S^{3}MKL: Scalable Semi-supervised Multiple Kernel Learning for Image Data Mining

Shuhui Wang[1] Shuqiang Jiang[1] Qingming Huang[1, 2] Qi Tian[3]

[1]Key Lab of Intell. Info. Process., Inst. of Comput. Tech., CAS, Beijing, 100190, China
{shwang,sqjiang}@jdl.ac.cn

[2]Graduate University, Chinese Academy of Sciences, Beijing, 100049, China
qmhuang@jdl.ac.cn

[3]Dept. of Computer Science, Univ. of Texas at San Antonio, TX78249, U.S.A.
qitian@cs.utsa.edu

ABSTRACT

For large scale image data mining, a challenging problem is to design a method that could work efficiently under the situation of little ground-truth annotation and a mass of unlabeled or noisy data. As one of the major solutions, semi-supervised learning (SSL) has been deeply investigated and widely used in image classification, ranking and retrieval. However, most SSL approaches are not able to incorporate multiple information sources. Furthermore, no sample selection is done on unlabeled data, leading to the unpredictable risk brought by uncontrolled unlabeled data and heavy computational burden that is not suitable for learning on real world dataset. In this paper, we propose a scalable semi-supervised multiple kernel learning method (S^{3}MKL) to deal with the first problem. Our method imposes group LASSO regularization on the kernel coefficients to avoid over-fitting and conditional expectation consensus for regularizing the behaviors of different kernel on the unlabeled data. To reduce the risk of using unlabeled data, we also design a hashing system where multiple kernel locality sensitive hashing (MKLSH) are constructed with respect to different kernels to identify a set of "*informative*" and "*compact*" unlabeled training subset from a large unlabeled data corpus. Combining S^{3}MKL with MKLSH, the method is suitable for real world image classification and personalized web image re-ranking with very little user interaction. Comprehensive experiments are conducted to test the performance of our method, and the results show that our method provides promising powers for large scale real world image classification and retrieval.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing; I.2.6 [**Artificial Intelligence**]: Learning; I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis

General Terms

Algorithms, Experimentation.

Keywords

scalable semi-supervised learning, multiple kernel learning, multiple kernel locality sensitive hashing, image categorization, personalized image re-ranking.

MM'10, October 25–29, 2010, Firenze, Italy.

1. INTRODUCTION

Due to the popularity of digital camera and the ubiquitousness of the Internet, digital photos and videos are produced, uploaded and exchanged online every day. This leads to the explosive growth of web multimedia data. For such a huge database with massive un-annotated images, effective methods for web media data mining are indispensible for ordinary users. Without these, it is very hard for people to retrieval what they want from the Internet. One major solution for this urgent need is to train a set of automatic image annotators on well-labeled images, and annotate the new images with one or more visual concepts. Then the users may easily find those relevant images by querying with text. Numerous literatures have been devoted to this area [4, 7, 8, 18, 24, 28, 32, 35, 37, 38].

Although promising results have been reported these years, the visual classification is still far from being perfect. The most realistic issue for training a good model is that it is laborious to obtain sufficiently large scale well-labeled database. Though a lot of efforts have been devoted to building up large scale ground truth data, such as ImageNet [6], the growth of labeled data would never catch up with the growth of new un-annotated images on the web. Therefore, semi-supervised learning (SSL) [39], which could employ both labeled and unlabeled data, would be a better choice than supervised learning which is only based on labeled data.

Nevertheless, for real world image and video classification, most of SSL may only work on some small size and clean datasets. The reasons lie in two folds. Firstly, one cause of *semantic gap* is the lack of discrimination power of every single feature under uncontrolled appearance changes such as light, occlusion and various viewing angles. To bridge the *semantic gap*, a practical solution is to combine the discrimination power of heterogeneous features. Since most previous SSL are based on the manifold assumption [21, 22, 39, 40] or cluster assumption [20] on specific graph and feature representation, they are not able to incorporate multiple features efficiently.

Many studies have been devoted to effectively utilizing multiple features, from simple feature concatenation [24], to feature weight determination. Recently, Multiple Kernel Learning (MKL) [2, 14, 19, 25] has been proposed and applied to image classification. Unlike previous single kernel machines and manual kernel weight setting, MKL is capable of approximating the optimal similarity measures by optimizing the weights of the linear combination of a set of kernels, and minimizing the objective function simultaneously. More recently, Multiple Kernel Learning methods with sample and group specific kernel coefficient are proved to be more effective than the original MKL with uniform kernel weight [4, 9, 37]. However, studies have been

rarely done, except Tsuda *et al*. [31], on how MKL employs unlabeled data to improve the prediction power of the model. However, being directly derived from graph based SSL [40] and the original uniform MKL approaches [2, 14, 19, 25], the semi-supervised MKL in [31] is transductive, which could not flexibly predict an unseen sample. Also, it is not suitable for large scale image classification since expensive computations are required.

To mitigate the *semantic gap* in semi-supervised learning scenario, we propose S^3MKL, an inductive and scalable semi-supervised multiple kernel learning method. S^3MKL incorporates different kernels in a sample specific way, which provides more precise feature/kernel fusion capability. The proposed model regularizes the kernel coefficients by group LASSO to ensure the sparsity over group level. We also propose a new regularization that is suitable for large scale SSL, which penalizes the expectation inconsistence on unlabeled data with respect to different kernels, since some expectations, such as the "*label prior*" is very easy to obtain in real world [15, 16]. To train S^3MKL, we propose a fast and robust optimization method that is an improved version of the Block Coordinate Gradient Descent method (BCGD) [30]. Our optimization method converges more than 2 times faster than [30], as we heuristically impose more computational resources on positive data.

The second limitation of previous SSL approaches is that most of them assumed implicitly, that there are some positive examples among the unlabeled data corpus. However, the issues of how to identify "*informative*" unlabeled samples and how to filter those "*harmful*" samples are usually ignored by most studies because this is not necessary for many clean machine learning datasets. In [23], a theoretical analysis of how unlabeled data will help is provided, but it is too strict to be generalized in real world application. Our intuition comes from the fact that unlabeled data is not always helpful for model enhancement without certain control, especially for web image retrieval, where noisy information usually pervades. For example, a good model for detecting images of horse could be impaired by training with unlabeled data with all car images.

To this point, a system that is able to obtain an "*informative*" and "*compact*" unlabeled dataset is pursued in our study. The reason is understandable, since filtering out "*irrelevant*" samples can not only reduce the chance of model degradation, but also save a lot of computational resources. We adopt the kernel version of LSH (KLSH) [13] to construct an approximated nearest neighbor search system. Moreover, to combine the discrimination power of different features/kernels, we build up a multiple kernel LSH (MKLSH) based on a set of KLSH functions with respect to several kernels. The advantage over a single kernel hashing system is that it can reduce the search bias caused by specific distance metric by combining multiple visual properties of the images.

As a summary, the key contributions of this paper include: (1) S^3MKL, a scalable semi-supervised multiple kernel learning method for classifier training with labeled and unlabeled data and multiple kernels; (2) A more faster optimization method, the heuristic BCGD for training S^3MKL; (3) MKLSH, an approximated nearest neighbor search system based on multiple KLSH; and (4) solution of using S^3MKL and MKLSH for real world image annotation and personalized web image re-ranking.

The rest of the paper is organized as follows. Section 2 reviews related works. In Section 3 we introduce S^3MKL and the optimization procedure. In Section 4 the detail of MKLSH is discussed. In Section 5, we present how to combine S^3MKL and MKLSH for image classification and personalized image re-ranking. Section 6 introduces the experiments. Finally, Conclusion and future work are provided in Section 7.

2. RELATED WORK

Our work is mainly related to three groups of research. The first is object recognition and automatic image/video annotation. The second is semi-supervised learning, and the third is learning with multiple features/kernels. Due to the limited space, we only brief some recent and representative works on these three aspects.

2.1 Image Annotation and Object Recognition

Image annotation has always been a hot issue in computer vision and multimedia communities. A well studied paradigm, as we call annotation by learning, is to train a set of automatic annotators based on a collection of labeled image data using learning method such as SVM [8, 18] and adaboost [28, 35]. Specifically, a localized approach was proposed by [38], in which every test sample is predicted by the SVM classifier trained on its nearest k samples. It is very time-efficient and promising results are reported. Our method is related to [38], but we apply nearest neighbor search to find some informative unlabeled data, while their aim is to find the training sample that is informative to the test samples.

The study of annotation by learning usually suffers from the shortage of ground truth data. Many studies have been devoted to build up a large scale ground truth database [6, 29], with the cooperative work of human and artificial intelligence [6, 29, 36]. Since the growth of annotated images can not catch up with the growth of new unlabeled images, to efficiently exploit the abundant unlabeled image resources, semi-supervised learning [20, 26] is studied in the context of image application.

Recently, there are growing interests in solving annotation and object recognition by exploring the web repository. The advantage of this method is that it can deal with hundreds and thousands of image categories by utilizing the abundant tagging information provided by users on the web. Wang *et al*. [33] proposed a scalable search based method. Torralba *et al*. [27] declared that with extremely large tiny image database, say 80 millions, simple methods such as k-NN could work significantly well for image annotation, although the tags of the 80 million images are very noisy. This statement was also emphasized by Deng *et al*. [6]. Wu *et al*. [34] proposed a novel probabilistic distance metric learning approach to deal with the noisy tag based on RCA. These methods are flexible in processing popular queries such as images of human or pets, which are very abundant on the web. However, the models will degrade when the returned web images contains less true tags and more noise, especially when considering the fact that the user tags usually include personal preferences. In this case, we argue that annotation by learning still plays an important role.

2.2 Semi-supervised Learning

Instead of only minimizing the empirical risk or structural risk, *semi-supervised learning* also minimizes the risk defined on unlabeled data based on some assumptions, such as: (a) Manifold assumption [21, 22], which assumes that similar data would be more likely to have the same label; (b) the max-margin criterion [12, 20], which prevents the classification boundary located on higher density areas, and maximizes the margin defined on both labeled and unlabeled data; (c) minimum entropy [10], in which the entropy of some conditional distribution on unlabeled data is

minimized. To apply SSL for studies of image annotation, in [26], the graph-based SSL [39, 40] is modified to incorporate the local density difference of samples, and achieved promising results on video concept annotation task. In [20], a semi-supervised multi-class boosting method is proposed, which is based on cluster assumption and expectation regularization [15, 16] is applied to for regularization on unlabeled data, and the defined "*margin*" on both labeled and unlabeled data is maximized.

A relevant semi-supervised learning of incorporating multiple features is co-training [3] where two classifiers boost each other by introducing new reliably predicted unlabeled training samples to each other. This knowledge is then promoted in multi-view learning that multiple hypotheses are trained from different views of the same labeled data, and are required to make consistent predictions on any given unlabeled instance [22]. Multi-view learning could be easily distinguished from our method since the former trains several classifiers simultaneously, and ensemble all the classifiers in a late fusion style [24], while our target is to take advantage of the discriminative power provided by all the kernels.

The most related work with our study could be found in [31], where multiple feature networks is combined with graph based SSL. However, it is not capable for real world image retrieval, because it is transductive and not flexible to predict unseen samples; it requires expensive inverse matrix operation for minimizing the objective function; uniform weight is assigned to the graph *Laplacian* of each kernel among all the samples, which have more restricted fusion capability than sample dependent kernel weight assignment [4, 9, 37].

2.3 Employing Multiple Features/Kernels

A lot of studies have been conducted on efficiently combining the discrimination power of different features. A simple scheme is feature concatenation [24], but sparseness in feature space will be introduced, leading to the "*curse of dimensionality*". Another solution is the late fusion [24] of individual classifier, but it just obtains a more stable decision based on the individual decision output.

As one of the most promising feature fusion methods, Multiple Kernel Learning [2, 14, 19, 25] combines multiple features in the way of linear combination of kernels. It avoids the "*curse of dimensionality*" incurred by feature concatenation. The MKL has also been used for object recognition [19, 32], and promising results have been reported on many challenging datasets.

Early studies on MKL usually impose uniform kernel weighted combination to each sample. In more recent studies, kernel weight is assigned differently to each sample or each group of samples [9, 37]. Compared with the uniform kernel weight setting, the local weight approaches perform more favorable because the local kernel weight provides more deliberate fusion capability. The disadvantage of studies in [9, 37] is that no global convergence is guaranteed. A new localized MKL is proposed by [4], where group LASSO is applied directly on regularizing the local kernel coefficients, instead of fitting a gating function for the calculation of the coefficients as in [9, 37]. By using the BCGD method [30], the model is guaranteed to converge to global optimal solution.

3. S^{3}MKL

3.1 Problem and Objective Function

Suppose we are given a dataset of N training samples of two classes: $(x_i, y_i, o_i), y_i \in \{0,1\}, o_i \in \{0,1\}, 1 \le i \le N$ where y_i denotes the label of the i^{th} training sample if the sample is labeled, *i.e.*, $o_i = 1$ and $o_i = 0$ for the unlabeled samples. We use $L = \{(x_i, y_i) \mid o_i = 1\}$ to denote the set of labeled data, and $U = \{(x_i) \mid o_i = 0\}$ for unlabeled data in the rest of the paper.

The discriminative function $\mathbf{f}$ in our learning framework is a kernel logistic regression model integrating M different kernels. It is formulated as:

$$\mathbf{f}(x) = \alpha_0 + \sum_{i=1}^{|L|} \boldsymbol{\alpha}_i \mathbf{K}_i(x) \tag{1}$$

where $\boldsymbol{\alpha}_i = [\alpha_i^1, ..., \alpha_i^M]$ is the unknown kernel logistic regression parameter of the i^{th} sample and $\mathbf{K}_i(x) = [K_1(x_i, x), ..., K_M(x_i, x)]^T$ denotes the similarity measures between x_i and x with respect to M different kernels; α_0 is the bias term. Under the binary classification scenario, the probability of sample x belonging to the positive class (+1) is calculated as: $P(1/x) = (1 + e^{-f(x)})^{-1}$. It is very easy to extend this model to the multi-class version, but in this paper, we only demonstrate how our method works in binary classification.

We minimize the following objective function with respect to the classification function $\mathbf{f}$:

$$\min_{\mathbf{f}} Q(\mathbf{f}) = \min_{\mathbf{f}} Z(\mathbf{f}) + \gamma \Theta(\mathbf{f}) + \lambda \Omega(\mathbf{f}) \tag{2}$$

where $Z(\mathbf{f})$ denotes the weighted negative log-likelihood loss on labeled data; $\Omega(\mathbf{f})$ denotes the group LASSO [1, 4, 17] regularization on $\mathbf{f}$, which ensures the model parameter is sparse on group level, and $\Theta(\mathbf{f})$ denotes the conditional expectation consensus penalty on unlabeled data, which regularizes the behavior of different kernels on unlabeled data by minimizing the difference of the conditional expectation on unlabeled data and the given reference expectation. γ and λ are the weights of Θ and Ω respectively. It is worth noting that when $\gamma = 0$, the objective function is identical to [4] and becomes a supervised learning method.

The loss on the labeled data is defined as the weighted negative log-likelihood as:

$$Z(\mathbf{f}) = \sum_{x_j \in L} b_j \log(1 + \exp(\mathbf{f}(x_j))) - \sum_{x_j \in L} b_j y_j \mathbf{f}(x_j) \tag{3}$$

where b_j denotes the weight of the j^{th} sample. The weight factor b_j is introduced to avoid the class imbalance problem. In this paper, we set $b_j = 2$ for positive data, and $b_j = 1$ for negative data.

3.2 Group LASSO Regularization

Previous studies have shown that l_1 regularization will lead to the sparse solution of the model parameter [11]. However, in practical situation, people often know the group structure on the model parameter, so that parameters in the same group tend to be zeros or non-zeros simultaneously. In this case, group LASSO [1, 4, 11, 17], which ensures the model is sparse on group level is more preferred. Huang *et al*. [11] studied the benefit of group LASSO, and provides a convincing theoretical justification for using group sparse regularization when the underlying group structure is consistent with the data. In our method, we define the "group" as all the kernel coefficients corresponding to each labeled sample, and the regularization term could be written as:

$$\Omega(\mathbf{f}) = \sum_{i=1}^{|L|} \| \boldsymbol{\alpha}_i \|_2 \tag{4}$$

The group LASSO regularization in (4) prevents those outlier samples to be chosen into the model by setting all the kernel coefficients of the sample to zero. This function leads to the sparsity of the kernel coefficients $\boldsymbol{\alpha}_i$ on group level. Comparison with SVM would be very interesting. Although the inherent mechanisms are quite different, for both methods only a small portion of samples will contribute to the final model.

In fact, when using larger λ, the model tends to be sparser [4]. In this paper, we found that $\lambda = 0.01 \sim 0.05$ is a reasonable choice to guarantee the performance and robustness of the model. However, the setting of λ need not to be precisely tuned for every dataset. After a preliminary experiment, for all the experiments in this paper, we set $\lambda = 0.02$.

3.3 Conditional Expectation Consensus

We propose conditional expectation consensus based on the idea from expectation regularization [15, 16]. It is a scalable regularization framework for semi-supervised learning method, especially for exponential family parametric models. It augments the traditional conditional label-likelihood objective function with an additional term that encourages model predictions on unlabeled data to match certain reference expectations. This regularization is suitable for solving the real world classification problem, for example, the ratio of "car" images in a large set of web images could be easily obtained from the web even we do not know exactly whether a certain image contains a car. We could also obtain the knowledge of how likely an image would contain a car if the word "car" appears in the surrounding text. The former situation is known as the "*label priors*", and the latter is an example of "*feature labeling*" [15]. Both of them could be used as the reference expectation in the expectation regularization framework.

The expectation regularization could be seen as a generalized version of entropy regularization [10]. However, there is no such function in expectation regularization for regularizing the behavior of the coefficients of different kernels. To adapt the expectation regularization for multiple kernel learning, we propose a new regularization based on [15, 16], which we call conditional expectation consensus in this paper. We regularize the conditional label distribution on each kernel, where the response of each data x_i on each kernel is denoted as $g_m^\pi(x)$, then the conditional expectation consensus regularization is formulated as:

$$\Theta(\mathbf{f}) = \frac{1}{\Pi}\sum_{\pi=1}^{\Pi}\sum_{m=1}^{M} D(\hat{p}_m^\pi(y \mid g_m^\pi(x)) \,\|\, \tilde{p}_m^\pi(\mathbf{f}(x) \mid g_m^\pi(x); \boldsymbol{\alpha})) \tag{5}$$

where D denotes the KL-divergence of $\hat{p}$ and $\tilde{p}$, the kernel is indexed by m, where $m = 1,...,M$. We repeatedly generate Π responses on m^{th} kernel by random projection for each kernel. We denote the index of random projection as π. $\hat{p}$ represents a reference conditional distribution, either provided by some prior knowledge by "*feature labeling*" [15, 16], or calculated using the labeled data. $\tilde{p}$ represents the conditional distribution of the predicted class over the unlabeled data. The two conditional distributions are calculated using the following equation:

$$\begin{aligned} \hat{p}_m^\pi(y \mid g_m^\pi(x)) &= \frac{1}{G_m^\pi}\sum_{x\in L} p\big(y(x)\big) g_m^\pi(x), \quad G_m^\pi = \sum_{x\in L} g_m^\pi(x) \\ \tilde{p}_m^\pi(\mathbf{f}(x) \mid g_m^\pi(x)) &= \frac{1}{C_m^\pi}\sum_{x\in U} p\big(\mathbf{f}(x)\big) g_m^\pi(x), \quad C_m^\pi = \sum_{x\in U} g_m^\pi(x) \end{aligned} \tag{6}$$

$g_m^\pi(x)$ is a random projective inner product function such as:

$$g_m^\pi(x) = \left\langle \phi_m^\pi, \psi_m(x) \right\rangle \tag{7}$$

where $\psi_m(x) = [K_m(x_1, x), ..., K_m(x_L, x)]^T$. ϕ_m^π is a random vector which is independently generated from a multi-variant Gaussian distribution. The aim of random projection is to project the representation of certain kernel into one-dimensional conditional distribution, and then the conditional distribution could be obtained by using equation (6). To avoid the projection bias, we adopt the criterion of projecting Π times for each kernel. In our experiment, we set $\Pi = 5$ for all the experiments in this paper.

In practice, we also find that γ does not need careful tuning for each dataset. Unless specified in this paper, we empirically set $\gamma = 0.1 \times$ #labeled examples.

3.4 Optimization

To effectively minimize $Q(\mathbf{f})$, we propose a new solution based on the Block Coordinate Gradient Descent (BCGD) method by Tseng *et al*. [30]. The BCGD method was also employed in [4, 17], solving group LASSO logistic regression in different contexts. We propose heuristic BCGD, which imposes more optimization on positive data. Compared with the original BCGD, for learning problems on very unbalanced dataset, it achieves better convergence rate. We introduce the details as follows.

Since the group LASSO regularization is not differentiable everywhere, Tseng *et al*. [30] suggest that a good way of minimizing this objective function is decomposing $Q(\mathbf{f})$ into group-wise differentiable sub-problems. Quadratic approximation and line search is combined to solve every sub-problem.

Firstly, the overall loss function is rewritten as:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} Q(\boldsymbol{\alpha}) &= \min_{\boldsymbol{\alpha}} C(\boldsymbol{\alpha}) + \lambda \sum_{i=1}^{|L|} \| \boldsymbol{\alpha}_i \|_2 \\ C(\boldsymbol{\alpha}) &= Z(\boldsymbol{\alpha}) + \gamma \Theta(\boldsymbol{\alpha}) \end{aligned} \tag{8}$$

For each step, the loss function is firstly approximated by Taylor expansion:

$$Q(\boldsymbol{\alpha} + \mathbf{d}) \approx C(\boldsymbol{\alpha}) + \nabla C \cdot \mathbf{d} + \frac{1}{2}\mathbf{d}^T \mathbf{H} \mathbf{d} + \lambda \sum_i \| \boldsymbol{\alpha}_i + \mathbf{d}_i \|_2 \tag{9}$$

where $\mathbf{d}$ denotes the direction in which $\boldsymbol{\alpha}$ should be updated. $\mathbf{H}$ is a diagonal matrix approximating the Hessian of $C(\boldsymbol{\alpha})$ with the form:

$$\mathbf{H} = diag(\mathbf{H}_1, ..., \mathbf{H}_{|L|}), \ \mathbf{H}_i = h_i \mathbf{I}, \ h_i = \max(diag(\nabla_{ii}^2 C), 10^{-3}) \tag{10}$$

Since $\mathbf{H}$ is diagonal and separable, we alternatively optimize the decomposed sub-problems:

$$Q_i = C(\boldsymbol{\alpha}) + \nabla_i C \cdot \mathbf{d}_i + \frac{1}{2}\mathbf{d}_i^T \mathbf{H}_i \mathbf{d}_i + \lambda \| \boldsymbol{\alpha}_i + \mathbf{d}_i \|_2 \tag{11}$$

where the Hessian is calculated as:

$$\nabla_{ii}C = \nabla_{ii}Z + \nabla_{ii}\Theta = \sum_{j=1}^{N} b_j P_j(1)P_j(0)\mathbf{k}_i^2(x_j)$$
$$+\frac{\gamma}{\Pi}\sum_{x\in U} P_x(1)P_x(0)\mathbf{k}_i^2(x)\left(\sum_{\pi=1}^{\Pi}\sum_{m=1}^{M}\frac{g_m^{\pi}(x)}{C_m^{\pi}}\left(\frac{\hat{p}_m^{\pi}(1)}{\hat{p}_m^{\pi}(1)} - \frac{1-\hat{p}_m^{\pi}(1)}{1-\hat{p}_m^{\pi}(1)}\right)\right) \quad (12)$$
$$+\frac{\gamma}{\Pi}\sum_{x\in U} P_x^2(1)\mathbf{k}_i^2(x)\left(\sum_{\pi=1}^{\Pi}\sum_{m=1}^{M}\frac{g_m^{\pi}(x)}{C_m^{\pi}}\left(\frac{\hat{p}_m^{\pi}(1)}{\left(\hat{p}_m^{\pi}(1)\right)^2} + \frac{1-\hat{p}_m^{\pi}(1)}{\left(1-\hat{p}_m^{\pi}(1)\right)^2}\right)\right)$$

and the gradient is calculated as:

$$\nabla_i C = \nabla_i Z + \nabla_i \Theta = \sum_{j=1}^{N} b_j (P_j(1) - y_j)\mathbf{k}_i(x_j)$$
$$+\frac{\gamma}{\Pi}\sum_{x\in U} P_x(1)\mathbf{k}_i(x)\left(\sum_{\pi=1}^{\Pi}\sum_{m=1}^{M}\frac{g_m^{\pi}(x)}{C_m^{\pi}}\left(\frac{\hat{p}_m^{\pi}(1)}{\hat{p}_m^{\pi}(1)} - \frac{1-\hat{p}_m^{\pi}(1)}{1-\hat{p}_m^{\pi}(1)}\right)\right) \quad (13)$$

When $\| \nabla_i C - h_i \boldsymbol{\alpha}_i \|_2 < \lambda$, $\boldsymbol{\alpha}_i(t+1) = \mathbf{0}$, Otherwise:

$$\mathbf{d}_i = -\mathbf{H}_i^{-1}\left[\nabla_i C - \lambda \frac{\nabla_i C - h_i \boldsymbol{\alpha}_i}{\| \nabla_i C - h_i \boldsymbol{\alpha}_i \|_2}\right] \quad (14)$$

Then $\boldsymbol{\alpha}_i$ is updated by $\boldsymbol{\alpha}_i(t+1) = \boldsymbol{\alpha}_i(t) + a(t)\mathbf{d}_i$. $a(t)$ is the step size determined by Armijo rule which satisfies:

$$a(t) = \min[\delta^0, \delta^1, ..., \delta^l], 0 < \delta < 1, l > 0$$
$$s.t. \quad Q_i(\boldsymbol{\alpha}_i + a(t)\cdot \mathbf{d}_i) - Q_i(\boldsymbol{\alpha}_i) \le \sigma a(t) \| \nabla Q_i \| \quad (15)$$

where $\| \nabla Q_i \| = -\nabla C_i \cdot \mathbf{d}_i + \lambda(\| \boldsymbol{\alpha}_i + \mathbf{d}_i \|_2 - \| \boldsymbol{\alpha}_i \|_2)$. In this paper, we set $l = 20$, $\sigma = 0.618$, and $\delta = 0.5$, ensuring both efficiency and search precision. For the un-regularized bias, it is directly optimized by:

$$d_0 = -\frac{\nabla_0 C}{\nabla_{00} C}, \quad \alpha_0(t+1) = \alpha_0(t) + d_0 \quad (16)$$

In the original BCGD method, each group of coefficients is optimized cyclically. However, in many binary classification problems, the ratio of positive data is usually much smaller than that of negative data. According to our observation, more optimization of those kernel coefficients corresponding to the positive labeled data makes the convergence procedure much faster. Therefore, based on [30], we heuristically allocate more computation on coefficients of positive data, by update the coefficients of negative data every Q rounds and update the coefficients of positive data every round. We outline the procedure in Algorithm 1. In this paper, we find Q=5 is a reasonable choice.

After the training procedure, we can estimate how likely a sample belongs to the positive class by computing the probability as $P(1/x) = (1 + e^{-\mathbf{f}(x)})^{-1}$. For a multi-class problem, we adopt one-vs.-all scheme. The predicted label is the class corresponding to the largest class probability.

Algorithm 1: Heuristic BCGD solver for S^{3}MKL

1: **Initialize** $\boldsymbol{\alpha}_0(0)$, $\boldsymbol{\alpha}(0)$, t=1.
2: **while** Stop Criterion not meet and $t <= t_{max}$ **do**
3: **For each** group $\boldsymbol{\alpha}_i$, **If** $y(i) = 1$ repeat each round **else** repeat each Q rounds
4: Compute $\mathbf{H}_i$, C and $\nabla_i C$
5: **If** $\| \nabla_i C - h_i \boldsymbol{\alpha}_i(t) \|_2 < \lambda$
6: Set $\boldsymbol{\alpha}_i(t+1) = 0$
7: **else**
8: Get optimal $\mathbf{d}_i(t)$
9: Get optimal a^t using Armijo line search
10: Set $\boldsymbol{\alpha}_i(t+1) = \boldsymbol{\alpha}_i(t) + a(t)\mathbf{d}_i$
11: **end if**
12: **end For**
13: Update $\boldsymbol{\alpha}_0(t)$ by (16)
14: **end while**
15: **Output:** classification model $\boldsymbol{f}$ with coefficients $\boldsymbol{\alpha}_0$ and $\boldsymbol{\alpha}$

4. MKLSH

Nearest neighbor search is one of the key components in modern retrieval system. Among the relevant researches, locality sensitive hashing (LSH) [5] is one well-known method which performs probabilistic dimension reduction and approximated nearest neighbor search for high-dimensional data. The basic idea is to hash the input items so that similar items are mapped to the same bucket with high probability. Based on LSH, the nearest neighbors of the query could be approximately identified very quickly. Compared to other approximating nearest neighbor search methods such as kd-tree, the robustness and efficiency of LSH has been proved in many studies such as [5]. Among many versions of LSH, we adopt a recent developed kernel version of LSH [13]. The intuitive of kernel LSH is to perform LSH in an unknown high dimensional space. With similar theorem used in Kernel PCA [13], the hashing functions of KLSH are related to the kernel representation. By using the representer theorem, the kernel LSH is written as:

$$h(\phi(x)) = sign\left(\sum_{i=1}^{P} \mathbf{w}(i)\kappa(x_i, x)\right) \quad (17)$$

where $\phi(x)$ denotes the unknown feature representation in reproductive Hilbert space. The weight vector $\mathbf{w}$ is calculated as:

$$\mathbf{w} = K^{-1/2}\left(\frac{1}{T}\mathbf{e}_s - \frac{1}{P}\mathbf{e}\right), K = U\Lambda U^T, K^{-1/2} = U\Lambda^{-1/2}U^T \quad (18)$$

where K is the kernel matrix of the randomly chosen P items of the whole database, usually P is very small (we set P=400 for our experiments unless special statement is given) compared with the whole data size. $\mathbf{e}$ is a vector with P ones and $\mathbf{e}_s$ is an indicator vector for a subset S of the P items where:

$$\mathbf{e}_s(i) = \begin{cases} 1, & if\ i \in S \\ 0, & else \end{cases}, i = 1,..P \quad (19)$$

The size of S is T. In this paper we set T = 40. A set of hash mapping could be obtained by randomly choosing the subset S.

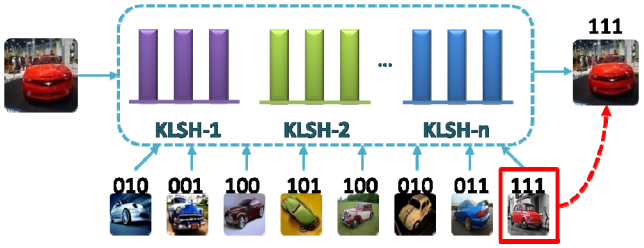

Figure 1: A multiple kernel locality sensitive hashing system.

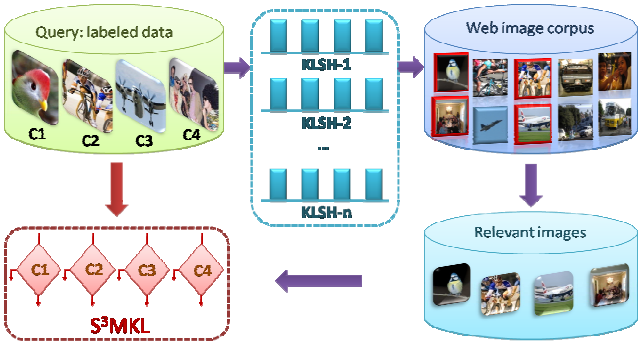

Figure 2: Semi-supervised learning with multiple kernel locality sensitive hashing.

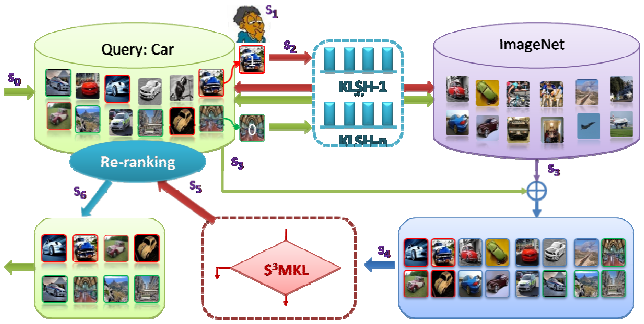

Figure 3: Framework of personalized re-ranking based on MKLSH and our semi-supervised learning method.

Since the original KLSH is built based on some specific kernel, the retrieved samples are highly dependent on the discrimination power of this kernel. To enhance the system's capacity under the scenario of real world image processing, we propose to build a KLSH system using multiple kernels. It is analogous to a nearest neighbor classifier voting, which could reduce the search variance by classifier ensemble. The system is presented in Figure 1, where the images which have the same binary code with the query are returned by the system. In our experiment, we employ four kinds of kernels as is listed in Table 3. For each kernel we generate H hash functions, As a result, a set of $4H$ dimensional binary codes are generated for each image in the database and the query images. Distance calculation is directly done on calculating the Hamming distance of the $4H$ length codes, when searching for a query's nearest neighbors, the system do not need to search over the whole database, but only check those images with the binary codes that have the top B minimum hamming distances to the code of the query image. To ensure that the approximated neighbor search has more chance to select those nearest neighbors, we repeatedly generate 3 hash tables, and then all the examples retrieved by each table are put together for each query as the resulting nearest neighbor sets.

5. APPLICATION

In this paper, we apply our method on two image applications. The first one is automatic image classification. The framework is described in Figure 2. And the second one is personalized image re-ranking. We introduce them in the following paragraphs.

In Figure 2, suppose we have a set of labeled image data. A web image dataset is firstly downloaded from the search engine. The size of the web dataset is much larger than the labeled image dataset. A multiple kernel hashing system is built both on the web image and labeled image dataset. In the second step, the unlabeled data used for S^3MKL training is those web images obtained by querying with all the labeled data. Finally, both the labeled data and the chosen "informative" unlabeled data are combined for S^3MKL training.

For the second image application, we conduct experiments on the task of personalized re-ranking in this paper. The definition of re-ranking is provided with different consideration in different studies. In this paper, we consider re-ranking as the technique that improves the results returned by the web search to better fit the expectation of individual users. Therefore, some interaction is needed during the procedure of re-ranking. For example, as Figure 3 shows, firstly the users input a "car" query to the search engine. A set of web images are returned and ranked. Then the user interface provides the users with some samples chosen randomly from the Top N ranked images. The users label the images they prefer as 1, and images they do not as 0. The numbers of the labeled images do not need to be large, where one or two is enough for the next step processing. Next, the labeled images are fit into the MKLSH system built on the ground-truth database such as the ImageNet. Some nearest neighbors of the labeled image in the "*car*" images of ImageNet are returned. Additionally, all the other unlabeled web images are also indexed with a hash code by the same MKLSH. Those images whose hash codes are in the top 3 nearest buckets of the user labeled web images are picked up as "pseudo" labeled data. Then the user labeled image, the chosen image from ImageNet, and the "pseudo" labeled images from the web query are put together as the labeled training set, and all the other web images are unlabeled training set. We conduct a transductive learning procedure by S^3MKL on these image data. Finally the web query images are re-ranked according to the output of S^3MKL.

In Figure 3, the reason we choose samples in ImageNet is that we want to augment the labeled data. However, we assume all the user preference information has been encoded in the user labeled images. Since the user would not provide many labels, to propagate the user's preference in a reasonable manner, nearest neighbor search is needed to prevent that the user preference is washed out by other images. And the reason we choose samples in the web query is that we believe that in the web query, the user may prefer more than one or two images. Therefore, propagating the preference to some possible samples within the web would help to prevent the "*concept drift*" problem of the learning process, and fully express the information of users' preference at the same time.

6. EXPERIMENTS

In this paper, we conduct a series of experiments to testify our method. In Section 6.1, we conduct experiment on a machine learning dataset, the USPS data. In Section 6.2, we provide some analysis on the training time and comparison of convergence rate of our heuristic BCGD and the original BCGD [30]. In Section 6.3, we show how MKLSH selects "*informative*" images. In Section 6.4, we present experiments on image annotation. Finally, in Section 6.5, we conduct experiment on personalized image re-ranking based on S^3MKL and MKLSH. Some discussion on experimental results is provided in Section 6.6. For all of our experiments, the experimental settings are shown in Table 1. All the features in this paper are pre-computed and all the analysis of training time does not include the time of feature computation.

Table 1: Overview of the experimental setup

Data sources
• USPS dataset [1] (Section 6.1, 6.2): hand written digit recognition data. Training/testing: 7291/2007. • PASCAL VOC 2007 [7] (Section 6.3, 6.4): Visual Object Class Challenge. Training/testing: 5011/4952. • ImageNet [6] (Section 6.5): subset of 20 classes of the large scale ground truth image data corpus. Total number: 21500. • Web data (Section 6.4, 6.5): 20 class image dataset downloaded from the web with 20 class names of VOC 2007. Total number: 48740.

Environment
OS: Windows XP; Computer: Dell Optiplex 745 desktop; CPU: Intel(R) Core Duo E6300 @1.86GHz; Memory: 3.0G RAM; Programming language: Microsoft Visual Studio 2008

6.1 Handwritten Digit Recognition

We firstly conduct experiments on USPS handwritten digit dataset. This dataset contains 10 handwritten digits, with 7291 training data and 2007 testing data. Excellent performances are achieved on this dataset using classifier combination and specific distance measure such as the tangent distance. In this paper, we only empirically generate 5 ordinary kernels using inner product, polynomial kernel and RBF kernel with 3 different bandwidths for experiments. We randomly choose part of the training data as the labeled samples, and the rest as the unlabeled data. We run 5 versions of our method, which is denoted as S^3MKL-1, S^3MKL-2, S^3MKL-3, S^3MKL-5 and S^3MKL-5+MKLSH, respectively. We also implement the supervised multi-kernel logistic regression [4] and the semi-supervised MKL [31] for comparison. For [31], we only present results under the labeled data ratio of 0.1, 0.2, 0.3, 0.4 and 0.5. An MKLSH is built on the unlabeled data by using 40 hash functions for each kernel. Since the setting of parameter γ and λ is not very sensitive, we set $\gamma = 0.02 \times |L|$ and $\lambda = 0.02$. The number of maximum iteration is set to 40. We repeat the experiments 10 times, and the mean accuracy with different ratios of labeled data is shown in Figure 4.

In Figure 4, we use S^3MKL-1 as a baseline method since it only employs one kernel. We compare the baseline with S^3MKL using 2 kernels (S^3MKL-2), 3 kernels (S^3MKL-3), 5 kernels (S^3MKL-5) and S^3MKL-5 with MKLSH (S^3MKL-5+MKLSH). We see that our method is capable of taking advantage of multiple kernels, as the accuracy increases when more kernels are incorporated. When the ratio of the labeled data is small, our methods (S^3MKL-5 and S^3MKL-5+MKLSH) significantly outperform other methods. For experiments on this dataset, the accuracy of S^3MKL-5 is almost the same as S^3MKL+MKLSH. The reason is that this dataset is very clean, and the intra-class variance is not as large as the real world image. When the size of the labeled data increases, the performance of our approach converges to [4] as the influence of the regularization of the unlabeled data reduces. We also observe that our methods outperform [31], since the sample specific kernel weight assignment provides more deliberate fusion capability than the uniform kernel weight assignment.

[1] http://www-i6.informatik.rwth-aachen.de/~keysers/usps.html

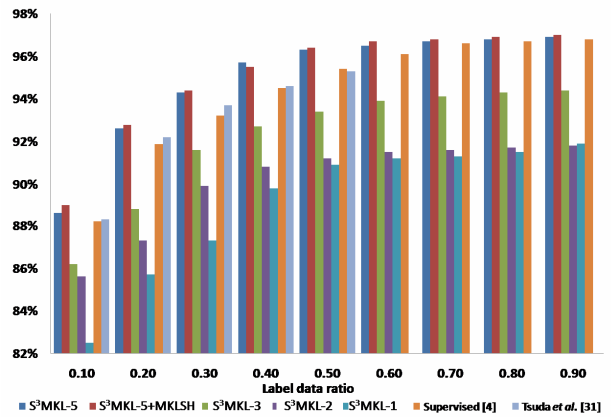

Figure 4: Accuracy with different ratio of labeled data on USPS dataset.

6.2 Optimization Issues

We compare the training time of our approach and [31] on USPS dataset. We sample 10% and the other 90% of the training samples as the labeled data and the unlabeled data for our approach, respectively. We use the same labeled data and treat the test data as unlabeled data for [31]. The result is shown in Figure 5. We see that our method is more efficient than [31] as S^3MKL-5 is 5 times faster and S^3MKL-5+MKLSH is about 10 times faster than [31]. By using MKLSH, about 50% of the unlabeled data is filtered out, which accelerates the calculation of gradient and Hessian, as could be seen in equation (12) and (13).

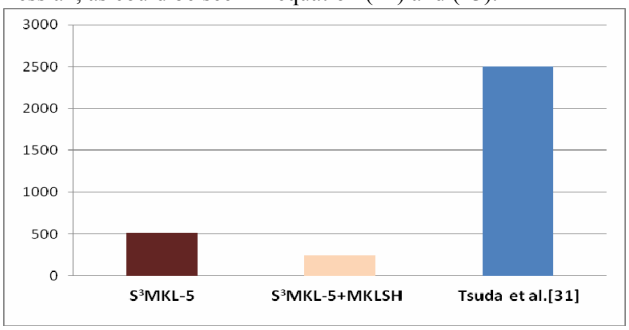

Figure 5: Training time of S^3MKL, S^3MKL+MKLSH, and [31] (in seconds).

The comparison of the convergence rate of our heuristic BCGD (Algorithm 1) and the original BCGD [30] is shown in Figure 6. The experiment is done on the same data as is used for Figure 5. In Figure 6, the red curve represents the log cost at each round using the heuristic BCGD, and the blue curve represents the cost at each round using original BCGD. Our method converges 2.5 times faster than the original one in average. The loss function achieves a predefined threshold (dashed line) after about 4500 rounds by optimizing with heuristic BCGD, but the same threshold is achieved by using the original BCGD after about 11000 rounds. The reason why heuristic BCGD is more efficiently than the original BCGD is that the ratio of zeros coefficient groups in negative training data is much larger than the ratio in positive training data. Therefore, imposing more computational resource on positive training data will help to increase the chance to find more non-zero coefficient groups while consuming as little time as possible, and reduce the times of calculating gradient and Hessian of those zero coefficient groups.

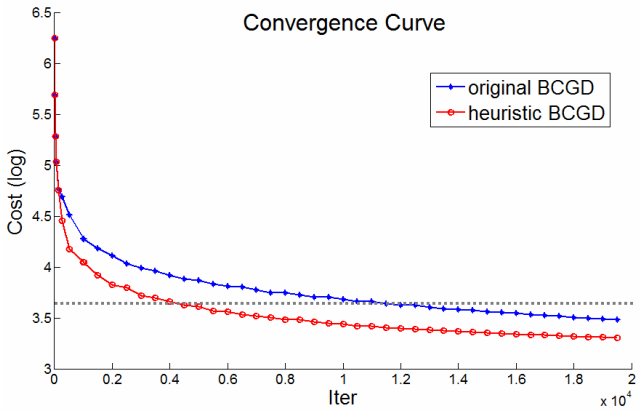

Figure 6: Convergence curve of the heuristic BCGD (red) and the original BCGD (blue).

6.3 Searching Images by MKLSH

To demonstrate the advantage of MKLSH, we conduct experiments on VOC 2007 dataset [7]. The kernels used for constructing the hashing system are listed in Table 3. We construct MKLSH on the training data of VOC 2007 in the way as described in Section 4. Four kernels are used in this experiment. We set H=64, which means a 64-bit hash function for each kernel, and 256-bit hash function for MKLSH. 100 images are chosen from the test set as queries. The returned images are those images in the top 3 nearest buckets of the test queries and they are ranked by their average similarity calculated using the four kernels. We also construct the kernel hashing system with two kernels (Gist and PHOG) and with one kernel (PHOG) for comparison. Since there are many images containing multiple labels, the precision of the selected images is measured as: $precision = n_c / (n_c + n_w)$, where n_c denotes the number of images with at least one class label that is the same as one of the class labels of the query, and n_w denotes the number of images whose class labels are different from the query.

Table 2: Average precision of the 100 queries

Method	Average precision
MKLSH (4 kernels)	0.51
MKLSH (Gist and PHOG)	0.38
KLSH (PHOG)	0.33

The average precision of all the 100 queries is shown in Table 2 and some examples are demonstrated in Figure 7. We can see that the average precision is improved when more kernels are incorporated. For the first two queries in Figure 7, we can see that, when the background is very clean or the object is large enough, the retrieved images are very relevant. Some query that returns bad result is also demonstrated in Figure 7, as shown in the third example. We notice that the returned images look very similar, but the background of the images is much cluttered, which leads to the difficulty of identifying these samples. In this case, a hashing system with class specific kernel weight setting is a better choice. We will investigate this issue in future study.

6.4 Image Annotation

In this sub-section, we conduct image annotation experiments on two datasets, the PASCAL VOC 2007 dataset and the web data

Figure 7: Some results retrieved by MKLSH (4 kernels) on VOC 2007 dataset. The left images are the query images from the test data; the right images are returned results for each query. The dots in red represent the images of the same class label with the queries, and dots in green represents the images with different labels of the queries.

in Table 1. The experiment includes two parts. For the first experiment, we test S^{3}MKL, S^{3}MKL+MKLSH and [31] on VOC 2007 test data, where the labeled data and the unlabeled data are chosen from the VOC 2007 training data. The aim of this part of experiment is to show how S^{3}MKL performs image classification.

For the second experiment, S^{3}MKL and S^{3}MKL+MKLSH are tested on VOC 2007 test data. But the labeled data comes from a part of the training data of VOC 2007, and the unlabeled training data are the web data. The aim of second experiment is to show how S^{3}MKL+MKLSH works under the situation of unconstrained unlabeled dataset. We denote the two methods in part 2 as S^{3}MKL-web and S^{3}MKL+MKLSH-web respectively to make them distinguished from the two in the first experiment. The experimental evaluation is *average precision* (AP) [7].

Previously, various types of features/kernels have been studied and utilized [4, 27, 32, 37, 38]. In general, a complete image feature set should be able to include as more characteristics of color, shape, and texture as possible. We conduct some preliminary experiments on these previously used features/kernels. We choose 8 kinds of features/kernels from them for S^{3}MKL and 4 kinds of features/kernels for MKLSH, which guarantee the good performance of S^{3}MKL as well as the time and storage efficiency of KLSH. The details are listed in Table 3.

For experiments in this Section and Section 6.5, we firstly build a MKLSH system in the way described in Section 4 on the entire image database, which contains the VOC 2007, the web data and the ImageNet data. As in Section 6.3, we set H=64 and the samples of top 3 nearest buckets are retrieved for each query.

We run S^{3}MKL, S^{3}MKL+MKLSH, S^{3}MKL-web, S^{3}MKL+MKLSH-web and method of [31] for different labeled data sampling ratios on VOC 2007 training data. For S^{3}MKL, all the rest of data in VOC training set is used as the unlabeled data. For S^{3}MKL-web, we randomly choose an image subset from the web image category whose class name is identical to the positive class as the unlabeled positive data, and randomly choose a subset from the rest of the web images as the unlabeled negative data. The size of the chosen positive and negative unlabeled subset is twice the size of the labeled data.

For S^{3}MKL+MKLSH-web, we use the labeled data as query, and used the retrieved samples in the web image category whose class names is identical to the positive class as the unlabeled positive samples. We use the returned examples chosen by the MKLSH in the rest of the web image categories as the unlabeled negative samples. For evaluation of [31], we only sample a subset of the training data of VOC 2007 as the labeled data, and use the test data of VOC 2007 as the unlabeled data. We repeat the

experiment for 10 times for each result, and the average precision is demonstrated in Figure 8.

Table 3: The details of kernels used in S^{3}MKL and MKLSH

Kernel used in S^{3}MKL
● 3 level PHOG-180 with Gaussian + χ^2 distance. ● 4×4 Color moment with RBF kernels. ● 2×2 Local binary pattern with Gaussian + χ^2 distance. ● Self Similarity. ● Geometric blur. ● Gist descriptor with Gaussian + χ^2 distance. ● 3 level spatial pyramid kernel on dense visual words with histogram intersection. ● 3 level spatial pyramid kernel on dense color visual words with histogram intersection.
Kernel used in MKLSH
● 3 level PHOG-180 with Gaussian + χ^2 distance. ● 4×4 Color moment with RBF kernels. ● 2×2 Local binary pattern with Gaussian + χ^2 distance. ● Gist descriptor with Gaussian + χ^2 distance.

As can be seen from Figure 8, the performance of S^{3}MKL and S^{3}MKL+MKLSH is higher than others. S^{3}MKL is generally better than S^{3}MKL+MKLSH when the labeled data size is small. The reason is that MKLSH prevents more unlabeled data from being chosen, which reduces the influence of the unlabeled data for model enhancement. When the labeled data size increases, both perform equally since the sample filtering of MKLSH vanishes. However, different results are observed when the web data is used as the candidate unlabeled dataset. S^{3}MKL+MKLSH-web performs better than S^{3}MKL-web for all the labeled sample ratios. When the sample ratio is small, say, 0.1, the difference of S^{3}MKL-web and S^{3}MKL+MKL-web is small since little unlabeled data are chosen. When the sample ratio is large, the hashing system begins to suppress the noise induced by sample selection, especially for the positive unlabeled data. This phenomenon also verifies our previous conjecture that using the real world unlabeled image data without control is very risky. In general, our approaches outperform [31]. When the ratio of the labeled data is 0.6, our methods are comparable with the state-of-the-art [7] which uses all the training set as training data.

6.5 Personalized Image Re-ranking

In this sub-section, we present some qualitative evaluation of personalized re-ranking whose procedure is performed according to the introduction in section 5. We invite 3 non-expert users to take part in the test. The result is shown in Figure 9. The arrows stand for the samples that users choose to annotate. The red arrows point to the images that users mark as relevant (1), and the green ones as irrelevant (0). The dots on the images are the 4 level ground truth annotation provided by the users after the re-ranking procedure. We see from Figure 9 that our method achieves significant ranking improvements. For example, in the first results, the user picks an airplane landing images as his/her favorite images. The returned images contain more airplane landing images. We believe the result provides an alternative view for the study of web image re-ranking.

7. DISCUSSION AND CONCLUSION

We address two critical problems in real world image data mining. The first approach is: how to bridge the *semantic gap* by incorporating multiple features in semi-supervised learning. We propose a scalable semi-supervised multiple kernel learning approach. The second problem is how to choose "*informative*" and "*compact*" subset from the unlabeled samples in order to apply semi-supervised learning for web image applications. To this end, MKLSH is proposed, which combines multiple features/kernels for conducting effective approximated nearest neighbor search. The experimental results prove that our approach is suitable for image annotation and personalized re-ranking.

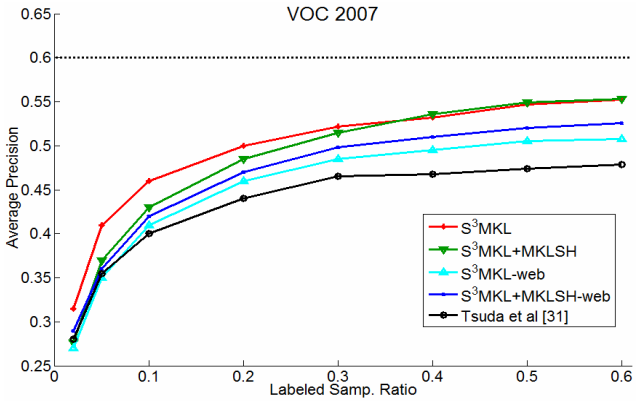

Figure 8: Average precision with respect to the ratio of labeled samples.

Although promising results are obtained in this paper, our method has not solved all the problems. Firstly, our learning system could only work on predefined set of visual concepts. It is not as flexible as automatic photo tagging method [27, 33, 34], which could label the images with any visual concepts appearing online. Future work will investigate how to extend the size of visual concepts by combining our method with automatic data acquisition from the web.

Secondly, from equation (12) we see that the Hessian of C is not guaranteed to be positive definite. Since we use a second order approximation of C, the global convergence is guaranteed only when C is convex or quadratic [30]. Therefore, it will be a little risky when we use a positive diagonal Hessian when its actual value is negative. This situation is more likely to occur when γ is set with a large value, say, $\gamma = 10\times\#$ labeled samples. We observe in several trials that the cost is stuck and vibrating around certain values, but this only lasts for several iterations, for we have guaranteed the descending of the objective function on each Armijo line search. Fortunately, we do not need to impose heavy penalty on the unlabeled data, where a small value of γ as in Section 3.3 is enough for a good solution. Despite this, in future work we will investigate how to make the optimization more robust.

Finally, in MKLSH system we treat different kernels equally. However, we notice that different genres of images have different responses to the features. Therefore, the performance could be further improved by adaptive weight determination. It will be interesting to study hashing with multiple features in the future.

8. ACKNOWLEDGEMENT

This work was supported in part by National Natural Science Foundation of China: 60833006 and 60702035, in part by National Basic Research Program of China (973 Program): 2009CB320906, and in part by Beijing Natural Science Foundation: 4092042. This work was also supported in part by Akiira Media Systems, Inc. for Dr. Qi Tian.

Figure 9: Personalized image re-ranking. The top rows are initial top results. The second rows are selected images from ImageNet. The last row is the top images of re-ranking. (Red: favorite, orange: acceptable, yellow: relevant, green: junk).

9. REFERENCES

[1] F. Bach. Consistency of the Group Lasso and Multiple Kernel Learning. JMLR, 9:1179– 1225, 2008.

[2] F. Bach, G. Lanckriet, and M. Jordan. Multiple Kernel Learning, Conic Duality, and the SMO Algorithm. In ICML, 2004.

[3] A. Blum, T. Mitchell. Combining Labeled and Unlabeled Data with Co-training. In Proceeding of 11th Annual Conference on Computational Learning Theory, pp. 92 – 100, 1998.

[4] L. Cao, J. Luo, F. Liang, and T. S. Huang. Heterogeneous Feature Machine for Visual Recognition. In ICCV, 2009.

[5] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. Locality-sensitive hashing scheme based on *p*-stable distributions. In Proceedings of the 20th annual symposium on Computational Geometry, pp. 253 – 262, 2004.

[6] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, L Fei-Fei. ImageNet: a large-scale hierarchical image database. In CVPR, 2009.

[7] M. Everingham, A. Zisserman, C. K. I. Williams, and L. Van Gool. The pascal visual object classes challenge 2007 results. Technical report, 2007.

[8] J. Fan, Y. Gao, and H. Luo. Multi-level annotation of natural scenes using dominant image components and semantic concepts. In ACM Multimedia, 2004.

[9] M. Gonen, and E. Alpaydin. Localized Multiple Kernel Learning. In ICML, 2008.

[10] Y. Grandvalet, Y. Bengio. Semi-supervised Learning with Entropy Regularization. In NIPS, 2005.

[11] J. Huang and T. Zhang. The benefit of group sparsity. Technical report, arXiv: 0901.2962, 2009.

[12] T. Joachims. Transductive inference for text classification using support vector machines. In ICML, 1999.

[13] B. Kulis, K. Grauman. Kernelized Locality Sensitive Hashing for Scalable Image Search. In ICCV, 2009.

[14] G. Lanckriet, N. Cristianini, P. Bartlett, L. Ghaoui and M. Jordan. Learning the Kernel Matrix with Semi-definite Programming. In JMLR, 5:27– 72, 2004.

[15] G. Mann and A. McCallum. Simple, Robust, Scalable Semi-supervised Learning via Expectation Regularization. In ICML, 2007.

[16] A. McCallum, G. Mann, and G. Druck. Generalized Expectation Criteria. Technical Report 2007-60, University of Massachusetts, Amherst, 2007.

[17] L. Meier, S. van de Geer, and P. Buhlmann. The Group Lasso for Logistic Regression. Journal of the Royal Statistical Society: Series B, 70(1): 53– 71, 2008.

[18] G. Qi, X. Hua, Y. Rui, J. Tang, T. Mei, H. J. Zhang. Correlative Multi-label video annotation. In ACM Multimedia, 2007.

[19] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. SimpleMKL. JMLR, 9:2491– 2521, 2008.

[20] A. Saffari, C. Leistner and H. Bischof. Regularized Multi-Class Semi-Supervised Boosting. In CVPR, 2009.

[21] V. Sindhwani, P. Niyogi, and M. Belkin. Beyond the point cloud: from transductive to semi-supervised learning. In ICML, 2005.

[22] V. Sindhwani, D. S. Rosenberg. An RKHS for multi-view learning and manifold co-regularization. In ICML, 2008.

[23] A. Singh, R. D. Nowak, and X. Zhu. Unlabeled Data: Now it helps, now it doesn't. In NIPS, 2008.

[24] C. G. M. Snoek and M. Worring. Early versus late fusion in semantic video analysis. In ACM Multimedia, 2005.

[25] S. Sonnenburg, G. Ratsch, C. Schafer, and B. Scholkopf. Large Scale Multiple Kernel Learning. JMLR, 7:1531-1565, 2006.

[26] J. Tang, X. Hua, Q. Qi, M. Wang, T. Mei, and X. Wu. Structure Sensitive Manifold Ranking for Video Concept Detection. In ACM multimedia, 2007.

[27] A. Torralba, R. Fergus, W. T. Freeman. 80 Million Tiny Images: A Large Data Set for Nonparametric Object and Scene Recognition. In PAMI, 30 (11): 1958 - 1970, 2008.

[28] A. Torralba, K. P. Murphy, and W. T. Freeeman. Sharing features: efficient boosting procedures for multiclass object detection. In CVPR, 2004.

[29] A. Torralba, B. C. Russell, and J. Yuen. LabelMe: online image annotation and applications. MIT CSAIL Technical Report, 2009.

[30] P. Tseng and S. Yun. A Coordinate Gradient Descent Method for Non-smooth Separable Minimization. Mathematical Programming B, 117 (1-2), 2009

[31] K. Tsuda, H. J. Shin, and B. Scholkopf. Fast Protein Classification with Multiple Networks. In Bioinformatics 21(2): ii59 – ii65, 2005.

[32] M. Varma and D. Ray. Learning the Discriminative Power-invariance Trade-off. In ICCV, 2007.

[33] X. J. Wang, L. Zhang, F. Jing, W. Y. Ma. AnnoSearch: Image Auto-Annotation by Search. In CVPR, 2006.

[34] L. Wu, S. C. H. Hoi, R. Jin, J. Zhu, and N. Yu. Distance Metric Learning from Uncertain Side Information with Application to Automated Photo Tagging. In ACM Multimedia, 2009.

[35] R. Yan, J. Tesic, J. R. Smith. Model-shared Subspace Boosting for Multi-label Classification. In ACM SIGKDD, 2007.

[36] R. Yan, A. Natsev, and M. Campbell. A learning-based hybrid tagging and browsing approach for efficient manual image annotation. In CVPR, 2008

[37] J. Yang, Y. Li, Y. Tian, L. Duan, and W. Gao. Group Sensitive Multiple Kernel Learning for Object Categorization. In ICCV, 2009.

[38] H. Zhang, A. C. Berg, M. Maire and J. Malik. SVM-KNN: Discriminative Nearest Neighbor Classification for Visual Category Recognition. In CVPR, 2006.

[39] X. Zhu. Semi-supervised Learning Literature Survey. Technical Report 1530, University of Wisconsin - Madison, 2006.

[40] X. Zhu, Z. Ghahramani, Semi-supervised learning using Gaussian fields and harmonic functions. In ICML, 2003.