}\Big(\sum_{m\in\mathbb{N}_M}\|w_j^{(m)}\|_2^{\frac{2p}{p+1}}\Big)^{\frac{p+1}{p}} + C\sum_{i\in\mathbb{N}_n}\ell(t_i, y_i)\\
&\text{s.t.} \quad \sum_{j\in\mathbb{N}_l}\Big[c_j(x_i)\sum_{m\in\mathbb{N}_M}\langle w_j^{(m)}, \phi_m(x_i)\rangle\Big] + b = t_i, \ \forall i \in \mathbb{N}_n.
\end{aligned}$$

Proof (Contd.)



Introducing Lagrangian multipliers $\alpha_i, i \in \mathbb{N}_n$, the Lagrangian saddle problem of Eq. (2) is

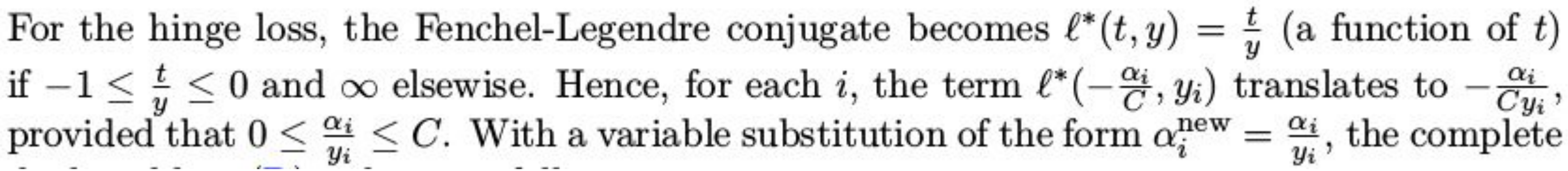
$$\begin{aligned}
&\sup_{\alpha} \inf_{w,t,b} \frac{1}{2} \sum_{j \in \mathbb{N}_l} \Big(\sum_{m \in \mathbb{N}_M} \|w_j^{(m)}\|_2^{\frac{2p}{p+1}} \Big)^{\frac{p+1}{p}} + C \sum_{i \in \mathbb{N}_n} \ell(t_i, y_i) - \sum_{i \in \mathbb{N}_n} \alpha_i \Big(\sum_{j \in \mathbb{N}_l} c_j(x_i) \sum_{m \in \mathbb{N}_M} \langle w_j^{(m)}, \phi_m(x_i) \rangle + b - t_i \Big) \\
&= \sup_{\alpha} \Big\{ -C \sum_{i \in \mathbb{N}_n} \sup_{t_i} [-\ell(t_i, y_i) - \frac{1}{C} \alpha_i t_i] - \sup_{b} \sum_{i \in \mathbb{N}_n} \alpha_i b - \\
&\quad \sup_{w} \Big[\sum_{j \in \mathbb{N}_l} \sum_{m \in \mathbb{N}_M} \langle w_j^{(m)}, \sum_{i \in \mathbb{N}_n} \alpha_i c_j(x_i) \phi_m(x_i) \rangle - \frac{1}{2} \sum_{j \in \mathbb{N}_l} \Big(\sum_{m \in \mathbb{N}_M} \|w_j^{(m)}\|_2^{\frac{2p}{p+1}} \Big)^{\frac{p+1}{p}} \Big] \Big\} \qquad (3) \\
&\overset{\text{def}}{=} \sup_{\sum_{i \in \mathbb{N}_n} \alpha_i = 0} \Big\{ -C \sum_{i \in \mathbb{N}_n} \ell^*(-\frac{\alpha_i}{C}, y_i) - \sum_{j \in \mathbb{N}_l} \Big[\frac{1}{2} \Big\| \Big(\sum_{i \in \mathbb{N}_n} \alpha_i c_j(x_i) \phi_m(x_i) \Big)_{m=1}^{M} \Big\|_{2, \frac{2p}{p+1}}^2 \Big]^* \Big\}
\end{aligned}$$

The result (2) now follows by recalling that for a norm $\|\cdot\|$, its dual norm $\|\cdot\|_*$ is defined by $\|x\|_* = \sup_{\|\mu\|=1} \langle x, \mu \rangle$ and satisfies: $(\frac{1}{2}\|\cdot\|^2)^* = \frac{1}{2}\|\cdot\|_*^2$ (Boyd and Vandenberghe, 2004). Furthermore, it is straightforward to show that $\|\cdot\|_{2,\frac{2p}{p-1}}$ is the dual norm of $\|\cdot\|_{2,\frac{2p}{p+1}}$. ■

Loss Function and its Conjugate

- Here we are using the Hinge Loss Function, which is as follows

$$\ell(y) = \max(0, 1 - t \cdot y)$$



For the hinge loss, the Fenchel-Legendre conjugate becomes $\ell^*(t, y) = \frac{t}{y}$ (a function of t) if $-1 \leq \frac{t}{y} \leq 0$ and ∞ elsewise. Hence, for each i, the term $\ell^*(-\frac{\alpha_i}{C}, y_i)$ translates to $-\frac{\alpha_i}{C y_i}$, provided that $0 \leq \frac{\alpha_i}{y_i} \leq C$. With a variable substitution of the form $\alpha_i^{\text{new}} = \frac{\alpha_i}{y_i}$, the complete

- Show Proof of the Conjugate Loss Function

SVM Formulation

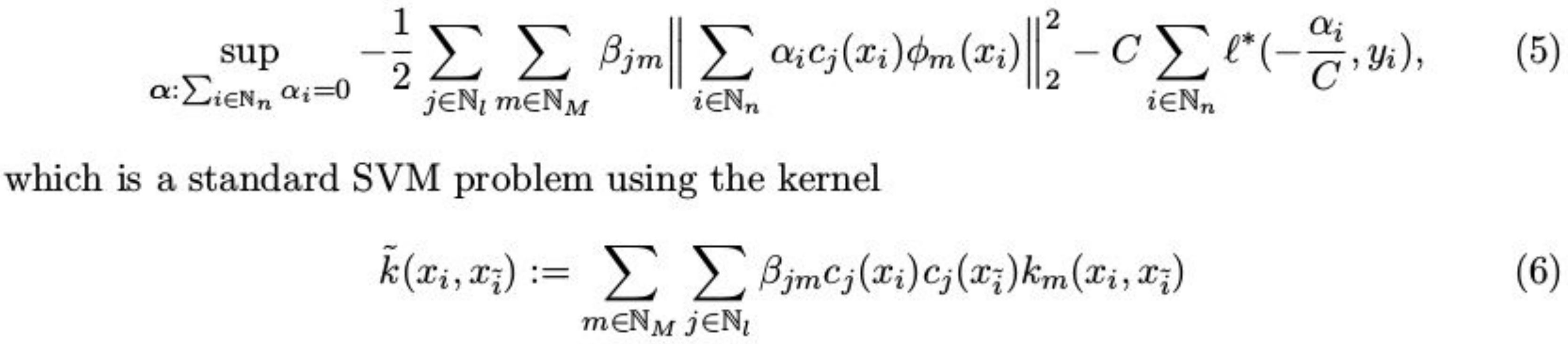
- Now we convert the formula to the SVM Formulation form which is obtained by replacing alpha with alpha new

$$\sup_{\alpha:0\leq\alpha\leq C,\sum_{i\in\mathbb{N}_n}\alpha_i y_i=0} -\frac{1}{2}\sum_{j\in\mathbb{N}_l}\Big\|\Big(\sum_{i\in\mathbb{N}_n}\alpha_i y_i c_j(x_i)\phi_m(x_i)\Big)_{m=1}^{M}\Big\|_{2,\frac{2p}{p-1}}^{2} + \sum_{i\in\mathbb{N}_n}\alpha_i,$$

Optimization Algorithms

- we consider here a two-layer optimization procedure to solve the problem (P) where the variables are divided into two groups:
  - The group of kernel weights $\{\beta_{jm}\}_{j,m=1}^{l,M}$
  - The group of weight vectors $\{w_j^{(m)}\}_{j,m=1}^{l,M}$

- Given Fixed Kernel Weights, the CLMKL dual problem is given by



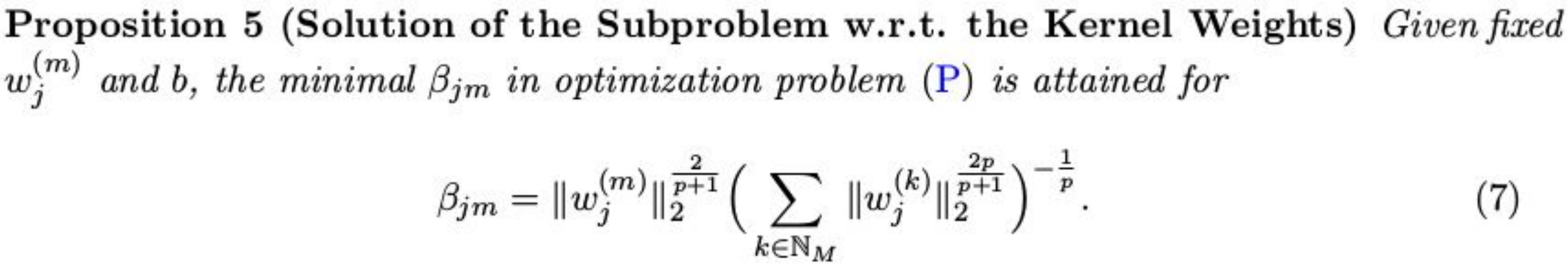
$$\sup_{\alpha:\sum_{i\in\mathbb{N}_n}\alpha_i=0} -\frac{1}{2}\sum_{j\in\mathbb{N}_l}\sum_{m\in\mathbb{N}_M}\beta_{jm}\Big\|\sum_{i\in\mathbb{N}_n}\alpha_i c_j(x_i)\phi_m(x_i)\Big\|_2^2 - C\sum_{i\in\mathbb{N}_n}\ell^*(-\frac{\alpha_i}{C},y_i), \qquad (5)$$

which is a standard SVM problem using the kernel

$$\tilde{k}(x_i,x_{\tilde{i}}) := \sum_{m\in\mathbb{N}_M}\sum_{j\in\mathbb{N}_l}\beta_{jm}c_j(x_i)c_j(x_{\tilde{i}})k_m(x_i,x_{\tilde{i}}) \qquad (6)$$

Optimization Algorithms (Contd.)

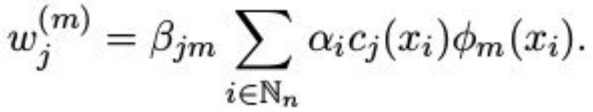
- The subproblem of optimizing the kernel weights for fixed $w_j^{(m)}$ and b has a closed-form solution.



Proposition 5 (Solution of the Subproblem w.r.t. the Kernel Weights) *Given fixed $w_j^{(m)}$ and b, the minimal β_{jm} in optimization problem (P) is attained for*

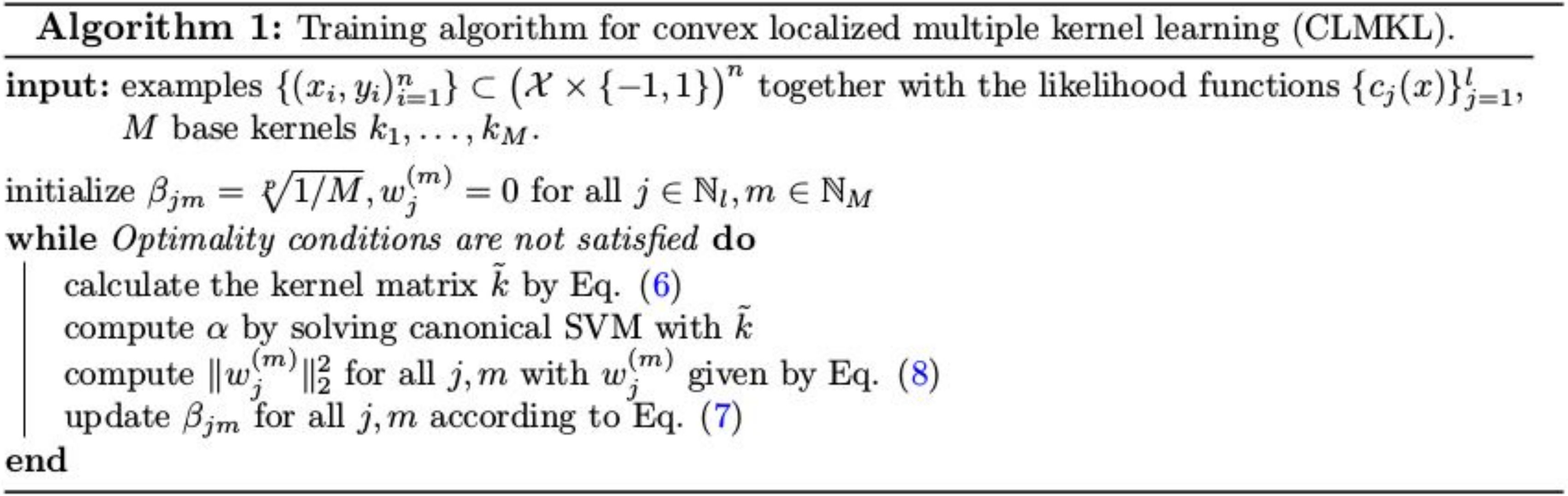
$$\beta_{jm} = \|w_j^{(m)}\|_2^{\frac{2}{p+1}} \Big(\sum_{k \in \mathbb{N}_M} \|w_j^{(k)}\|_2^{\frac{2p}{p+1}} \Big)^{-\frac{1}{p}}. \tag{7}$$

- for updating β_{jm} , we need to compute the norm of $w_j^{(m)}$, and this can be accomplished by the following representation of $w_j^{(m)}$ given fixed β_{jm}



$$w_j^{(m)} = \beta_{jm} \sum_{i \in \mathbb{N}_n} \alpha_i c_j(x_i) \phi_m(x_i).$$

Pseudo Code



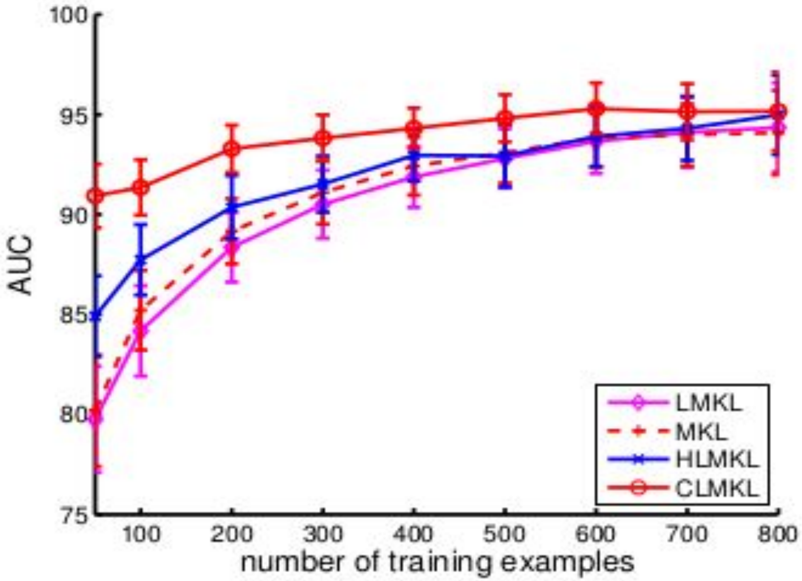
Algorithm 1: Training algorithm for convex localized multiple kernel learning (CLMKL).

```
input: examples $\{(x_i, y_i)_{i=1}^n\} \subset \left(\mathcal{X} \times \{-1,1\}\right)^n$ together with the likelihood functions $\{c_j(x)\}_{j=1}^l$,
       $M$ base kernels $k_1, \ldots, k_M$.
initialize $\beta_{jm} = \sqrt[p]{1/M}, w_j^{(m)} = 0$ for all $j \in \mathbb{N}_l, m \in \mathbb{N}_M$
while Optimality conditions are not satisfied do
    calculate the kernel matrix $\tilde{k}$ by Eq. (6)
    compute $\alpha$ by solving canonical SVM with $\tilde{k}$
    compute $\|w_j^{(m)}\|_2^2$ for all $j, m$ with $w_j^{(m)}$ given by Eq. (8)
    update $\beta_{jm}$ for all $j, m$ according to Eq. (7)
end
```

Runtime Complexity Analysis

- At each iteration, we have the following complexities.
  Here L is the number of clusters and M is the number of kernels
  - We need O(n^2Ml) operations to calculate the kernel (Refer Equation in above Slides)
  - We need O(n^2 n_s) operations to solve a standard SVM problem, where n_s is the number of support vectors in the SVM.
  - O(Mln^2_s) operations to calculate the norm according to the representation (Refer Equation in above Slides)
  - O(Ml) operations to update the kernel weights (Refer Equation in above Slides)
- Thus, the computational cost at each iteration for testing is O(n^2Ml).
- The computational cost for testing phase is O(n_tMl), where n_t is the number of testing data points

Experimental Results



LMKL means Localized Multiple Kernel Learning, CLMKL means Convex Localized Multiple Kernel Learning, HLMKL means Hard Clustered Localized Multiple Kernel Learning.

Conclusions

- Localized approaches to multiple kernel learning (LMKL) allow for flexible distribution of kernel weights over the input space. This can be a great advantage when samples require varying kernel importance.
- However, almost prevalent approaches to localized MKL require solving difficult non-convex optimization problems, which makes them potentially prone to overfitting as theoretical guarantees such as generalization error bounds are yet unknown.
- In this paper, we propose a theoretically grounded approach to localized MKL, consisting of two subsequent steps
    - Clustering the training instances
    - Computation of the kernel weights for each cluster through a single convex optimization problem.
    - Now we derive an efficient optimization algorithm based on Fenchel duality

Thank You