Journal of Machine Learning Research 12 (2011) 565-592 Submitted 12/09; Revised 10/10; Published 2/11

Variable Sparsity Kernel Learning

Jonathan Aflalo YAFLALO@CS.TECHNION.AC.IL
Faculty of Computer Science
Technion–Israel Institute of Technology
Haifa 32000, ISRAEL

Aharon Ben-Tal ABENTAL@IE.TECHNION.AC.IL
Faculty of Industrial Engineering and Management
Technion–Israel Institute of Technology
Haifa 32000, ISRAEL

Chiranjib Bhattacharyya CHIRU@CSA.IISC.ERNET.IN
Department of Computer Science and Automation
Indian Institute of Science
Bangalore 560012, INDIA

Jagarlapudi Saketha Nath SAKETH@CSE.IITB.AC.IN
Department of Computer Science and Engg.
Indian Institute of Technology Bombay
Mumbai 400076, INDIA

Sankaran Raman RAMANS@CSA.IISC.ERNET.IN
Department of Computer Science and Automation
Indian Institute of Science
Bangalore 560012, INDIA

Editor: Soeren Sonnenburg

Abstract

This paper[1] presents novel algorithms and applications for a particular class of mixed-norm regularization based Multiple Kernel Learning (MKL) formulations. The formulations assume that the given kernels are grouped and employ l_1 norm regularization for promoting sparsity within RKHS norms of each group and $l_s, s \geq 2$ norm regularization for promoting non-sparse combinations across groups. Various sparsity levels in combining the kernels can be achieved by varying the grouping of kernels—hence we name the formulations as Variable Sparsity Kernel Learning (VSKL) formulations. While previous attempts have a non-convex formulation, here we present a convex formulation which admits efficient Mirror-Descent (MD) based solving techniques. The proposed MD based algorithm optimizes over product of simplices and has a computational complexity of $O\left(m^2 n_{\text{tot}} \log n_{\max}/\varepsilon^2\right)$ where m is no. training data points, $n_{\max}, n_{\text{tot}}$ are the maximum no. kernels in any group, total no. kernels respectively and ε is the error in approximating the objective. A detailed proof of convergence of the algorithm is also presented. Experimental results show that the VSKL formulations are well-suited for multi-modal learning tasks like object categorization. Results also show that the MD based algorithm outperforms state-of-the-art MKL solvers in terms of computational efficiency.

Keywords: multiple kernel learning, mirror descent, mixed-norm, object categorization, scalability

[1] All authors contributed equally. The author names appear in alphabetical order.

1. Introduction

This paper studies the problem of Multiple Kernel Learning (MKL) (Lanckriet et al., 2004; Bach et al., 2004; Sonnenburg et al., 2006; Rakotomamonjy et al., 2008) when the given kernels are assumed to be grouped into distinct components. Further, the focus is on the scenario where prior/domain knowledge warrants that each component is crucial for the learning task at hand. One of the key contributions of this paper is a highly efficient learning algorithm for this problem.

Recently Szafranski et al. (2008) extended the framework of MKL to the case where kernels are partitioned into groups and introduced a generic mixed-norm (that is (r,s)-norm; $r,s \geq 0$) regularization based MKL formulation (refer (11) in Szafranski et al., 2008) in order to handle groups of kernels. The idea is to employ a r-norm regularization over RKHS norms for kernels belonging to the same group and a s-norm regularization across groups. Though a generic formulation was presented, the focus of Szafranski et al. (2008) was on applications where it is known that most of the groups of kernels are noisy/redundant and hence only those mixed-norms promoting sparsity among kernels within and across groups were employed, for example, $0 < r,s < 2$ (following the terminology of Szafranski et al. (2008) this class of mixed-norm MKL formulations are henceforth called as “Composite Kernel Learning (CKL)” formulations). This paper presents a complementary study and focuses on applications where the domain knowledge guarantees that every group of kernels is crucial. Needless to say, all the groups of kernels need not be “equally” important and not all kernels belonging to a group may be important. More specifically, the focus of this paper is on the cases where $r = 1$ and $s \geq 2$ (including the limiting case[2] $s = \infty$). Here, $p = 1$ is employed for promoting sparsity among kernels belonging to the same group and $s \geq 2$ for promoting non-sparse combinations of kernels across groups. Note that the extreme cases: a) all of the kernels belong to one group b) Each group consists of a single kernel; correspond to the extreme sparse and non-sparse combinations of the given kernels. Since by varying the values of s and the groupings of kernels various levels of sparsity in combining the given kernels can be achieved, the formulations studied here are henceforth called as “Variable Sparsity Kernel Learning” (VSKL) formulations. As mentioned earlier, VSKL formulations are not well-studied in literature and this paper presents novel algorithms and applications for these formulations.

The VSKL formulations are motivated by multi-modal learning applications like object categorization where multiple feature representations need to be employed simultaneously for achieving good generalization. For instance, in the case of flower categorization feature descriptors for shape, color and texture need to be employed in order to achieve good visual discrimination as well as significant within-class variation (Nilsback and Zisserman, 2006). Combining feature descriptors for object categorization using the framework of MKL for object categorization has been a topic of interest for many recent studies (Varma and Ray, 2007; Nilsback and Zisserman, 2008) and is shown to achieve state-of-the-art performance. A key finding of Nilsback and Zisserman (2006) is the following: in object categorization tasks, employing few of the feature descriptors or employing a canonical combination of them often leads to sub-optimal solutions. Hence, in the framework of MKL, employing a block l_1 regularization, which is equivalent to selecting the “best” among the given kernels, as well as employing a l_2 regularization, which is equivalent to working with a canonical combination of the given kernels, may lead to sub-optimality. This observation clearly shows that state-of-the-art object categorization techniques (which are based on block l_1 regularized formulation) can further be improved. This paper proposes to employ the VSKL formulations for

[2] This limiting case was discussed in an earlier version of this paper (Nath et al., 2009).

object categorization where the kernel are grouped based on the feature descriptor generating them. The $l_s (s \geq 2)$-norm regularization leads to non-sparse combinations of kernels generated from different feature descriptors and the l_1 norm leads to sparse selection of non-redundant/noisy kernels generated from a feature descriptor.

With this motivation, the key aspect investigated in this paper is an efficient algorithm for solving the VSKL formulations which are instances of non-smooth convex optimization problems. Except in the cases where $s = 2$ or $s = \infty$ or no. groups is one, the formulations cannot be solved using standard interior point based convex optimization software. Moreover, even in these special cases the generic interior point algorithms do not scale well to large data sets. The wrapper approach presented in Szafranski et al. (2008) cannot be employed for solving the VSKL formulations (that is, with $l_s, s \geq 2$ regularization across groups) efficiently as it solves a non-convex variant of the original convex formulation! The methods discussed in Szafranski et al. (2008); Kloft et al. (2010) are however efficient in the case $1 \leq s < 2$ (that is, sparse regularization across groups). In summary, efficient techniques for solving VSKL formulations indeed need to be devised. This paper adapts the Mirror-Descent (MD) (Ben-Tal et al., 2001; Beck and Teboulle, 2003; Ben-Tal and Nemirovski, 2001) procedure for solving a specific dual of VSKL leading to extremely scalable algorithms. MD is similar in spirit to the steepest descent algorithm; however involves a prox-function based regularizer rather than Euclidean norm based regularizer in the per-step auxiliary problem solved at each iteration. The prox-function is cleverly chosen based on the geometry of the feasibility set. Here, the feasibility set for the optimization problem tackled by MD turns out to be direct product of simplices, which is not a standard set-up discussed in optimization literature. We propose to employ the entropy function as the prox-function in the auxiliary problem solved by MD at each iteration and justify its suitability for the case of direct product of simplices. The MD based procedure for solving the dual of VSKL is henceforth called as `mirrorVSKL`.

Apart from the derivation of the `mirrorVSKL` algorithm, we also provide a detailed proof of its asymptotic convergence. `mirrorVSKL` is also of independent interest to the MKL community as it can solve the traditional MKL problem; namely the case when the number of groups is unity. The key advantages of `mirrorVSKL` over `simpleMKL` are: **a)** In case of `simpleMKL` in addition to gradient computation, the reduced gradient and step-size need to be determined which requires substantial computational effort; whereas in case of `mirrorVSKL`, pre-dominant computation at each iteration is that of calculating the gradient since the auxiliary problem has an analytical solution and the step-size can be computed easily **b)** It can be shown that the number of iterations with `mirrorVSKL` is nearly-independent of the number of kernels whereas no such a statement can be made in case of `simpleMKL`.

Simulations were performed on three real-world object categorization data sets: Caltech-101 (Fei-Fei et al., 2004), Caltech-256 (Griffin et al., 2007) and Oxford flowers (Nilsback and Zisserman, 2006) for comparing the generalization ability of the VSKL and existing MKL formulations. The results show that the proposed formulation are well-suited for multi-modal tasks like object categorization. In the special case of number of groups unity, the `mirrorVSKL` and `simpleMKL` algorithms were compared in terms of computational effort and scalability. The scalability experiments were performed on few UCI data sets (Blake and Merz, 1998) following the experimental set-up of Szafranski et al. (2008). Results showed that `mirrorVSKL` scales well to large data sets with large no. kernels and in some cases was eight times faster than `simpleMKL`.

The remainder of this paper is organized as follows: in Section 2, the VSKL and related MKL formulations are presented. The section also presents a specific dual of VSKL which admits efficient

MD based solving techniques. The main contribution of the paper, `mirrorVSKL` is presented in Section 3. A detailed proof of convergence of `mirrorVSKL` is also presented. Section 4 presents a summary of the numerical experiments carried for verifying the major claims of the paper. Section 5 concludes the paper with a brief summary and discussion.

2. Variable Sparsity Kernel Learning Formulation

This section presents the VSKL formulation and a specific dual of it. Though the formalism can be extended to various learning tasks we focus on the task of binary classification in the rest of the paper. We begin by introducing some notation: let the training data set be denoted by $\mathcal{D} = \{(\mathbf{x}_i, y_i),\ i = 1, \ldots, m \mid \mathbf{x}_i \in \mathcal{X},\ y_i \in \{-1, 1\}\}$. Here, $\mathbf{x}_i$ represents the i^{th} training data point with label y_i. Let $\mathbf{Y}$ denote the diagonal matrix with entries as y_i. Suppose the given kernels are divided into n groups and the j^{th} group has n_j number of kernels. Let the feature-space mapping induced by the k^{th} kernel of the j^{th} component be $\phi_{jk}(\cdot)$ and the corresponding gram-matrix of training data points be $\mathbf{K}_{jk}$.[3] Also, let $\mathbf{Q}_{jk} = \mathbf{Y}\mathbf{K}_{jk}\mathbf{Y}$.

For now, to keep things simple, let us assume that each of the kernels is such that the induced feature mapping is finite dimensional; later on we will generalize and remove this assumption. Each individual example can now be described by a concatenation of all the feature vectors:

$$\mathbf{x}^\top = \left[\phi_{11}(\mathbf{x})^\top \ldots \phi_{jk}(\mathbf{x})^\top \ldots \phi_{nn_n}(\mathbf{x})^\top\right].$$

Consider the problem of learning a linear discriminant function of the form

$$f(\mathbf{x}) = \sum_{j=1}^{n} \sum_{k=1}^{n_j} \mathbf{w}_{jk}^\top \phi_{jk}(\mathbf{x}) - b.$$

Given a training set the idea is to learn a $\mathbf{w} \equiv [\mathbf{w}_{11}^\top \mathbf{w}_{12}^\top \ldots \mathbf{w}_{nn_n}^\top]^\top$ and b which generalizes well. This could be achieved by minimizing an objective of the form (Vapnik, 1998):

$$J(\mathbf{w}) = \Omega(\mathbf{w}) + C\, L(\mathcal{D}),$$

where Ω is a suitable regularizing function, L is a loss function which penalizes errors on the training set and C is a regularization parameter. SVMs (Vapnik, 1998) usually use $\Omega(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|_2^2$ and $L = \sum_{i=1}^{m} \max(1 - y_i(\mathbf{w}^\top \mathbf{x}_i - b), 0)$. It is easy to see that this formulation corresponds to employing a kernel which is essentially the sum of all the given kernels. Hence-forth, we denote this formulation by **SVM** and use it as a baseline for comparison in the experiments.

The regularization term can be an extremely useful tool for modelling various different kinds of data. The choice of Ω should be such that this is tractable and yet flexible enough to enforce different relationships between groups dictated by modelling requirements. Recently (Szafranski et al., 2008) employed a regularization of the form

$$\frac{1}{2}\|\mathbf{w}\|_{r,s}^2 \; 0 \le r < 2, 0 \le s < 2,$$

where

$$\|\mathbf{w}\|_{(r,s)} = \left\{ \sum_{j=1}^{n} \left\{ \sum_{k=1}^{n_j} \|\mathbf{w}_{jk}\|_2^r \right\}^{\frac{s}{r}} \right\}^{\frac{1}{s}}.$$

[3] The gram-matrices are unit-trace normalized.

Since the primary goal of Szafranski et al. (2008) is to achieve sparsity, the focus was only on the cases $0 \leq r < 2, 0 \leq s < 2$ making most of the individual norms $\|\mathbf{w}_{jk}\|$ zero at optimality. Henceforth, this formulation is denoted by $\mathbf{CKL}_{r,s}$ where r, s represent the within and across group norms respectively.

However as discussed above in case of multi-modal tasks like object categorization, it is often desirable that there is sparsity within the group but all the groups need be active. In view of this we begin by defining

$$\Omega_{(p,q)}(\mathbf{w}) = \frac{1}{2}\left\{\sum_{j=1}^{n}\left\{\sum_{k=1}^{n_j}\|\mathbf{w}_{jk}\|_2^{2p}\right\}^{\frac{q}{p}}\right\}^{\frac{1}{q}}.$$

This can be interpreted as a mixed norm operating on $\|\mathbf{w}_{jk}\|^2$ and the following relationship holds

$$\Omega_{(p,q)}(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|_{r,s}^2,\ r = 2p, s = 2q.$$

In this paper we analyze the case $p = \frac{1}{2}$ and $q \geq 1$ which is equivalent to considering an l_1 (sparse) norm regularization within kernels of each group and $l_s(s \geq 2)$ (non-sparse) norm across groups. In other words, we consider the following regularization:

$$\Omega(\mathbf{w}) = \frac{1}{2}\left\{\sum_{j=1}^{n}\left\{\sum_{k=1}^{n_j}\|\mathbf{w}_{jk}\|_2\right\}^{2q}\right\}^{\frac{1}{q}},$$

where $q \geq 1$. By varying the groupings of kernels various levels of sparsity can be achieved: no. of groups is unity corresponds to extreme sparse selection of kernels and no. groups equal to no. kernels corresponds to non-sparse combinations of kernels. The flexibility in choice of q offers different modelling perspectives and correspond to various ways for achieving non-sparse combinations across groups. Since this formulation allows for flexibility from sparsity to non-sparsity, it is called as the Variable Sparsity Kernel Learning (VSKL) formulation and denoted by $\mathbf{VSKL}_q$, where $q \geq 1$:

$$\begin{aligned}\min_{\mathbf{w}_{jk}, b, \xi_i} \quad & \tfrac{1}{2}\left[\textstyle\sum_j \left(\sum_{k=1}^{n_j}\|\mathbf{w}_{jk}\|_2\right)^{2q}\right]^{\frac{1}{q}} + C\textstyle\sum_i \xi_i \\ \text{s.t.} \quad & y_i\left(\textstyle\sum_{j=1}^{n}\sum_{k=1}^{n_j}\mathbf{w}_{jk}^\top \phi_{jk}(\mathbf{x}_i) - b\right) \geq 1 - \xi_i,\ \xi_i \geq 0\ \forall\, i.\end{aligned} \tag{1}$$

In the extreme case $q \to \infty$, the regularization term is to be written as $\frac{1}{2}\max_j\left(\sum_{k=1}^{n_j}\|\mathbf{w}_{jk}\|_2\right)^2$. Note that the traditional MKL formulation using the block l_1 norm regularization (Rakotomamonjy et al., 2008) is a special case of VSKL when the number of groups is unity. We denote this special case by **MKL** and as mentioned earlier, state-of-the-art object categorization performance is achieved using this methodology.

Existing wrapper approaches (Szafranski et al., 2008; Rakotomamonjy et al., 2008) are useful in solving (1) only for the cases $q < 1$. For $1 \leq q < \infty$, the wrapper approaches solve a non-convex variant of the convex formulation and hence are not well-suited. Moreover these wrapper approaches cannot be easily extended to handle the important case $q \to \infty$. In this paper we describe a first order method based on mirror descent procedure which efficiently solves the VSKL formulation for all values of $q \geq 1$ (including $q \to \infty$) and provably converges to the global optimum. The mirror descent procedure solves a specific dual of the VSKL formulation—details of which are presented in the following.

2.1 Dual of VSKL

This section presents a dual of **VSKL** which admits efficient MD solving techniques. In the rest of the paper $q^* = \frac{q}{q-1}$, $q \geq 1$ (if $q = 1$ then $q^* = \infty$ and if $q = \infty$ then $q^* = 1$). If $1 \leq r < \infty$, the following sets $\Delta_{d,r} = \left\{\gamma \equiv [\gamma_1 \dots \gamma_d]^\top \mid \sum_{i=1}^d \gamma_i^r \leq 1, \gamma_i \geq 0, i = 1, \dots, d\right\}$ are convex. As $r \to \infty$ one obtains a d-dimensional box $\Delta_{d,\infty} = B_d = \{\gamma \mid 0 \leq \gamma_i \leq 1 \; i = 1, \dots, d\}$. If $r = 1$ we get back a d-dimensional simplex, and to lighten notation we will denote $\Delta_{d,1} = \Delta_d$. At this point it would be useful to recall the following lemma (see Boyd and Vandenberghe, 2004, Section A.1.6):

Lemma 2.1 *Let* **a** *be a d-dimensional vector with non-negative components, that is, $a_i \geq 0$ $i = 1, \dots, d$. Then*

$$\|\mathbf{a}\|_r = sup_{\gamma \in \Delta_{d,r^*}} \gamma^\top \mathbf{a},$$

where $r \geq 1$ and r^* verifies $\frac{1}{r} + \frac{1}{r^*} = 1$.

A specialization of this lemma for $r \to \infty$ is:

$$\max_i \{a_i\} = sup_{\gamma \in \Delta_d} \gamma^\top \mathbf{a}.$$

We also note the following result which will be used in later derivations (see Micchelli and Pontil, 2005):

Lemma 2.2 *Let $a_i \geq 0, i = 1, \dots, d$ and $1 \leq r < \infty$. Then, for $\Delta_{d,r}$ defined as before,*

$$\min_{\eta \in \Delta_{d,r}} \sum_i \frac{a_i}{\eta_i} = \left(\sum_{i=1}^d a_i^{\frac{r}{r+1}} \right)^{1+\frac{1}{r}},$$

and the minimum is attained at

$$\eta_i = \frac{a_i^{\frac{1}{r+1}}}{\left(\sum_{i=1}^d a_i^{\frac{r}{r+1}}\right)^{\frac{1}{r}}}.$$

Here, by convention, $a/0$ is 0 if $a = 0$ and is ∞ otherwise. In the limit $r \to \infty$ the following holds

$$\min_{\eta \in B_d} \sum_i \frac{a_i}{\eta_i} = \sum_{i=1}^d a_i,$$

where B_d is defined as before and equality is is attained at $\eta_i = 1 \; \forall \; a_i > 0$.

Proof The proof follows by employing the Karush-Kuhn-Tucker conditions, which are here necessary and sufficient for optimality. ■

Using Lemma 2.1, the objective in (1), for any $q \geq 1$, becomes:

$$\frac{1}{2} \max_{\gamma \in \Delta_{n,q^*}} \sum_{j=1}^n \gamma_j \left(\sum_{k=1}^{n_j} \|\mathbf{w}_{jk}\| \right)^2 + C \sum_i \xi_i. \tag{2}$$

For the case $q \to \infty$ the set Δ_{n,q^*} reduces to a simplex, Δ_n. Further, by Lemma 2.2 (with $d = n, r = 1$):

$$\left(\sum_{i=1}^{n} \sqrt{a_i}\right)^2 = \min_{\lambda \in \Delta_n} \sum_{i=1}^{n} \frac{a_i}{\lambda_i},$$

so (2) can be written equivalently as:

$$\max_{\gamma \in \Delta_{n,q^*}} \min_{\lambda_j \in \Delta_{n_j}} \left[\underbrace{\frac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{n_j} \gamma_j \frac{\|\mathbf{w}_{jk}\|^2}{\lambda_{jk}} + C \sum_i \xi_i}_{f(w,\lambda,\gamma,\xi)} \right].$$

The equivalent primal formulation we arrive at is finally

Problem (P)

$$\min_{\xi_i, b, \mathbf{w}_{jk}} \left[\max_{\gamma \in \Delta_{n,q^*}} \min_{\lambda_j \in \Delta_{n_j}} f(\mathbf{w}, \lambda, \gamma, \xi) \right]$$

$$\text{s.t.} \quad y_i \left(\sum_{j=1}^{n} \sum_{k=1}^{n_j} \mathbf{w}_{jk}^T \phi_{jk}(\mathbf{x}_i) - b \right) \geqslant 1 - \xi_i, \ \forall\, i, \tag{3}$$

$$\xi_i \geqslant 0\ , \ \ \forall i. \tag{4}$$

Note that at optimality, the following relations hold

$$\lambda_{jk} = 0 \quad \Rightarrow \quad \mathbf{w}_{jk} = 0,$$
$$\text{if } q \neq \infty, \text{then } \gamma_j = 0 \quad \Leftrightarrow \quad \mathbf{w}_{jk} = 0\ \forall\, k.$$

In case $q = \infty$, $\mathbf{w}_{jk} = 0\ \forall\, k \Rightarrow \gamma_j = 0$ unless $\mathbf{w}_{jk} = 0\ \forall\ j,k$, which is an un-interesting case. Let us fix the variables ξ, b and $\mathbf{w}$ in problem (P) and consider the $\max_\gamma \min_\lambda$ part in the square brackets:

$$\max_{\gamma} \min_{\lambda} \left\{ f(\mathbf{w}, \lambda, \gamma, \xi) \mid \lambda \in \bigotimes_j \Delta_{n_j},\, \gamma \in \Delta_{n,q^*} \right\}.$$

The objective function is concave (linear) in γ and convex in λ, and the feasible sets $\bigotimes_j \Delta_{n_j}, \Delta_{n,q^*}$ are convex and compact. Hence, by the Sion-Kakutani minmax theorem (Sion, 1958), the maxmin can be interchanged, and when this is done, problem (P) becomes

$$\min_{\xi_i, b, \mathbf{w}_{jk}} \min_{\lambda \in \bigotimes_j \Delta_{n_j}} \max_{\gamma \in \Delta_{n,q^*}} f(\mathbf{w}, \lambda, \gamma, \xi)\,, \qquad \text{s.t. (3), (4)},$$

or similarly

$$\min_{\lambda \in \bigotimes_j \Delta_{n_j}} \left[\min_{\xi_i, b, \mathbf{w}_{jk}} \max_{\gamma \in \Delta_{n,q^*}} f(\mathbf{w}, \lambda, \gamma, \xi)\,, \qquad \text{s.t. (3), (4)} \right]. \tag{5}$$

Now, f is convex in $(\xi, b, \mathbf{w})$ and concave (linear) in γ. The set for feasible $(\xi, b, \mathbf{w})$, expressed in (3), (4) is closed and convex, and $\bigotimes_j \Delta_{n_j}$ is convex compact. Hence, by a minmax theorem (Rockafellar, 1964), the minmax in the square brackets in (5) can be interchanged and we arrive at

$$\min_{\lambda \in \bigotimes_j \Delta_{n_j}} \max_{\gamma \in \Delta_{n,q^*}} \left\{ \min_{\xi_i, b, \mathbf{w}_{jk}} f(\mathbf{w}, \lambda, \gamma, \xi) \mid \ \text{s.t. (3), (4)} \right\}. \tag{6}$$

Replacing the convex problem in the curly brackets in (6) by its dual the following theorem is immediate:

Theorem 2.3 *Let $\mathbf{Q}_{jk}$ be the $m \times m$ matrix*

$$\left(\mathbf{Q}_{jk}\right)_{ih} = y_h y_i \phi_{jk}(\mathbf{x}_i)^\top \phi_{jk}(\mathbf{x}_h) \qquad i,h = 1,\ldots,m.$$

The dual problem of (P) w.r.t. the variables $(\mathbf{w}, b, \xi)$ is the following:[4]

$$\text{Problem (D)} \left\{ \begin{array}{l} \displaystyle\min_{\lambda \in \bigotimes_j \Delta_{n_j}} \max_{\alpha \in S_m, \gamma \in \Delta_{n,q^*}} \left\{ \sum \alpha_i - \frac{1}{2}\alpha^T \left(\sum_{j=1}^{n} \sum_{k=1}^{n_j} \frac{\lambda_{jk}\mathbf{Q}_{jk}}{\gamma_j} \right) \alpha \right\}, \\ \text{where } S_m = \left\{ \alpha \in \mathbb{R}^m \mid \displaystyle\sum_{i=1}^{m} \alpha_i y_i = 0, \ 0 \leqslant \alpha_i \leqslant C, \quad i = 1,\ldots,m \right\}. \end{array} \right.$$

The relation between the primal and dual variables is given by: $\gamma_j \frac{\mathbf{w}_{jk}}{\lambda_{jk}} = \sum_{i=1}^{m} \alpha_i y_i \phi_{jk}(\mathbf{x}_i)$. Note that (D) is only a partial dual (wrt. variables $\mathbf{w}, b, \xi$) of (P) and is not the joint dual. Interestingly the partial dual can be efficiently solved using a non-Euclidean gradient-descent based approach (see Section 3) and hence is explored here. In the following, we generalize this discussion using the functional framework and remove the restriction that the induced feature maps are finite dimensional.

2.1.1 THE FUNCTIONAL FRAMEWORK

We first consider the case $1 \leq q < \infty$. Let K_{jk} be positive kernel functions defined over the same input space $\mathcal{X}$. Each K_{jk} defines a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}_{jk}$ with the inner product $\langle .,. \rangle_{\mathcal{H}_{jk}}$. An element $h \in \mathcal{H}_{jk}$ has the norm $\|h\|_{\mathcal{H}_{jk}} = \sqrt{\langle h, h \rangle_{\mathcal{H}_{jk}}}$. Now for any λ_{jk} non-negative, define a new Hilbert space

$$\mathcal{H}_{jk}^{'} = \{h | h \in H_{jk}, \ \frac{\|h\|_{\mathcal{H}_{jk}}}{\lambda_{jk}} < \infty\}$$

with inner product as $\langle .,. \rangle_{\mathcal{H}_{jk}^{'}} = \frac{1}{\lambda_{jk}} \langle .,. \rangle_{\mathcal{H}_{jk}}$. We use the convention that if $\lambda_{jk} = 0$ then the only member of $\mathcal{H}_{jk}^{'}$ is $h = 0$. It is easy to see that $\mathcal{H}_{jk}^{'}$ is an RKHS with kernel as $\lambda_{jk}K_{jk}$ (see Rakotomamonjy et al., 2008). A direct sum of such RKHS, $\mathcal{H}_j = \bigoplus_k \mathcal{H}_{jk}^{'}$ is also an RKHS with the kernel as $K_j = \sum_k \lambda_{jk} K_{jk}$. Now again, for a given γ_j non-negative, consider Hilbert spaces $\mathcal{H}_j^{'}$ derived from $\mathcal{H}_j$ as follows: a) if $\gamma_j = 0$ then $\mathcal{H}_j^{'}$ contains only the zero element and if $\gamma_j > 0$ then elements in $\mathcal{H}_j^{'}$ as same as those in $\mathcal{H}_j$ however $\langle .,. \rangle_{\mathcal{H}_j^{'}} = \gamma_j \langle .,. \rangle_{\mathcal{H}_j}$. Again $\mathcal{H}_j^{'}$ are RKHS with kernels as $\frac{1}{\gamma_j} K_j = \frac{1}{\gamma_j} \sum_k \lambda_{jk} K_{jk}$ and their direct sum is in-turn an RKHS $\mathcal{H}$ with kernel as $K = \sum_{j=1}^{n} \frac{1}{\gamma_j} \sum_{k=1}^{n_j} \lambda_{jk} K_{jk}$. With this functional framework in mind we now let $\mathbf{w}_{jk}$ be an element of $\mathcal{H}_{jk}$ with the norm $\|\mathbf{w}_{jk}\|_{\mathcal{H}_{jk}} = \sqrt{\langle w_{jk}, w_{jk} \rangle_{\mathcal{H}_{jk}}}$ and let $\mathbf{w} \in \mathcal{H}$ where $\mathcal{H}$ is as defined above. The primal (P) in this case reads as follows:

[4] Only for the case $q = \infty$, we make an additional assumption that all the base kernels are strictly positive in order to write the dual in the form of problem (D) above.

$$\begin{aligned}&\min_{\xi_i, b, \mathbf{w}_{jk}\in\mathcal{H}_{jk}} \left\{ \max_{\gamma_j\in\Delta_{n,q^*}} \min_{\lambda_j\in\Delta_{n_j}} f(\mathbf{w},\lambda,\gamma,\xi) \right\} \\ &\text{s.t.} \quad y_i\left(\langle \mathbf{w}, \mathbf{x}_i\rangle_{\mathcal{H}} - b\right) \geqslant 1-\xi_i, \xi_i \geqslant 0,\end{aligned} \tag{7}$$

where $f(\mathbf{w},\lambda,\gamma,\xi) = \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n_j}\gamma_j \frac{\|\mathbf{w}_{jk}\|^2_{\mathcal{H}_{jk}}}{\lambda_{jk}} + C\sum_i \xi_i$.

Following the usual procedure for generalizing linear SVMs to RKHS via Representer theorem one obtains the following generalization of Theorem 2.3:

Theorem 2.4 *Let $\mathbf{Q}_{jk}$ be the $m \times m$ matrix*

$$\left(\mathbf{Q}_{jk}\right)_{ih} = y_h y_i K_{jk}(\mathbf{x}_i, \mathbf{x}_h) \qquad i,h = 1,\ldots,m.$$

The dual problem of (7) with respect to $\{\mathbf{w}, b, \xi\}$ is the following optimization problem:

$$\min_{\lambda_j\in\Delta_{n_j}} \underbrace{\max_{\alpha\in S_m,\gamma\in\Delta_{n,q^*}} \overbrace{\mathbf{1}^T\alpha - \frac{1}{2}\alpha^T\left(\sum_{j=1}^{n}\sum_{k=1}^{n_j}\frac{\lambda_{jk}Q_{jk}}{\gamma_j}\right)\alpha}^{f_\lambda(\alpha,\gamma)}}_{G(\lambda)}, \tag{D}$$

where $S_m = \left\{\alpha \in \mathbb{R}^m | 0 \leqslant \alpha \leqslant C, y^T\alpha = 0\right\}$.

We omit the proof as it is straightforward. To be noted that $\frac{\gamma_j}{\lambda_{jk}}\mathbf{w}_{jk}(.) = \sum_i \alpha_i y_i K_{jk}(., x_i)$ and all other conditions remain same.[5]

We will refer (D) as the dual problem. The dual (D) problem provides more insight into the formulation: λ_{jk} can be viewed as a weight given to the kernel K_{jk} and $\frac{1}{\gamma_j}$ can be thought of as an additional weight factor for the entire j^{th} group/descriptor. Since $\lambda_j \in \Delta_{n_j}$ (that is, λ_js are l_1 regularized), most of the λ_js will be zero at optimality and since $\gamma \in \Delta_{n,q^*}$, it amounts to combining kernels across descriptors in a non-trivial (and in case $q^* \geq 2$ in a non-sparse) fashion. Indeed, this is in-sync with findings of Nilsback and Zisserman (2006): kernels from different feature descriptors (components) are combined using non-trivial weights (that is, $\frac{1}{\gamma_j}$); moreover, only the "best" kernels from each feature descriptor (component) are employed by the model. This sparsity feature leads to better interpretability as well as computational benefits during the prediction stage. Note that in the case optimal weights (λ,γ) are known/fixed, then the problem is equivalent to solving an SVM with an effective kernel: $\mathbf{K}_{eff} \equiv \sum_{j=1}^{n}\left(\frac{\sum_{k=1}^{n_j}\lambda_{jk}\mathbf{K}_{jk}}{\gamma_j}\right)$. This observation leads to an efficient algorithm for solving the dual which is described in the subsequent section.

3. Algorithm for Solving the Dual Problem

This section presents the mirror descent based algorithm for efficiently solving the dual (D). A detailed proof of convergence of the algorithm is also presented. We begin by re-writing problem (D) as a minimization problem, rather than a minimax problem:

$$\min\{G(\lambda_1,\lambda_2,\ldots,\lambda_n) \mid \lambda_j \in \Delta_{n_j},\ j = 1,\ldots,n\}, \tag{8}$$

[5] Again, for the case $q = \infty$, we make the assumption that all base kernels are strictly positive in order that Theorem 2.4 is true.

where the objective function G is the optimal value function of the following problem:

$$G(\lambda_1,\lambda_2,\ldots,\lambda_n) = \max_{\gamma\in\Delta_{n,q^*},\alpha\in S_m} \underbrace{\left\{\mathbf{1}^T\alpha - \frac{1}{2}\alpha^T \sum_{j=1}^{n} \left(\frac{\sum_k \lambda_{jk}Q_{jk}}{\gamma_j}\right)\alpha\right\}}_{f_\lambda(\alpha,\gamma)}. \tag{9}$$

The function G is *convex* in $\lambda \in \mathbb{R}^n$ since it is the point-wise maximize of functions which are *linear* in λ. The minimization problem (8) is then that of minimizing a convex (possibly non-differentiable) function over a product of simplices. Problems with these features, even large-scale ones, can be solved efficiently by a Mirror Descent (MD) type algorithm (Ben-Tal et al., 2001; Beck and Teboulle, 2003) which is reviewed in the next subsection. An MD algorithm needs as input in each iteration a sub-gradient $G'(\lambda)$ belonging to the sub-gradient set $\partial G(\lambda)$. Using Danskin's theorem (see Bertsekas, 1999, prop. B.25), these elements are readily available from the solution of the *concave maximization problem* (in vector variables, γ and α) in (9).[6] A procedure for solving this maximization problem efficiently is presented in Section 3.3. Note that the maximum problem is solved numerically and hence the approximate sub-gradient is only obtained. Though we provide convergence analysis, it does not deal with the issue of approximate sub-gradient. Analysis of such situations is more involved and we postpone it to future work (see D'Aspermont, 2008).

3.1 Introduction to Mirror Descent

Consider the following problem.

$$\min f(x) \qquad x \in X, \tag{10}$$

where:

1. $X \subset \mathbb{R}^n$ is convex and closed with nonempty interior.

2. The objective function $f : X \to \mathbb{R}$ is a convex Lipschitz continuous function, with respect to a fixed given norm $\|\cdot\|$, that is:

$$\exists L, |f(x) - f(y)| \leqslant L\|x-y\| \quad \forall x,y \in \operatorname{int} X.$$

3. There exists an *oracle* which given $x \in X$ computes $f(x)$ and $f'(x) \in \partial f(x)$.

For such problems a classical algorithm is the Sub-gradient Projection Algorithm (SPA), which generates iteratively the sequence $\{x^t\}$ via:

$$x^{t+1} = \pi_X(x^t - s_t f'(x^t)),$$

where s_t is a step-size, and $\pi_X(y) = \underset{x\in X}{\operatorname{argmin}} \{\|x-y\|_2\}$ is the projection of y on X. The SPA can be rewritten equivalently as

$$x^{t+1} = \underset{x\in X}{\operatorname{argmin}} \left\{\langle x, s_t f'(x^t)\rangle + \frac{\|x-x^t\|_2^2}{2}\right\}.$$

[6] If α^*,γ^* represent the variables maximizing f for given λ, then the jk^{th} component of the sub-gradient $G'(\lambda)$ is $-\frac{1}{2}\frac{\alpha^{*\top}\mathbf{Q}_{jk}\alpha^*}{\gamma_j^*}$.

The main idea of Mirror Descent Algorithm(MDA) is to replace the distance function $\frac{1}{2}\|x-x^t\|_2^2$ based on the Euclidean norm by a general distance-like function $D(x,x^t)$ (also referred to as prox-function). The basic iteration step then becomes

$$x^{t+1} = \operatorname*{argmin}_{x\in X} \left\{ \langle x, s_t f'(x^t)\rangle + D(x,x^t) \right\}. \tag{11}$$

With the freedom to choose D one can adapt it to the specific constraint set X. The minimal requirements on the “distance function” are

1. D is nonnegative,
2. $D(u,v)=0$ if and only if $u=v$.

A possible way to construct such a distance-like function is as follows: Let $\Phi : X \to \mathbb{R}$ be strongly convex with parameter $\sigma > 0$ with respect to a norm $\| \; \|$, that is:

$$\langle \nabla\Phi(x) - \nabla\Phi(y), x-y\rangle \geq \sigma\|x-y\|^2, \quad \forall x,y \in X\,.$$

Then

$$B_\Phi(x,y) = \Phi(x) - \Phi(y) - \langle x-y, \nabla\Phi(y)\rangle$$

is a distance-like function (often called Bregman Divergences). With this choice, the iteration scheme (11) is equivalent (see Beck and Teboulle, 2003) to the following three step procedure

$$\begin{array}{ll} 1. & x^t \leftarrow \nabla\Phi^*(y^t), \\ 2. & y^{t+1} \leftarrow \nabla\Phi(x^t) - s_t f'(x^t), \\ 3. & x^{t+1} \leftarrow \nabla\Phi^*(y^{t+1}) = \nabla\Phi^*(\nabla\Phi(x^t) - s_t f'(x^t)). \end{array} \tag{12}$$

Here $\Phi^*(y) = \max\limits_{x\in X}\{\langle x,y\rangle - \Phi(x)\}$ is the *conjugate* function of Φ.

This procedure yields efficient convergent algorithms for solving (10). More formally we state the following theorem proved in Beck and Teboulle (2003)

Theorem 3.1 *Let $\{x^t\}$ be the sequence generated from a starting point $x^1 \in \operatorname{int} X$ by the MD procedure outlined in (12) with the D being the Bregman Divergence $B_\phi(\cdot,\cdot)$. Let $f^* = \min_{x\in X} f(x)$, and let $x^* \in X$ be a point where the minimum is attained. Then for every $t \geqslant 1$*

1. $$\min_{1\leqslant\tilde{t}\leqslant t} f(x^{\tilde{t}}) - f^* \leqslant \frac{B_\Phi(x^*,x^1) + 2\sigma^{-1}\sum_{\tilde{t}=1}^t s_{\tilde{t}}^2 \|f'(x^{\tilde{t}})\|_*^2}{\sum_{\tilde{t}=1}^t s_{\tilde{t}}},$$

   where σ is the strong-convexity parameter of Φ.

2. *In particular if the step size sequence $\{s_t\}$ satisfies*

   $$\sum_{\tilde{t}=1}^{t} s_{\tilde{t}} \to \infty, s_t \to 0, t \to \infty,$$

   then the method converges, that is:

   $$t \to \infty \Rightarrow \min_{1\leq\tilde{t}\leq t} f(x^{\tilde{t}}) - f^* \to 0,$$

3. *Moreover if the step-sizes s_t are chosen as*

$$s_t = \sqrt{\frac{2\sigma\Gamma(x^1)}{L_f^2 t}},$$

then the following efficiency estimate holds

$$\min_{1\leqslant\tilde{t}\leqslant t} f(x^{\tilde{t}}) - f^* \leqslant L_f \sqrt{\frac{2\Gamma(x^1)}{\sigma t}},$$

where $\Gamma(x^1) = \max_{x\in X} B_\Phi(x,x^1)$ measures the "width" of the feasible set X. □

The above theorem shows that MD procedures require $O(\frac{L_f^2\Gamma(x^1)}{\sigma\varepsilon^2})$ iterations for attaining an ε accurate solution where each iteration is very cheap, requiring just a gradient computation.

3.2 Minimizing G by Mirror Descent Procedures

In the following we discuss the suitability of MD procedures outlined in (12) for minimizing G given in (9).

For an MD procedure to apply we first need to demonstrate that G is convex and Lipschitz continuous. We also need to devise a Distance generating function which is suitable for a feasible set comprised of a product of simplices. We begin with the proposition

Proposition 3.1 *If there exists scalars $0 < \tau < 1$, $\mu > 0$ such that all eigenvalues of each $\mathbf{Q}_{jk}$ matrix lie within an interval $(\tau\mu, \mu)$, then the function G given by*

$$G(\lambda_1,\cdots,\lambda_n) = \max_{\alpha\in S_m,\gamma\in\Delta_{n,q^*}} \mathbf{1}^T\alpha - \frac{1}{2}\alpha^T \left[\sum_{j=1}^{n}\left(\frac{\sum_{k=1}^{n_j}\lambda_{jk}\mathbf{Q}_{jk}}{\gamma_j}\right)\right]\alpha$$

is convex and Lipschitz continuous w.r.t. in the l_1 norm for any $q \geq 1$.

Proof See Appendix for a proof. ■

A suitable Distance generating function of the form B_Φ over product of simplices is given in the following

Proposition 3.2 *Let*

$$\Phi_j(\lambda_j) = \sum_{k=1}^{n_j}\lambda_{jk}\ln(\lambda_{jk}),\ \lambda_j \in \Delta_j\ \forall j = 1,\ldots,n.$$

The function $\Phi(\lambda) = \sum_{j=1}^{n}\Phi_j(\lambda_j) = \sum_{j=1}^{n}\sum_{k=1}^{n_j}\lambda_{jk}\ln(\lambda_{jk})$ is strongly convex with parameter $\frac{1}{n}$ with respect to the l_1 norm. The corresponding distance generating function is given by

$$B_\Phi(\lambda^*,\lambda^1) = \sum_{j=1}^{n}\sum_{k=1}^{n_j}\lambda_{jk}^*\ln\left(\frac{\lambda_{jk}^*}{\lambda_{jk}^1}\right).$$

Proof The function Φ_j is convex in λ_j as its Hessian is positive definite over the interior of its domain. Since Φ is a sum of such functions it is also convex.

Recall that a necessary and sufficient condition (Rockafellar, 1970) for a convex function Φ to be strongly convex with respect to a norm, $\|.\|$, and parameter σ, is

$$\langle \nabla\Phi(\lambda) - \nabla\Phi(\lambda^*), \lambda - \lambda^* \rangle \geq \sigma \|\lambda - \lambda^*\|^2,$$

where $\nabla\Phi(\lambda)$ is an element in the sub-gradient set of Φ evaluated at λ.

The proof can now be constructed as follows

$$\begin{aligned} \langle \nabla\Phi(\lambda) - \nabla\Phi(\lambda^*), \lambda - \lambda^* \rangle &= \sum_{j=1}^{n} KL(\lambda_j, \lambda_j^*) \\ &\geq \sum_{j=1}^{n} \|\lambda_j - \lambda_j^*\|_1^2 \\ &\geq \frac{1}{n} \|\lambda - \lambda^*\|_1^2, \end{aligned}$$

wherein the first equality $KL(p,q) = \sum_i p_i \log \frac{p_i}{q_i}$. The first inequality is obtained by noting that $KL(p,q) \geq \|p-q\|_1^2$ (see Cover and Thomas, 2006). The second inequality is valid since for any nonnegative a_j one has by Cauchy-Schwartz inequality, $\frac{1}{n}(\sum_{j=1}^{n} a_j)^2 \leq \sum_{j=1}^{n} a_j^2$. This proves that Φ is strongly convex with parameter $\frac{1}{n}$ in the l_1 norm.

Finally, the function B_Φ can be written as

$$B_\Phi(\lambda^*, \lambda) = \Phi(\lambda^*) - \Phi(\lambda) - \langle \nabla\Phi(\lambda), \lambda^* - \lambda \rangle.$$

Hence, it is indeed a Bregman-type distance generating function ■

3.2.1 THE CHOICE OF STEP-SIZE

By Theorem 3.1 the choice of step-size is guided by the term $\Gamma(\lambda^1)$, where λ^1 is in the interior of the product of simplices. If one chooses $\lambda_{jk}^1 = \frac{1}{n_j}$ then one can obtain an estimate of $\Gamma(\lambda^1)$ as follows:

$$B_\Phi(\lambda^*, \lambda^1) \leq \sum_{j=1}^{n} \log n_j \leq n \log n_{\max} \text{ where } n_{\max} = \max_j n_j.$$

The first inequality follows from the fact that $\sum_k \lambda_{jk} \log \lambda_{jk} \leq 0, \forall \lambda \in \bigotimes_j \Delta_{n_j}$ and the second inequality follows from the definition of $n_{\max}$. This upper bound immediately yields $\Gamma(\lambda^1) \leqslant n \log n_{\max}$. The candidate step-size (refer Theorem 3.1) now writes as

$$s_t = \frac{\sqrt{2\frac{1}{n} n \log n_{\max}}}{L_G} \frac{1}{\sqrt{t}} = \frac{\sqrt{2 \log n_{\max}}}{L_G} \frac{1}{\sqrt{t}},$$

where L_G is the Lipschitz constant of G. However this step-size estimate is impractical as L_G will not be known a priori. A more pragmatic choice could be

$$s_t = A\sqrt{\Gamma(\lambda^1)\sigma} \frac{1}{\|\nabla_\lambda G(\lambda^t)\|_\infty} \frac{1}{\sqrt{t}} = A\sqrt{\log n_{\max}} \frac{1}{\|\nabla_\lambda G(\lambda^t)\|_\infty} \frac{1}{\sqrt{t}},$$

where A is a constant. It can be shown (Ben-Tal et al., 2001) that even for this step-size an efficiency estimate, similar to the one given in Theorem 3.1, is valid.

3.2.2 A SKETCH OF MD-BASED ALGORITHM

We are now ready to state an MD procedure for computing G. Given a sub-gradient by the oracle, and a suitably chosen step-size, one needs to compute a projection (step 3 in (12)) to complete one iteration of MD. Owing to the clever choice of prox-function, the projection step in our case is very easy to calculate and has an analytical expression given by:

$$\nabla\Phi(\lambda)_{jk} = \big(\ln(\lambda_{jk}) + 1\big)\,,$$

$$\nabla\Phi^*(\tilde{\lambda})_{jk} = \left(\frac{e^{\tilde{\lambda}_{jk}}}{\sum\limits_{l=1}^{n_j} e^{\tilde{\lambda}_{jl}}} \right).$$

The final MD procedure for minimizing G now reads:

```
Algorithm 1:
  Require: $\lambda^1 \in \left\{ \bigotimes_{1 \leqslant j \leqslant n} \Delta_{n_j} \right\}$
    repeat
      $(\alpha^*, \gamma^*) \leftarrow \text{argmax}_{\alpha \in S_m, \gamma \in \Delta_{n,q^*}} f_{\lambda^t}(\alpha, \gamma)$    (Oracle computation)
      $\tilde{\lambda}_{jk}^{t+1} \leftarrow (\nabla\Phi(\lambda^t) - s_t G'(\lambda))_{jk} = \Big(\ln(\lambda_{jk}^t) + 1\Big) + s_t \alpha^{*T} \frac{Q_{jk}}{\gamma_j^*} \alpha^*$    (Descent Direction)
      $\lambda_{jk}^{t+1} \leftarrow \nabla\Phi^*\left(\tilde{\lambda}^{t+1}\right) = \left( e^{\tilde{\lambda}_{jk}^{t+1}} / \sum_{k=1}^{n_j} e^{\tilde{\lambda}_{jk}^{t+1}} \right)$    (Projection step)
    until convergence
```

By virtue of Theorem 3.1 (and using bound on Lipschitz constant derived in Appendix) this algorithm obtains an ε accurate minimizer of G in $O(n^{2+\frac{2}{q^*}} \log n_{\max}/\varepsilon^2)$ steps. Note that in practice the number of groups n (intuitively, the number of feature descriptors) is never high (typically < 10) and infact one can assume it to be $O(1)$; in which case the number of iterations will be nearly-independent of the number of kernels! The cost of each iteration depends on how efficiently one can maximize $f_\lambda(\alpha, \gamma)$ as a function of α, γ for a fixed λ. Note that gradient computation (that is, maximizing f) is the predominant computation in the mirror-descent based algorithm as the projection and step-size can be computed very easily from the analytical expressions presented above. On passing, we also note that there exist efficient projection algorithms for l_1-l_∞ regularization (Quattoni et al., 2009). In the next section we show that maximizing f can be achieved by solving a series of SVMs.

Again note that in the special case $n = 1$, where $\mathbf{VSKL}_q$ (for any q) is equivalent to **MKL**, maximizing f is nothing but solving an SVM problem (with effective kernel computed with current weights). Since the per-step computation, in this special case, is predominantly that of solving an SVM (the projection and step-size computations are negligible) and the number of iterations is nearly-independent of the number of kernels, the proposed MD based algorithm is expected to perform far better than traditional reduced (projected) gradient based MKL solvers like `simpleMKL`. Also, in this case, the no. iterations is $O\left(\log n_{\max}/\varepsilon^2\right)$ and $n_{\max} = n_{\text{tot}}$ where n_{tot} is the total number of kernels. Cost of computing the effective kernel at each step depends on the sparsity of λ_j

however a conservative estimate gives $O(m^2 n_{\text{tot}})$ and projection, step size computations are $O(n_{\text{tot}})$ (negligible). Assuming the SVM problem can be solved in $O(m^2)$ time, we have the following complexity bound in case $n = 1$: $O\left(m^2 n_{\text{tot}} \log n_{\text{tot}}/\varepsilon^2\right)$. Also, in the case $q = 1$, the optimal value of γ_j is 1 for all j and hence maximizing f again corresponding to solving an SVM with effective kernel as canonical (equal-weight) sum of all the active kernels in each group. Again, in this case, the overall complexity is $O\left(m^2 n_{\text{tot}} \log n_{\max}/\varepsilon^2\right)$. The next section presents an efficient iterative scheme for maximizing f in a general case (that is, $n > 1, q > 1$).

3.3 Computing the Oracle

The joint maximization in (α,γ) of f_λ in the case $q = \infty$ can be posed as a Quadratically Constrained Quadratic Program (QCQP):

$$\max_{\alpha\in S_m,\gamma\in\Delta_n} f_\lambda(\gamma,\alpha) = \mathbf{1}^T\alpha - \frac{1}{2}\alpha^T\left[\sum_{j=1}^{n}\left(\frac{\sum_{k=1}^{n_j}\lambda_{jk}\mathbf{Q}_{jk}}{\gamma_j}\right)\right]\alpha$$

$$\left.\begin{aligned} = \quad & \max_{\alpha\in S_m,\gamma\in\Delta_n,v} \mathbf{1}^T\alpha - \sum_{j=1}^{n} v_j \\ \text{s.t.} \quad & 2\gamma_j v_j \geqslant \alpha^T\left[\sum_{k=1}^{n_j}\lambda_{jk}\mathbf{Q}_{jk}\right]\alpha \ \ \forall j \end{aligned}\right\}. \tag{13}$$

Using the identity

$$2\gamma_j v_j = \frac{1}{2}(\gamma_j + v_j)^2 - \frac{1}{2}(\gamma_j - v_j)^2,$$

the constraint in problem (13) becomes

$$\alpha^T\left[\sum_k \lambda_{jk} Q_{jk}\right]\alpha + \frac{1}{2}(\gamma_j - v_j)^2 \leq \frac{1}{2}(\gamma_j + v_j)^2,$$

and consequently problem (13) is a *conic quadratic* (CQ) *problem*.

A CQ problem can be solved with efficient solvers like $^{\text{TM}}$Mosek. However for an arbitrary norm, $q > 1$, such a formulation may not be possible and, even for $q = \infty$, very large-scale problems may require a more efficient algorithm. To this end we consider leveraging SVM solvers. Note that for each fixed value of γ one needs to solve an SVM problem in α. Moreover there exist *closed form solutions* when f is maximized over γ for fixed α. Such a Block Coordinate Ascent (BCA) (Tseng, 2001) procedure in general may not lead to convergence, but for the problem at hand we will show that the algorithm does indeed converge to a global maximum.

3.3.1 BLOCK COORDINATE ASCENT

In this section we describe a convergent and efficient algorithm based on the Block Coordinate Ascent (BCA) method. As a consequence of Lemma 2.2 the following is true

Proposition 3.3 *For a fixed λ,α the problem*

$$max_{\gamma\in\Delta_{n,q^*}} f_\lambda(\alpha,\gamma)$$

is optimized at

$$\gamma_i = \frac{D_i^{\frac{1}{q^*+1}}}{\left(\sum_{j=1}^n D_j^{\frac{q^*}{q^*+1}}\right)^{\frac{1}{q^*}}} \; i = 1,\ldots,n.$$

If $q = 1$ (that is, $q^* = \infty$), optimality is achieved at $\gamma_i = 1$ iff $D_i > 0$ where $D_j = \sum_{k=1}^{n_j} \lambda_{jk}\alpha^\top Q_{jk}\alpha$.

Proof Recall that

$$\max_{\alpha\in S_m,\gamma\in\Delta_{n,q^*}} f_\lambda(\alpha,\gamma) = \max_{\alpha\in S_m} \alpha^\top e - \frac{1}{2}\min_{\gamma\in\Delta_{n,q^*}} \sum_{j=1}^n \frac{D_j}{\gamma_j},$$

where $D_j = \sum_{k=1}^{n_j} \lambda_{jk}\alpha^\top Q_{jk}\alpha$. For a fixed α, the optimal γ is obtained by

$$\min_{\gamma\in\Delta_{n,q^*}} \sum_{j=1}^n \frac{D_j}{\gamma_j}.$$

The claim follows from Lemma 2.2. ■

This Proposition shows that one can use analytical expressions for γ when maximizing f_λ for a fixed α. Alternatively for a fixed γ, maximizing f_λ is equivalent to solving an SVM. These observations motivate the following algorithm for Oracle computation:

Algorithm 2:

```
Require: $\gamma^1 \in \Delta_{n,q^*}$
  repeat
    Compute $\alpha^{k+1} = \underset{\alpha\in S_m}{\operatorname{argmax}} \{f_\lambda(\alpha,\gamma^k)\}$ using SVM solver
    Compute $\gamma^{k+1} = \underset{\gamma\in\Delta_{n,q^*}}{\operatorname{argmax}} \{f_\lambda(\alpha^{k+1},\gamma)\}$ by Proposition 3.3
  until convergence
```

In the following subsection we establish the convergence of this algorithm.

3.3.2 CONVERGENCE OF BCA ALGORITHM

We begin by introducing some propositions.

Definition 3.1 *We say that $z = (\alpha,\gamma)$ is a strict coordinate-wise maximum point of f over $A\times\Gamma$ if $z \in A\times\Gamma$ and*

$$\begin{aligned} f(\alpha',\gamma) &< f(z) \qquad \forall\alpha' \in A,\\ f(\alpha,\gamma') &< f(z) \qquad \forall\gamma' \in \Gamma. \end{aligned}$$

Lemma 3.2 *Assume that A and Γ are convex sets, and f is a continuously differentiable function over $A\times\Gamma$. If z is a strict coordinate-wise maximum point of f over $A\times\Gamma$, then z is a local maximum point of f over $A\times\Gamma$.*

Proof Let $\alpha' \in A$, then $\forall u \in [0,1], u\alpha + (1-u)\alpha' \in A$ since A is convex. Let us consider $g(u) = f((1-u)\alpha + u\alpha',\gamma)$. g is differentiable and, since z is a strict coordinate-wise maximum point of f over $A\times\Gamma$, then $\forall u \in (0,1], g(0) > g(u)$, and this implies that $g'(0) < 0$, that is:

$$g'(0) = (\alpha'-\alpha)^T \nabla_\alpha f(\alpha,\gamma) < 0 \quad \forall\alpha' \in A,\ \alpha' \neq \alpha.$$

Following the same reasoning for γ, the following statement holds

$$\begin{aligned} &\nabla_\alpha f(\alpha,\gamma)^T(\alpha'-\alpha) < 0 \quad \forall\alpha' \in A,\ \alpha' \neq \alpha, \\ &\nabla_\gamma f(\alpha,\gamma)^T(\gamma'-\gamma) < 0 \quad \forall\gamma' \in \Gamma,\ \gamma' \neq \gamma. \end{aligned} \tag{14}$$

Now, by Taylor expansion,

$$f(\alpha',\gamma') = f(\alpha,\gamma) + \nabla_\alpha f(\alpha,\gamma)^T(\alpha'-\alpha) + \nabla_\gamma f(\alpha,\gamma)^T(\gamma'-\gamma) + O\left(\|\alpha-\alpha'\| + \|\gamma-\gamma'\|\right).$$

Using (14) we see that if (α',γ') is close enough to (α,γ), then $f(\alpha',\gamma') < f(\alpha,\gamma)$. ■

Proposition 3.4 *The BCA procedure (alg. 2) when applied to $f_\lambda(\alpha,\gamma)$ with respect to the blocks α and γ converge to a coordinate-wise maximum point of f_λ.*

Proof We begin by arguing that f_λ is bounded when Q_{jk} are p.d in the interior of simplex defined by γ, that is, $\gamma_j > 0$. Recall that at optimality, γ always lie in the interior for any $q > 1$. Hence for $q > 1$ we can as well restrict our search space to the interior of the simplex. For all such γ we have

$$f_\lambda(\alpha,\gamma) \leq \sum_{i=1}^{m}(\alpha_i - \frac{\tilde{\mu}}{2}\alpha_i^2),$$

where $\tilde{\mu} = \mu(\sum_{j=1}^{n}\gamma_j^{-1})$ and $\mu > 0$ is the greatest lower bound over all minimal eigenvalues of $Q_{j,k}$ matrices. For $q = 1$ case one can apply the above upper bound with $\gamma_i = 1$. Next, consider the following result.

Lemma 3.3 *f_λ is hemivariate over $S_m \times \Delta_n$.*

Proof Recall that a function f_λ is called hemivariate if it is not constant on any line segment of $S_m \times \Delta_n$. We proceed by contradiction. Let us assume that there exist $(\tilde{\alpha}^1,\tilde{\gamma}^1) \in S_m \times \Delta_n$ and $(\tilde{\alpha}^2,\tilde{\gamma}^2) \in S_m \times \Delta_n$ such that $\forall t \in [0,1]$, the following hold

$$g(t) = f_\lambda(t\tilde{\alpha}^1 + (1-t)\tilde{\alpha}^2, t\tilde{\gamma}^1 + (1-t)\tilde{\gamma}^2) = \text{ a constant.}$$

Then, $\forall t \in (0,1)$

$$\dot{g}(t) \equiv \frac{dg}{dt} = B_0 + \sum_j \frac{B_j}{(t + \frac{\tilde{\gamma}_j^1}{\tilde{\gamma}_j^2 - \tilde{\gamma}_j^1})^2} = 0, \tag{15}$$

where

$$B_j = \frac{1}{\tilde{\gamma}_j^2 - \tilde{\gamma}_j^1{}^3}\left[\tilde{\gamma}_j^2\tilde{\alpha}^1 - \tilde{\gamma}_j^1\tilde{\alpha}^2\right]^T Q_j \left[\tilde{\gamma}_j^2\tilde{\alpha}^1 - \tilde{\gamma}_j^1\tilde{\alpha}^2\right],$$

$$Q_j = \sum_{k=1}^{n_j}\lambda_{jk}\mathbf{Q}_{jk},$$

and

$$B_0 = e^\top(\tilde{\alpha}^2 - \tilde{\alpha}^1) - \frac{1}{2}(\tilde{\alpha}^1 - \tilde{\alpha}^2)^\top \sum_{j=1}^{n}\frac{Q_j}{\tilde{\gamma}_j^2 - \tilde{\gamma}_j^1}(\tilde{\alpha}^1 - \tilde{\alpha}^2).$$

$\dot{g}(t)$ is a rational function of t and is 0 on $(0,1)$. This is possible if and only if $B_0 = 0$ and $\sum_j \frac{B_j}{(t+\frac{\tilde{\gamma}_j^1}{\tilde{\gamma}_j^2-\tilde{\gamma}_j^1})^2} = 0$. To establish this recall that the higher order derivatives of g are also 0. This leads in particular to:

$$\sum_j \frac{B_j}{(t+\frac{\tilde{\gamma}_j^1}{\tilde{\gamma}_j^2-\tilde{\gamma}_j^1})^3} = 0.$$

Let us now consider the sets $\Theta = \left\{ s \in \mathbb{R} \middle| \exists j, \frac{\tilde{\gamma}_j^1}{\tilde{\gamma}_j^2-\tilde{\gamma}_j^1} = s \right\}$ and $\Omega_s = \left\{ j \in \mathbb{N} \middle| \frac{\tilde{\gamma}_j^1}{\tilde{\gamma}_j^2-\tilde{\gamma}_j^1} = s \right\}$. We have

$$\sum_j \frac{B_j}{(t+\frac{\tilde{\gamma}_j^1}{\tilde{\gamma}_j^2-\tilde{\gamma}_j^1})^3} = \sum_{s\in\Theta} \sum_{j\in\Omega_s} \frac{B_j}{(t+s)^3}.$$

The family of $\{(t+s)^3\}, s \in \mathbb{R}$ is linearly independent, then, $\forall s \in \Theta, \sum_{j\in\Omega_s} \frac{B_j}{(t+s)^2} = 0$ by (15), and since $s = \frac{\tilde{\gamma}_j^1}{\tilde{\gamma}_j^2-\tilde{\gamma}_j^1}$ and $\forall j, \tilde{\gamma}_j^1 > 0$, then, $\mathrm{sign}(\tilde{\gamma}_j^2 - \tilde{\gamma}_j^1)$ is constant over $\{j \in \Omega_s\}$. We know that Q_j is positive definite, thus, $\mathrm{sign} B_j$ is constant over $\{j \in \Omega_s\}$. This implies that $\forall j, B_j = 0$. The positiveness of Q_j implies that this is possible only if $\forall j, \tilde{\gamma}_j^2 \tilde{\alpha}^1 - \tilde{\gamma}_j^1 \tilde{\alpha}^2 = 0$, which is equivalent to $\forall (j,i), (\tilde{\gamma}_j^2 \tilde{\alpha}_i^1)^{q^*} = (\tilde{\gamma}_j^1 \tilde{\alpha}_i^2)^{q^*}$ and summing over j, $\tilde{\gamma}^2$ and $\tilde{\gamma}^2$ belonging to Δ_{n,q^*}, we obtain $\tilde{\alpha}_1 = \tilde{\alpha}_2$ and then $\tilde{\gamma}^1 = \tilde{\gamma}^2$. Hence, f_λ is hemivariate and, this proves as well that f_λ is strictly concave. ■

We continue now the proof of Proposition 3.4. Let us consider a sequence z^p such that $z^{2p} = (\alpha^{p+1}, \gamma^p)$ and $z^{2p+1} = (\alpha^{p+1}, \gamma^{p+1})$. Since, by definition of our algorithm, $f_\lambda(z^{p+1}) \geqslant f_\lambda(z^p)$, and f_λ is bounded over $S_m \times \Delta_n$, then $f_\lambda(z^p)$ converges. Moreover, $S_m \times \Delta_n$ is compact in $\mathbb{R}^{m+n}$, so by passing to a subsequence if necessary, we can assume that $z^{2\phi(p)+1}$ converges to some z_1. Next we show that z^{2p+1} has a unique cluster point.

First we show that if $z^{2\phi(p)}$ converges to a cluster point z_1 of z^p, so does $z^{2\phi(p)+1}$. Indeed, if not, then $z^{2\phi(p)+1}$ has another cluster point than say $z_2 \neq z_1$).Therefore, we can assume that $\exists \tilde{\phi}$, a subsequence of $\phi(p)$ such that $z^{2\tilde{\phi}(p)+1}$ converges to z_2. Since $f_\lambda(z^p)$ converges, we have

$$\lim_{p\to\infty} f_\lambda(z^{2\tilde{\phi}(p)}) = \lim_{p\to\infty} f_\lambda(z^{2\tilde{\phi}(p)}).$$

Fix any $u \in [0,1]$ and denote $\tilde{z}^p = z^{2\tilde{\phi}(p)} + u(z^{2\tilde{\phi}(p)+1} - z^{2\tilde{\phi}(p)})$. We notice that $\tilde{z}^p \in S_m \times \Delta_n$. It is obvious that $\tilde{z}^p$ converges to $(1-u)z_1 + uz_2$. Since, f_λ is jointly concave with regard to (α,γ), we have

$$f_\lambda(\tilde{z}^p) \geqslant (1-u)\, f_\lambda(z^{2\tilde{\phi}(p)}) + u\, f_\lambda(z^{2\tilde{\phi}(p)+1}),$$

and by passing to the limit,

$$f_\lambda(\tilde{z}) \geqslant (1-u)\, f_\lambda(z_1) + u\, f_\lambda(z_2).$$

We cannot have $\forall \lambda \in [0,1], f_\lambda(\tilde{z}) = (1-u)\, f_\lambda(z_1) + u\, f_\lambda(z_2)$ because f_λ is hemivariate. Hence,

$$\exists \lambda \mid f_\lambda(\tilde{z}) > (1-u)\, f_\lambda(z_1) + u\, f_\lambda(z_2). \tag{16}$$

Since: $f(z^{2\tilde{\phi}(p)+1}) = \max_{\gamma\in\Delta_{n,q}} \{ f(\alpha^{\tilde{\phi}(p)+1}, \gamma, \lambda) \}$, the following statement holds:

$$\forall \gamma \in \Delta_n, f_\lambda(z^{2\tilde{\phi}(p)+1}) \geqslant f_\lambda(\alpha^{\tilde{\phi}(p)+1}, \gamma),$$

and since $z^{2\tilde{\phi}(p)}$,$\tilde{z}^p$ and $z^{2\tilde{\phi}(p)+1}$ differ only in their second coordinate block γ, we have $f_\lambda(\tilde{z}^p) \leqslant (1-u)f_\lambda(z^{2\tilde{\phi}(p)}) + u f_\lambda(z^{2\tilde{\phi}(p)+1})$, and by passing to the limit, $f_\lambda(\tilde{z}) \leqslant (1-u)f_\lambda(z_1) + u f_\lambda(z_2)$ which contradicts (16). Hence, $z_1 = z_2$. We showed that $z^{2\phi(p)+1}$ has a unique cluster point z_1, hence it converges to z_1. We next prove that z_1 is a coordinate-wise maximum point of f_λ. Recall that

$$\forall \gamma \in \Delta_n, f_\lambda(x^{2\tilde{\phi}(p)+1}) \geqslant f_\lambda(\alpha^{\tilde{\phi}(p)+1}, \gamma).$$

Passing to the limit, we have:

$$\forall \gamma \in \Delta_n, f_\lambda(z_1) \geqslant f_\lambda(\alpha(z_1), \gamma), \tag{17}$$

where $\alpha(z_1) = \alpha^{\tilde{\phi}(\infty)+1}$. The same reasoning with regard to α shows that

$$\forall \alpha \in S_m, f_\lambda(x_1) \geqslant f_\lambda(\alpha, \gamma(x_1)), \tag{18}$$

where $\gamma(z_1) = \gamma^{\tilde{\phi}(\infty)+1}$. This shows that z_1 is a coordinate-wise maximum point of f_λ and, according to (3.2), z_1 is a local maximum of f_λ over $S_m \times \Delta_n$ and since f_λ is strictly concave outside the line where $\alpha^1\gamma^2 = \alpha^2\gamma^1$, and since f_λ is not constant on any of these lines, z_1 is the unique global maximum of f_λl; hence strict inequalities hold in (17) and (18). ■

Now that the mirror-descent as well as the block coordinate ascent procedures are presented and the respective convergences are proved, we now proceed to present the overall algorithm for solving the dual (D).

3.4 The mirrorVSKL Procedure

This section presents the mirrorVSKL algorithm for solving the dual (D):

Algorithm 3: mirrorVSKL

```
**Require:** $\lambda^1 \in \left\{ \bigotimes_{1 \leqslant j \leqslant n} \Delta_{n_j} \right\}$
  **repeat**
    $(\alpha^*, \gamma^*) \leftarrow \underset{\alpha \in S_m, \gamma \in \Delta_n}{\text{argmax}} \; f(\alpha, \gamma, \lambda^t)$      (Use BCA in Alg. 2)
    $\tilde{\lambda}_{jk}^{t+1} \leftarrow (\nabla\Phi(\lambda^t) - s_t G'(\lambda))_{jk} = \left(\ln(\lambda_{jk}^t) + 1\right) + s_t \alpha^{*T} \frac{Q_{jk}}{\gamma_j^*} \alpha^*$    (Descent Direction)
    $\lambda_{jk}^{t+1} \leftarrow \nabla\Phi^* \left(\tilde{\lambda}^{t+1}\right) = \left( e^{\tilde{\lambda}_{jk}^{t+1}} / \sum_{k=1}^{n_j} e^{\tilde{\lambda}_{jk}^{t+1}} \right)$      (Projection step)
  **until** convergence
```

The algorithm converges to the optimal of (D) for arbitrary $q \geq 1$. The per-step complexity in the mirror-descent iterations now depends on the number of iterations of the BCA algorithm. However it was observed in practice (see Section 4) that for the values of n encountered, the BCA converges in 2-4 iterations and hence can be assumed to be a constant. With this assumption, even in the general case ($n > 1, q > 1$), the computational complexity of mirrorVSKL remains to be $O\left(m^2 n_{\text{tot}} \log n_{\max}/\varepsilon^2\right)$. We conclude this section with the following note: convergence of the mirror descent algorithm is based on the fact that sub-gradients are exactly computable. However in mirrorVSKL, the sub-gradients are computed using an oracle numerically and hence is approximate. Convergence analysis with such approximate sub-gradients is non-trivial and a research problem in itself. The work by D'Aspermont (2008) is a good starting point for this.

4. Numerical Experiments

This section presents results of simulations which prove the suitability of employing the proposed VSKL formulations for multi-modal tasks like object categorization. Experimental results which demonstrate the scalability of the `mirrorVSKL` algorithm in solving the traditional block l_1 regularization based MKL formulation are also presented.

4.1 Performance on Object Categorization Data Sets

The experimental results summarized in this section aim at proving the suitability of employing the proposed VSKL formulations for tasks like object categorization. The following benchmark data sets were used in our experiments:

Caltech-101 (Fei-Fei et al., 2004) Collection of 9144 images[7] from 102 categories of objects like faces, watches, ants etc. The minimum, average and maximum number of images per category are 31, 90, 800 respectively.

Caltech-256 (Griffin et al., 2007) Collection of 30607 images[8] from 257 categories of objects. The minimum, average and maximum number of images per category are 80, 119, 827 respectively.

Oxford flowers (Nilsback and Zisserman, 2006) Collection of images of 17 varieties of flowers.[9] The number of images per category is 80.

Following the strategy of Vedaldi et al. (2009), the following four feature descriptors[10] were employed in the case of the Caltech data sets:

1. Geometric blur (Zhang et al., 2006; Berg et al., 2005). These descriptors are initially computed at representative points of the image. Later, the distance between two images is obtained as the average distance of nearest descriptor pairs.

2. PHOW gray/color (Lazebnik et al., 2006). SIFT features are computed densely on a regular grid and quantized in 300 visual words. Spatial histogram with 4 × 4 subdivisions are then formed. The color variant concatenates SIFT descriptors computed on the HSV channels.

3. Self-similarity (Shechtman and Irani, 2007). Similar to the PHOW features, descriptors are quantized in 300 visual words, and a spatial histogram of size 4 × 4.

In case of the Oxford flowers data set, the seven feature descriptors employed in Nilsback and Zisserman (2006, 2008) are used here.[11]

Each feature descriptor mentioned above, describes the image in terms of few feature values. As mentioned previously, it was observed in the literature (see Nilsback and Zisserman, 2006) that employing feature values obtained from various descriptors simultaneously is beneficial for object

[7] Available at `http://www.vision.caltech.edu/Image_Datasets/Caltech101`.

[8] Available at `http://www.vision.caltech.edu/Image_Datasets/Caltech256`.

[9] Available at `http://www.robots.ox.ac.uk/~vgg/data/flowers/17/17flowers.tgz`.

[10] Software available at `http://www.robots.ox.ac.uk/~vgg/software/MKL/v1.0/index.html`.

[11] Corresponding distance matrices are available at `http://www.robots.ox.ac.uk/~vgg/data/flowers/17/index.html`.

categorization; however not all of the features obtained using a feature descriptor may be useful. The state-of-the-art performance on these data sets is achieved by a methodology which generates kernels using each of the feature descriptors and then chooses the best among them using the framework of MKL (Varma and Ray, 2007; Nilsback and Zisserman, 2008). The MKL formulation employed in Varma and Ray (2007) Nilsback and Zisserman (2008) is equivalent to the traditional block l_1 regularization based MKL formulation[12] (henceforth denoted by **MKL**). Hence here we compare the performance of VSKL formulations with that of **MKL**. As a baseline we also compare performance with an SVM classifier built using the kernel as sum of all the given kernels (henceforth denoted by **SVM**).

From each feature descriptor, five kernels were generated by varying the width-parameter of the Gaussian kernel (from 10^{-4} to 1 on a log-scale). Since the resulting kernels are naturally grouped according to the descriptor they were generated from and also it is true that each feature descriptor is critical (may not be equally critical) for good categorization, it is obvious to employ the proposed VSKL formulations by assuming kernels are grouped according to descriptors generating them. Thus, in case of the Caltech data sets, $n = 4$ and $n_j = 5 \; \forall \; j$ and in case of Oxford flowers data set, $n = 7$ and $n_j = 5 \; \forall \; j$. Note that **SVM** and **VSKL**$_1$ differ exactly in the way the kernels are grouped: for **VSKL**$_1$ the kernels are grouped by their generating feature descriptors whereas for **SVM** each group is characterized by a single kernel (that is, for **SVM** $n = 20, n_j = 1 \; \forall \; j$ in case of Caltech data set and $n = 35, n_j = 1 \; \forall \; j$ in case of Oxford flowers data set).

In order that the experimental results are comparable to others in literature, we followed the usual practice of generating training and test sets, in case of each data set, using a fixed number of images from each object category and repeating the experiments with different random selections of images. For the Caltech-101, Caltech-256 and Oxford flowers data sets we have used 15, 25, 60 images per object category as training images and 15, 15, 20 images per object category as testing images respectively. The hyper-parameters of the various formulations were tuned using suitable cross-validation procedures. In case of the Caltech-101 data set, the accuracies reported are the test-set accuracies with the tuned set of hyper-parameters, averaged over 10 randomly sampled training and test splits. Since the Caltech-256 data set has large number of classes and the experiments are computationally intensive, the results are reported only for a single split. In case of Oxford flowers data set, the accuracies are averaged over the 3 standard data splits provided with the source images.[13] Also, we employ the 1-vs-rest methodology in order to handle the multi-class problems arising in these data sets. Table 1 reports the average testset accuracies achieved with the various kernel learning techniques. The numbers in brackets appearing below each accuracy indicate the total number of SVM calls made for solving the corresponding formulation[14] and throw light on the trade-off between accuracy and computation. In addition to comparison with **SVM** and **MKL**, we also report results of comparison with the CKL formulations (Szafranski et al., 2008), which also assume kernels are grouped. Note that the CKL formulations were not previously applied to object categorization and we wish to compare them here with VSKL in order to stress on the need for solving (1) for the cases $q \geq 1$. Recall that if $q < 1$ then (1) can be solved using the wrapper

[12] The formulation employed by Varma and Ray (2007) and Nilsback and Zisserman (2008) also has additional constraints for including prior information regarding weights of kernels. Since such constraints lead to independent improvements with all MKL formulations, the experiments here compare MKL formulations without the additional constraints.

[13] Available at `http://www.robots.ox.ac.uk/~vgg/data/flowers/17/datasplits.mat`.

[14] Stopping criterion was choosen same across different methods.

VSKL$_q$				**MKL**	**SVM**	**CKL**$_{1,2q}$		
$q=1$	$q=2$	$q=3$	$q=\infty$			$q=0.75$	$q=0.85$	$q=0.99$
Caltech-101								
66.44 (50)	67.03 (79)	67.06 (70)	**67.07** (65)	54.05 (24)	64.61 (1)	65.74 (36)	64.35 (34)	63.21 (36)
Caltech-256								
32.06 (100)	34.71 (201)	35.39 (188)	**36.69** (151)	21.07 (21)	34.04 (1)	34.43 (33)	34.40 (32)	34.43 (34)
Oxford								
85.59 (41)	85.69 (70)	85.69 (68)	85.29 (64)	85.49 (121)	85.98 (1)	**86.08** (54)	**86.08** (50)	**86.08** (51)

Table 1: Comparison of average testset accuracies achieved by the various formulations

approaches of Szafranski et al. (2008). Also recall the notation that formulation in (1) for $q \geq 1$ corresponds to **VSKL**$_q$ and for $q < 1$ corresponds to **CKL**$_{1,2q}$. The results clearly indicate that the proposed methodology is suitable for object categorization tasks and its performance better than state-of-the-art in case of the Caltech data sets; whereas in case of Oxford data set, the performance is comparable to state-of-the-art. Also, in case of oxford flowers data set, the performance of all the methods is more or less the same. Another important observation, which is especially evident in case of the Caltech-256 data set, is that the performance of VSKL depends on the parameter q and hence it is important to solve the VSKL formulation efficiently for various values of q. This demonstrates the usefulness of the proposed `mirrorVSKL` algorithm, which efficiently solves the formulation at various values of $q \geq 1$. Automatic tuning of q is indeed an open question and calls for further research. Lastly, we note that the accuracies with **SVM** and **VSKL**$_1$, which differ in the way the kernels are grouped, are noticeably different—which is expected.

4.2 Scalability Experiments

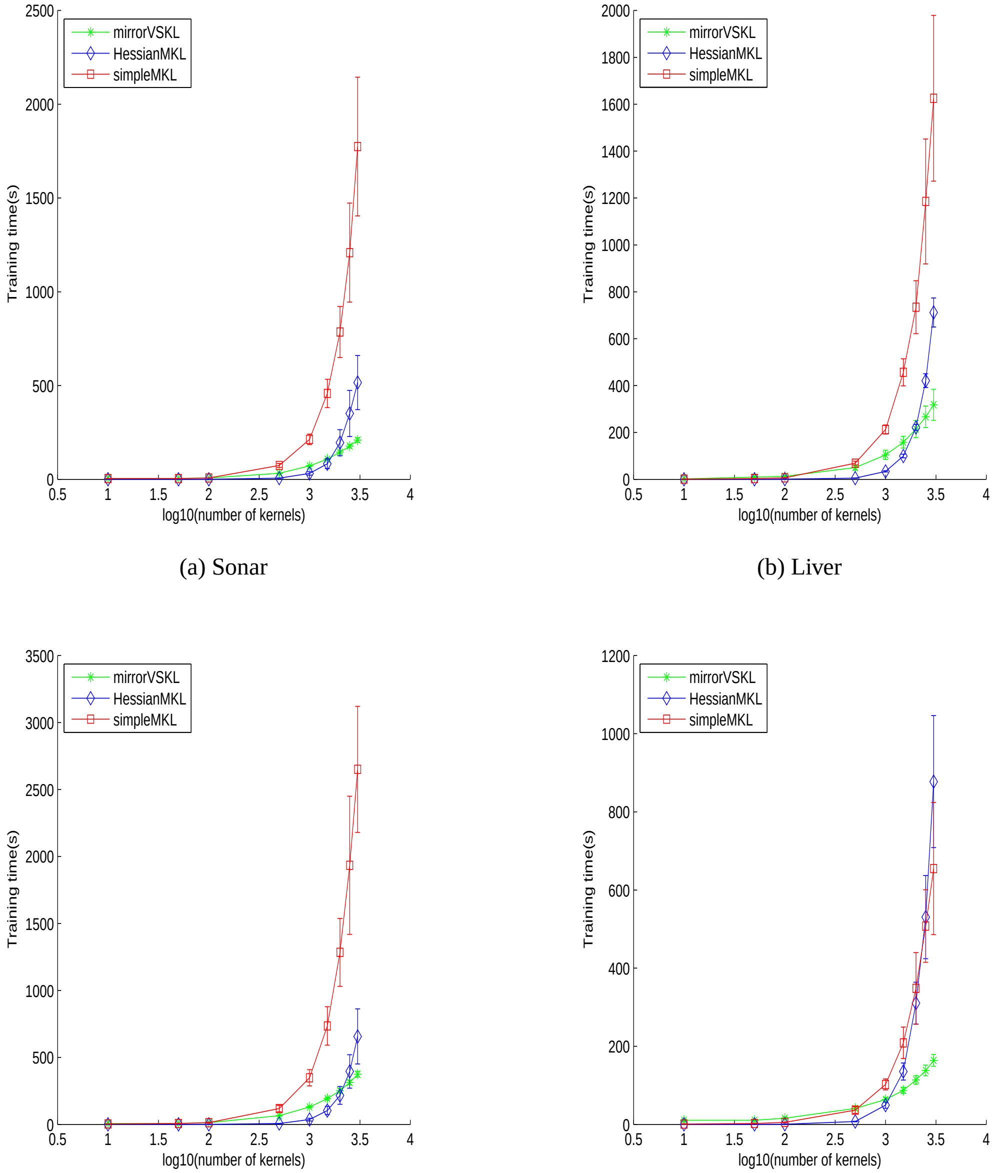
This section presents results comparing scalability of `mirrorVSKL`,[15] `SimpleMKL`[16] and Hessian-MKL[17] in solving the **MKL** formulation. Note that all these algorithms solve an SVM problem at each step and hence are comparable. For fairness in comparison, the SVM problem arising at each step was solved using the same solver in case of all the three algorithms. The stopping criteria employed in all cases was relative difference in objective value being less than 10^{-4} (that is, $(f_{old} - f_{new})/f_{old} < 10^{-4}$. The evaluation was made on four data sets from the UCI repository (Blake and Merz, 1998): Liver, Wpbc, Ionosphere and Sonar. Following the experimental set-up of Rakotomamonjy et al. (2008), each data set was split into training and test sets using 70% and 30% data points respectively. For each data set, kernels were generated based on individual features using different width parameters for the Gaussian kernel. Figure 1 compares the average time[18] taken for solving the formulation (this excludes time taken for building kernels) over 20 different random training-test splits as a function of the number of kernels. The value of regulariza-

[15] Code available at `http://mllab.csa.iisc.ernet.in/vskl.html`.

[16] Code available at `http://asi.insa-rouen.fr/enseignants/~arakotom/code/mklindex.html`.

[17] Code available at `http://www.chapelle.cc/olivier/ams/`.

[18] The standard deviation in the time taken is also shown using vertical bars at each point in the plot.



(a) Sonar

(b) Liver

(c) Ionosphere

(d) Wpbc

Figure 1: Plots of average time (in secs) taken by various solvers

tion parameter C was fixed at 1000 in all cases. The figure clearly shows that `mirrorVSKL` scales better than `simpleMKL` and `HessianMKL`. When large number of candidate kernels are available, `mirrorVSKL` outperforms them in terms of computational performance. In some cases, the solving time with the proposed method is as low as around $1/8$ of that with `simpleMKL` and around $1/6$ of that with `HessianMKL`!

The proposed algorithm scales better than simpleMKL primarily because of two reasons: firstly, the per-step auxiliary problem in case of the proposed algorithm has an analytical solution and the step-size also can be chosen very easily. Hence the predominant computation at every step is only that of computing the gradient (that is, solving the SVM). However, in case of simpleMKL, the reduced gradient needs to be computed and moreover the step-size needs to be computed using a 1-d line search (which may further involve solving few SVMs). Also, in case of HessianMKL, the per-step cost is high mainly due to the second order computations. Secondly, the number of iterations in solving the formulation is nearly-independent of the number of kernels in case of the proposed MD based algorithm. However no such statement can be made in case of either simpleMKL or HessianMKL.

In order to get a better insight, the number of SVM calls made by simpleMKL and mirrorVSKL (both of which are first order methods and hence comparable wrt. number of iterations/SVM calls) are compared in Figure 2. It is interesting to see that the number of SVM calls more or less remains a low value in case of mirrorVSKL; whereas it shoots up steeply in case of simpleMKL. The fact that the number of SVMs calls is low also implies that mirrorVSKL scales better than simpleMKL even wrt. no. of examples and hence is ideal for applications with large data sets as well as large number of candidate kernels.

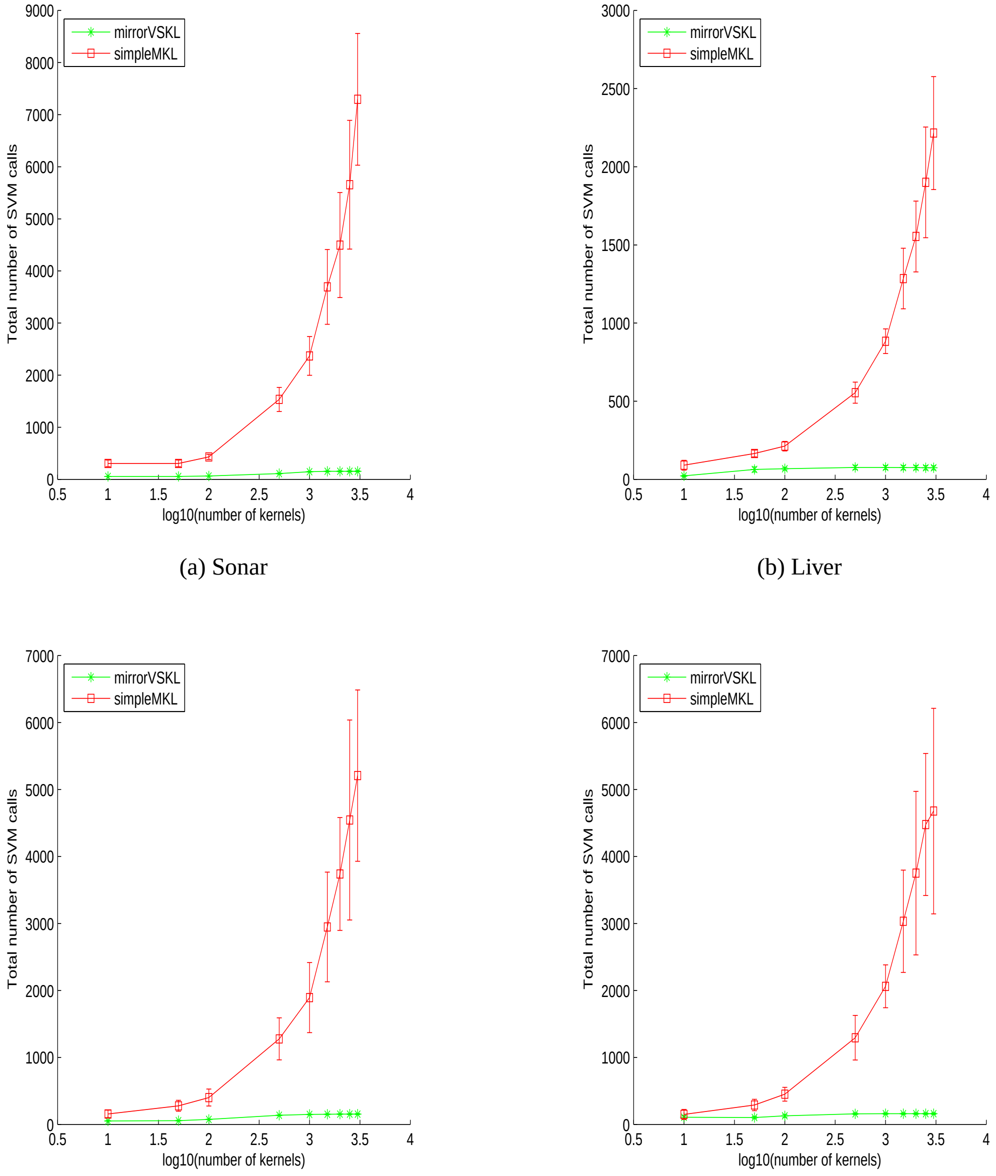
Also, it was observed that the number of iterations required by the BCA algorithm to converge (with various values of q) was typically very small. In case of all data sets, the maximum number of iterations for convergence of BCA was 4 iterations. Hence the number of iterations required by the BCA algorithm can be assumed to be a constant and the computational complexity bound $O\left(m^2 n_{\text{tot}} \log n_{\max}/\varepsilon^2\right)$ indeed is valid.

5. Conclusions

This paper makes two important contributions to the MKL literature: a) a specific mixed-norm regularization based MKL formulation which is well-suited for object categorization and other multi-modal tasks is studied. b) An efficient mirror-descent based algorithm for solving the new formulation is proposed. Since the traditional MKL formulation can be realized as a special of the proposed formulation, the efficient algorithm is also of independent interest to the MKL community. A detailed proof of convergence of the algorithm was also presented. Empirical results show that the new formulation achieves far better generalization than state-of-the-art object categorization techniques. Scaling experiments show that the mirror-descent based algorithm outperforms traditional gradient descent based approaches. In some cases the proposed MD based algorithm achieved a 8 times speed-up over simpleMKL!

Acknowledgments

We would like to thank Dinesh (previously at IISc, Bangalore and currently at Bell Labs, Bangalore) for his help with the initial version of the paper and the scripts for running the object categorization experiments. AB was supported by the US-Israel Binational Science Foundation. Part of the research was done while AB was a Visiting Professor at the Center for Mathematics and computer Science (CWI) in Amsterdam. CB was partially supported by the Yahoo! faculty award.



(a) Sonar

(b) Liver

(c) Ionosphere

(d) Wpbc

Figure 2: Plots of average number of SVM calls with `simpleMKL` and `mirrorVSKL`

Appendix A.

In this section we prove proposition 3.1, which says that G is convex and Lipschitz continuous under a mild regularity condition—all the eigenvalues of the given gram-matrices are finite and non-zero:

Proof The convexity of G follows from the fact that it is point-wise maximum over functions of the form

$$f(\alpha,\gamma,\lambda) = \mathbf{1}^T\alpha - \frac{1}{2}\alpha^T\left[\sum_{j=1}^{n}\left(\frac{\sum_{k=1}^{n_j}\lambda_{jk}\mathbf{Q}_{jk}}{\gamma_j}\right)\right]\alpha,$$

which are linear w.r.t λ. A sufficient condition for G to be Lipschitz continuous is the sub-gradient should be norm bounded. Define $D_{jk} = \alpha^{*\top}Q_{jk}\alpha^*$ where α^* and γ^* denote optimal values, that maximize $f_\lambda(\alpha,\gamma)$, for a given λ. From the definition of τ and μ we immediately have the following bound

$$\tau\mu\|\alpha^*\|_2^2 \leq D_{jk} \leq \mu\|\alpha^*\|_2^2.$$

The sub-gradient vector, evaluated at any λ, can be obtained by differentiating G at α^* and γ^*. The strategy would be to exploit the above limits on D_{jk} to bound the norm of the sub-gradient. To this end we eliminate γ in G (using proposition 3.3) and then examine the sub-gradient:

Case $q > 1$

$$\frac{\partial G}{\partial\lambda_{jk}} = \begin{cases} -\frac{1}{2}D_{jk}\left\{\frac{\sum_{j'}\left(\sum_{k'}\lambda_{j'k'}D_{j'k'}\right)^{\frac{q^*}{q^*+1}}}{\sum_{k'}\lambda_{jk'}D_{jk'}}\right\}^{\frac{1}{q^*}} & \text{if } \sum_{k'}\lambda_{jk'}D_{jk'} > 0, \\ 0 & \text{otherwise.}\end{cases}$$

Case $q = 1$

$$\frac{\partial G}{\partial\lambda_{jk}} = \begin{cases} -\frac{1}{2}D_{jk} & \text{if } \sum_{k'}\lambda_{jk'}D_{jk'} > 0, \\ 0 & \text{otherwise.}\end{cases}$$

From these equations, it is easy to see that:

$$\left|\frac{\partial G}{\partial\lambda_{jk}}\right| \leq \frac{1}{2}\left(\frac{n}{\tau}\right)^{\frac{1}{q^*}}\left(\mu\|\alpha^*\|_2^2\right)^{\frac{q^*(q^*+1)-1}{q^*(q^*+1)}}.$$

In case $q = 1$, we have $\left|\frac{\partial G}{\partial\lambda_{jk}}\right| \leq \frac{1}{2}\mu\|\alpha^*\|_2^2$. Now, we know that $\alpha \in S_m \Rightarrow \alpha_i < C \ \forall\ i \Rightarrow \|\alpha^*\|_\infty \leq C \Rightarrow \|\alpha^*\|_2 \leq \sqrt{m_{sv}}C$ where m_{sv} is the number of support vectors. These relationships shows that $\|\nabla_\lambda G\|_\infty \leq L_G$ where

$$L_G = \begin{cases} \frac{1}{2}\left(\frac{n}{\tau}\right)^{\frac{1}{q^*}}\left(\mu m_{sv}C^2\right)^{\frac{q^*(q^*+1)-1}{q^*(q^*+1)}} & \text{if } q > 1, \\ \frac{1}{2}\mu m_{sv}C^2 & \text{if } q = 1.\end{cases}$$

Now, since $\|\nabla_\lambda G\|_\infty$ is bounded, we have that G is Lipschitz continuous with respect to l_1 norm with Lipschitz constant L_G. ■

References

F. Bach, G. R. G. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *Proceedings of the International Conference on Machine Learning*, 2004.

A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31:167–175, 2003.

A. Ben-Tal and A. Nemirovski. Lectures on modern convex optimization: Analysis, algorithms and engineering applications. *MPS/ SIAM Series on Optimization*, 1, 2001.

A. Ben-Tal, T. Margalit, and A. Nemirovski. The ordered subsets mirror descent optimization method with applications to tomography. *SIAM Journal of Optimization*, 12(1):79–108, 2001.

A. C. Berg, T. L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, second edition, 1999.

C. Blake and C. Merz. Uci repository of machine learning databases, 1998. URL `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, 2006.

A. D'Aspermont. Smooth optimization with approximate gradient. *SIAM Journal on Optimization*, 19(3):1171–1183, 2008.

L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Workshop on Generative-Model Based Vision*, 2004.

G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report CNS-TR-2007-001, California Institute of Technology, 2007.

M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien. Non-sparse regularization and efficient training with multiple kernels. Technical Report UCB/EECS-2010-21, University of California, Berkeley, 2010.

G.R.G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M.I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

S. Lazebnik, C. Schmid, and J. Ponce. Beyond bag of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

C. A. Micchelli and M. Pontil. Learning the kernel function via regularization. *Journal of Machine Learning Research*, 6:1099–1125, 2005.

J. S. Nath, G. Dinesh, S. Raman, C. Bhattacharyya, A. Ben-Tal, and K. R. Ramakrishnan. On the algorithmics and applications of a mixed-norm regularization based kernel learning formulation. In *Advances in Neural Information Processing Systems (NIPS)*, 2009.

M-E. Nilsback and A. Zisserman. A visual vocabulary for flower classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1447–1454, 2006.

M-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Proceedings of the Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, 2008.

A. Quattoni, X. Carreras, M. Collins, and T. Darrell. An efficient projection for $l_{1,\infty}$ regularization. In *Proceedings of the International Conference on Machine Learning*, 2009.

A. Rakotomamonjy, F. Bach, S. Canu, and Y Grandvalet. Simplemkl. *Journal of Machine Learning Research*, 9:2491–2521, 2008.

R. T. Rockafellar. Minimax theorems and conjugate saddle-functions. *Mathematica Scandinava*, 14:151–173, 1964.

R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

M. Sion. On general minimax theorem. *Pacific Journal of Mathematics*, 1958.

S. Sonnenburg, G. Ratsch, C. Schafer, and B. Scholkopf. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7:1531–1565, 2006.

M. Szafranski, Y. Grandvalet, and A. Rakotomamonjy. Composite kernel learning. In *Proceedings of the International Conference on Machine Learning*, 2008.

P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimisation. *Journal of Optimization Theory and Applications*, 2001.

V. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

M. Varma and D. Ray. Learning the discriminative power invariance trade-off. In *Proceedings of the International Conference on Computer Vision*, 2007.

A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *Proceedings of the International Conference on Computer Vision*, 2009.

H. Zhang, A. Berg, M. Maire, and J. Malik. Svm-knn: Discriminative nearest neighbor classication for visual category recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2126–2136, 2006.