Large-scale Image Annotation by Efficient and Robust Kernel Metric Learning

Zheyun Feng Rong Jin Anil Jain
Department of Computer Science and Engineering, Michigan State University, East Lansing, MI, 48824, USA
{fengzhey, rongjin, jain}@cse.msu.edu

Abstract

One of the key challenges in search-based image annotation models is to define an appropriate similarity measure between images. Many kernel distance metric learning (KML) algorithms have been developed in order to capture the nonlinear relationships between visual features and semantics of the images. One fundamental limitation in applying KML to image annotation problem is that it requires converting image annotations into binary constraints, leading to a significant information loss. In addition, most KML algorithms suffer from high computational cost due to the requirement that the learned matrix has to be positive semi-definitive (PSD). In this paper, we propose a robust kernel metric learning (RKML) algorithm based on the regression technique that is able to directly utilize image annotations. The proposed method is also computationally more efficient because PSD property is automatically ensured by regression. We provide the theoretical guarantee for the proposed algorithm, and verify its efficiency and effectiveness for image annotation by comparing it to both state-of-the-art distance metric learning and image annotation approaches.

1. Introduction

The objective of image annotation is to automatically annotate an image with appropriate keywords, called *tags*, that reflect its visual content. Among various approaches developed for automatic image annotation, search based approaches have been proved to be quite effective, particularly for large image datasets with many keywords [12, 17, 21, 29]. The key idea of search based approach is to annotate a test image $\mathcal{I}$ with the common tags shared by the subset of training images that are visually similar to $\mathcal{I}$.

The crux of search based annotation approaches is to effectively measure the visual similarity between images. *Distance metric learning* (DML) tackles this problem by learning a metric that pulls semantically similar images closer and pushes semantically dissimilar images far apart. Many studies on DML are restricted to learning a linear Mahalanobis distance metric, and fail to capture the nonlinear relationships among the images. Multiple nonlinear DML algorithms have been proposed to overcome this limitation. The key idea is to map data points from the original vector space to a high (or even infinite) dimensional space through a nonlinear mapping, which can be either explicitly constructed using the boosting methods [14, 15, 26], or implicitly derived through kernel functions, referred to as *Kernel Metric Learning* (KML) [5, 7, 28], the focus of this work.

Despite their success, there are several limitations of KML that make it difficult to directly apply them to large-scale image annotation. First, most KML algorithms are developed for binary constraints, *i.e.*, must-links for pairs of "similar" instances and cannot-links for pairs of "dissimilar" instances. In the case of image annotation, it is difficult to construct these binary constraints as two images with different annotations may still share several common keywords, as shown in Table 5, where the 4-th and 5-th images share "palm" and the 1-st and 6-th images share "front". In [32], the authors proposed to generate binary constraints by clustering images using the topic model, as demonstrated in our experiments. However, this solution does not make full use of annotation information. Secondly, the high dimensionality (d) of KML usually leads to a high computational cost in solving the related optimization problems. In particular, to ensure the learned metric to be *Positive Semi-Definite* (PSD), the existing methods need to project the learned matrix into a PSD cone whose computational cost is $O(d^3)$. Finally, the high dimensionality of KML may lead to the overfitting of training data [18]. Although several heuristics [18, 28] were proposed to address this problem, none of them has a solid theoretic support.

In this paper, we propose a regression based approach for KML, termed *Regression based Kernel Metric Learning* (RKML), that explicitly addresses the challenges arising from high dimensionality and limitations of binary constraints. The proposed algorithm directly utilizes the real-valued similarity measure, based on image tags, for learning a distance metric, and therefore do not need to construct the binary constraints. The projection step is avoided by exploiting the special property of regression, and the overfitting risk is alleviated by appropriately regularizing the rank of the learned kernel metric. We demonstrate the robustness of the proposed RKML algorithm to dimensionality by proving the theoretical guarantee of the learned kernel metric. We also verify the efficiency and effectiveness of RKML for search-based image annotation by comparing it to the state-of-the-art approaches for both DML and image annotation over several benchmark datasets.

2. Related Work

In this section we review the related work on image annotation and distance metric learning. Given the rich literature on both subjects, we only discuss the studies closely related to this work, and refer the readers to [12, 21, 33, 35] for the detailed surveys of the two topics.

Image Annotation According to [12], automatic image annotation methods can be categorized into three groups: (i) generative models [3, 10], which are designed to model the joint distribution between tags and visual features, (ii) discriminative models [9, 22] that view image annotation as a classification problems where each keyword is treated as an independent class, and (iii) search based approaches [21, 29]. Recent studies on image annotation show that search based approaches are more effective than both generative and discriminative models. Here, we briefly review the most popular search-based approaches developed for image annotation. TagProp [12] constructs a similarity graph for all images, and propagates the label information via the graph. In [20] a majority voting scheme among the neighboring images is proposed. A sparse coding scheme is proposed in [11] to facilitate label propagation. Conditional Random Field model is adopted in [17] to capture the spatial correlation between annotations of neighboring images.

Distance Metric Learning Many algorithms have been developed to learn a linear DML from pairwise constraints [35], and some of them are designed exclusively for image annotation [17, 32, 34]. Recently, a number of nonlinear DML approaches have been developed to handle nonlinear and multimodal patterns. They are usually classified into two categories, boosting based approaches [14, 15, 26] and kernel based approaches, depending on how the nonlinear mapping is constructed. Many KML algorithms, such as Kernel DCA [16], KLMCA [28] and Kernel ITML [7], directly extend their linear counterparts to KML using the kernel trick. To handle the high dimensionality challenge in KML, a common approach is to apply dimensionality reduction before learning the metric [5, 28]. Although these studies show dimensionality reduction helps alleviate the overfitting risk in KML, no theoretical support is provided.

3. Annotate Images by Kernel Metric Learning

Let $X = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^\top$ be a set of training instances, where $\mathbf{x}_i \in \mathbb{R}^d$ is a d-dimensional instance. Let m be the number of classes, and $Y = (\mathbf{y}_1, \ldots, \mathbf{y}_n)^\top$ be the class assignments of the training instances, where $\mathbf{y}_i \in \{0,1\}^m$ with $y_{i,j} = 1$ if $\mathbf{x}_i$ is assigned to class j and zero, otherwise. In image annotation, each image can be assigned to multiple classes, and thus each vector $\mathbf{y}_i$ may contain multiple ones. Let $\kappa(\mathbf{x}, \mathbf{x}') : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$ be a kernel function, and $\mathcal{H}_\kappa$ be the corresponding *Reproducing Kernel Hilbert Space*. Without a metric, the similarity between two instances $\mathbf{x}_a$ and $\mathbf{x}_b$ could be assessed by the kernel function as $\langle \kappa(\mathbf{x}_a, \cdot), \kappa(\mathbf{x}_b, \cdot) \rangle_{\mathcal{H}_\kappa} = \kappa(\mathbf{x}_a, \mathbf{x}_b)$. Similar to linear DMLs, we modify the similarity measure as $\kappa(\mathbf{x}_a, \mathbf{x}_b) = \langle \kappa(\mathbf{x}_a, \cdot), T[\kappa(\mathbf{x}_b, \cdot)] \rangle_{\mathcal{H}_\kappa}$, where $T : \mathcal{H}_\kappa \mapsto \mathcal{H}_\kappa$ is a linear operator learned from the training examples. The objective of KML is to learn a PSD linear operator T that is consistent with the class assignments of training examples. Note that this is different from similarity learning [4] because we require T to be PSD. In this section, we first present the proposed algorithm (RKML) for KML, followed by its theoretical properties and implementation issues.

3.1. Regression based Kernel Metric Learning

The proposed RKML is a kernel metric learning algorithm based on the regression technique. Let $s_{i,j} \in \mathbb{R}$ be the similarity measure between two images $\mathbf{x}_i$ and $\mathbf{x}_j$ based on their annotations $\mathbf{y}_i$ and $\mathbf{y}_j$. We note that $s_{i,j}$ is a real-valued measurement, which is different from the conventional studies of DML that only consider a binary relationship between two instances. The discussion of $s_{i,j}$ will be delayed to Section 3.3.1. We adopt a regression model to learn a kernel distance metric consistent with the similarity measure $s_{i,j}$ by solving the optimization problem:

$$\widehat{T} = \operatorname*{arg\,min}_{T \succeq 0} \sum_{i,j=1}^{n} \frac{1}{2} \left(s_{i,j} - \langle \kappa(\mathbf{x}_i, \cdot), T[\kappa(\mathbf{x}_j, \cdot)] \rangle_{\mathcal{H}_\kappa}\right)^2 .$$

Following the representer theorem of kernel learning [24], it is sufficient to assume that $\widehat{T}$ only operates in the subspace spanned by $\kappa(\mathbf{x}_i, \cdot), i = 1, \ldots, n$, leading to the following definition for $\widehat{T}$:

$$\widehat{T}[f](\cdot) = \sum_{i,j=1}^{n} \kappa(\mathbf{x}_i, \cdot) A_{i,j} f(\mathbf{x}_j), \tag{1}$$

where $A \in \mathbb{R}^{n \times n}$ is a PSD matrix. Using (1), we can change the optimization problem for $\widehat{T}$ into an optimization problem for A as follows:

$$\min_{A \succeq 0} \quad \mathcal{L}(A) = \tfrac{1}{2} |\mathcal{S} - KAK^\top|_F^2, \tag{2}$$

where $K = [\kappa(\mathbf{x}_i, \mathbf{x}_j)]_{n \times n}$ is the kernel matrix and $\mathcal{S} = [s_{i,j}]_{n \times n}$ includes all the pairwise semantic similarities between any two training images.

It is straightforward to verify that $A = K^{\dagger}\mathcal{S}K^{\dagger}$ is an optimal solution to (2), where $K^{\dagger}$ stands for the pseudo inverse of K. Note that when the semantic similarity matrix $\mathcal{S}$ is PSD, A will also be PSD, thus no additional projection is needed to enforce the linear operator $\widehat{T}$ to be PSD. To avoid overfitting, we replace K with K_r, the best rank r approximation of K, and express A as

$$A = K_r^{-1}\mathcal{S}K_r^{-1}. \tag{3}$$

Evidently, the rank r makes the tradeoff between bias and variance in estimating A: the larger the rank r, the lower the bias and higher the variance. This will become clearer in our theoretical analysis.

Using the learned linear operator $\widehat{T}$, the similarity between any two data instances $\mathbf{x}_a$ and $\mathbf{x}_b$ is given by

$$\kappa(\mathbf{x}_a, \mathbf{x}_b) = \sum_{i,j=1}^{n} \kappa(\mathbf{x}_a, \mathbf{x}_i)\kappa(\mathbf{x}_b, \mathbf{x}_j)A_{i,j} = \Phi(\mathbf{x}_a)^{\top} A\Phi(\mathbf{x}_b),$$

where $\Phi(\mathbf{x}) : \mathbb{R}^d \mapsto \mathbb{R}^n$ is given by $\Phi(\mathbf{x}) = [\kappa(\mathbf{x}, \mathbf{x}_1), \ldots, \kappa(\mathbf{x}, \mathbf{x}_n)]^{\top}$. Thus, the proposed RKML algorithm maps a vector of d dimensions into one with at most m dimensions.

3.2. Theoretical Guarantee of RKML

We will show that the linear operator learned by the proposed algorithm is stochastically consistent, *i.e.*, the linear operator learned from finite samples provides a good approximation to the optimal one learned from an infinite number of samples. To simplify our analysis, we assume that the semantic similarity measure $s_{i,j} = \mathbf{y}_i^{\top}\mathbf{y}_j$ [1].

Define the optimal linear operator T_* that minimizes the expected loss as follows,

$$\min_{T'} \mathrm{E}_{(\mathbf{x}_a, \mathbf{x}_b, \mathbf{y}_a, \mathbf{y}_b)} \left[\left(\mathbf{y}_a^{\top}\mathbf{y}_b - \langle \kappa(\mathbf{x}_a, \cdot), T'[\kappa(\mathbf{x}_b, \cdot)] \rangle_{\mathcal{H}_\kappa} \right)^2 \right].$$

Let $T_*(r)$ be the best rank-r approximation of T_*, and $\widehat{T}$ be the linear operator constructed by A given in (3). We will show that under appropriate conditions, $\|T_* - \widehat{T}\|_2$ is relatively small, where $\|\cdot\|_2$ measures the spectral norm.

Let $g_k(\cdot)$ be the prediction function for the k-th class, *i.e.*, $y_{i,k} = g_k(\mathbf{x}_i)$. We make the following assumption for $g_k(\cdot)$ in our analysis:

$$\mathbf{A1}: \quad g_k(\cdot) \in \mathcal{H}_\kappa, \quad k = 1, \ldots, m.$$

Assumption **A1** essentially assumes that it is possible to accurately learn the prediction function $g_k(\cdot)$ given sufficiently large number of training examples. We also note that assumption **A1** holds if $g_k(\cdot)$ is a smooth function and $\kappa(\cdot, \cdot)$ is a universal kernel [23]. The following theorem shows that under assumption **A1**, with a high probability, the difference between T_* and $\widehat{T}$ will be small, provided n is sufficiently large.

Theorem 1 *Assume* **A1** *holds, and* $\kappa(\mathbf{x}, \mathbf{x}) \leq 1$ *for any* $\mathbf{x}$. *Let* $r < n$ *be a fixed rank, and* $\lambda_1, \ldots, \lambda_n$ *be the eigenvalues of kernel matrix* K/n *ranked in the descending order. For a fixed failure probability* $\delta \in (0, 1)$, *we assume* n *is large enough such that*

$$\lambda_r \geq \lambda_{r+1} + \frac{8}{\sqrt{n}} \ln(1/\delta). \tag{4}$$

Then, with a probability $1 - \delta$, *we have* $\|\widehat{T} - T_*(r)\|_2 \leq \varepsilon$, *where* $\|\cdot\|_2$ *is the spectral norm of a linear operator and* ε *is given by*

$$\varepsilon = \frac{8\ln(1/\delta)/\sqrt{n}}{\lambda_r - \lambda_{r+1} - 8\ln(1/\delta)/\sqrt{n}}.$$

The detailed proof can be found in the supplementary document.

Remark Using the result from Theorem 1, we can analyze how rank r affects $\|\widehat{T} - T_*\|$, the difference between the estimated linear operator and the optimal one. We have

$$\|\widehat{T} - T_*\|_2 \leq \|\widehat{T} - T_*(r)\|_2 + \|T_* - T_*(r)\|_2.$$

As indicated by Theorem 1, $\|\widehat{T} - T_*(r)\|_2 \leq O\left(\frac{1}{\sqrt{n}(\lambda_r - \lambda_{r+1})}\right)$, provided $\lambda_r \geq \lambda_{r+1} + 16/\sqrt{n}\ln(1/\delta)$. By choosing a small r, we would expect a large $\lambda_r - \lambda_{r+1}$ and consequentially a small $\|\widehat{T} - T_*(r)\|_2$, implying a small variance in approximating $T_*(r)$. On the other hand, as the r goes smaller, the $\|T_* - T_*(r)\|_2$ becomes larger, implying a large bias in approximating T_*. Thus, rank r essentially makes the tradeoff between the bias and variance in the estimation of the optimal linear operator T_*.

3.3. Implementation

Regarding implementation, we have two important issues to address: (1) how to appropriately measure the semantic similarity $s_{i,j}$, and (2) how to efficiently compute K_r, the best rank r approximation of K, without computing the full kernel matrix K. The second issue is particularly important for applying the proposed algorithm to large datasets consisted of millions of annotated images. Below, we will discuss these two issues separately.

3.3.1 Computing Semantic Similarity $s_{i,j}$

The most straightforward approach is to measure the semantic similarity as $s_{i,j} = \mathbf{y}_i^{\top}\mathbf{y}_j$. We improve upon this approach by incorporating the log-entropy weighting scheme [19] which has been used for document retrieval. It empirically computes the weighted class assignment $\tilde{y}_{i,j}$ as

$$\tilde{y}_{i,j} = \left(1 + \sum_{k}^{n} \frac{p_{k,j} \log p_{k,j}}{\log n} \right) \cdot \log(y_{i,j} + 1), \tag{5}$$

[1] We note that our analysis can be easily extended to the case when $s_{i,j} = \hat{\mathbf{y}}_i^{\top}\hat{\mathbf{y}}_j$, where $\hat{\mathbf{y}}_i$ is a deterministic transformation of $\mathbf{y}_i$.

where $p_{k,j} = y_{k,j}/\sum_i^n y_{i,j}$. We apply Latent Semantic Analysis (LSA) [19] to further enhance the estimation of semantic similarity, which allows us to remove the noise and correlation in/between annotations. Let $\widetilde{Y} = [\tilde{y}_{i,j}]_{n \times m}$ include the weighted class assignments for all the training images, and $\hat{Y} \in \mathbb{R}^{n \times m'}$ include the first m' singular vectors of $\widetilde{Y}$ with each of its row L_2-normalized by 1. We then compute the semantic similarity as $\mathcal{S} = \hat{Y}\hat{Y}^\top$.

3.3.2 Efficiently Computing K_r by Random Projection

The proposed RKML algorithm requires computing the full kernel matrix K and its top r singular vectors. Since the cost of computing K is $O(n^2)$, it will be expensive when the number of training instances n is large. We can improve the computational efficiency by exploiting the Nyström method [8] to approximate K_r. To this end, we randomly sample $n_s < n$ instances from the collection of n training examples, denoted by $\widehat{\mathbf{x}}_1, \ldots, \widehat{\mathbf{x}}_{n_s}$, then compute the rectangle matrix $K^b \in \mathbb{R}^{n \times n_s}$, and approximate K_r by

$$\tilde{K}_r = K^b[K_r^s]^{-1}[K^b]^\top, \tag{6}$$

where K_r^s is the best rank r approximation of $K^s = [\kappa(\widehat{\mathbf{x}}_i, \widehat{\mathbf{x}}_j)]_{n_s \times n_s}$, the kernel matrix for the sampled data. According to [6], with a high probability, we have

$$\|\tilde{K}_r - K_r\|_2 \leq O(1/\sqrt{n_s}),$$

implying that $\tilde{K}_r$ is an accurate approximation of K_r provided the number of samples n_s is sufficiently large. This is also supported by our empirical study, *i.e.*, kernel matrix K can be well approximated by the Nytröm method when n_s is a few thousands. According to our implementation, we observe that further approximating K^b in (6) to rank r usually yields more accurate prediction for tags. Thus, our final approximation of K_r is given by $\hat{K}_r = K_r^b[K_r^s]^{-1}[K_r^b]^\top$.

4. Experiments

4.1. Datasets and Experimental Setup

	ESP Game	IAPR TC12	Flickr1M
No. of Images	20,768	19,627	999,764
Vocabulary size	268	291	1,000
Tags per image	4.69/15	5.72/23	5.98/202
Images per tag	363/5,059	386/5,534	5,976/76,531

Table 1. Statistics for the datasets used in the experiments. The bottom two rows are given in the format mean/maximum.

Three benchmark datasets for image annotation are used in our study and their statistics are summarized in Table 1. For both ESP Game and IAPR TC12 datasets[2], a bag-of-words model based on densely sampled SIFT descriptors is used to represent the visual content. Flickr1M dataset is comprised of more than one million images crawled from the *Flickr* website that are annotated by more than $700,000$ keywords. Since most keywords are only associated with a small number of images, we only keep the $1,000$ most popular ones. We follow [32, 34] and represent each image with following features: grid color moment, local binary pattern, Gabor wavelet texture, and edge direction histogram.

We randomly select 90% of images from each dataset as training and use the remaining 10% for testing. Given a test image, we first identify the k most visually similar images from the training set using the learned distance metric, and then rank the tags by a majority vote over the k nearest neighbors, where k is chosen by cross-validation.

An RBF kernel is used in our study for all KML algorithms. In RKML we set $n_s = 5,000$ and $m' = 0.38m$ based on our experience, and determine the kernel width and rank r by cross-validation. Parameters for the baselines are directly set to their default values suggested by the original authors. Besides, annotation based on the Euclidean distance, denoted by *Euclid*, is used as a reference in our comparison. Since most DMLs are developed against must-links and cannot-links, we apply the procedure described in [32] to generate the binary constraints by performing a probabilistic clustering over the images based on their tags. More details of this procedure can be referred to [32].

We evaluate the annotation accuracy by the average precision for the top ranked image tags. Following [33, 34], we first compute the precision for each test image by comparing the top 10 annotated tags with the ground truth, and then take the average over the test set. Average recall and F1 score are reported in the supplementary document. The computational efficiency is measured by the running time[3]. Both the mean and standard deviation of evaluation metrics over 20 experimental trials are reported in this paper.

4.2. Comparison with State-of-the-art Distance Metric Learning Algorithms

Comparison to nonlinear DML algorithms. We first compare the proposed RKML[4] algorithm to six state-of-the-art KML methods: (1) Kernel PCA (*KPCA*) [25], (2) Generalized discriminant analysis (*GDA*) [2], (3) Kernel discriminative component analysis (*KDCA*) [16], (4) Kernel local Fisher discriminant analysis (*KLFDA*) [27], (5) Kernel information theoretic based metric learning (*KITML*) [7], and (6) Metric learning for kernel regression (*MLKR*) [31]. We also include three boosting DML algorithms, *i.e.*, Distance Boost (*DBoost*) [14], Kernel Boost (*KBoost*) [15], and metric learning with boosting (*BoostM*) [26], for comparison.

[2] The features of both the datasets could be obtain from [12] http://lear.inrialpes.fr/people/guillaumin/data.php.

[3] All the codes are downloaded from the authors' websites, and run in Matlab on the AMD 2 core @2.7GHz and 64 GB RAM machine.

[4] Without specific notification, RKML stands for the proposed RKML algorithm with Nyström approximation in this section. And the source code of RKML can be found in our website.

Figure 1 shows the average precision for the top t annotated tags obtained by nonlinear DML baselines and the proposed RKML. Surprisingly, we observe that most of the nonlinear DML algorithms are only able to yield performance similar to that based on the Euclidean distance, and more disturbingly, some of the nonlinear DML algorithms even perform significantly worse than the Euclidean distance. On the other hand, the proposed algorithm performs significantly better than the Euclidean distance for almost all cases. Table 5 shows the annotations of exemplar images by different DML algorithms.

We attribute the failure of baseline KML methods mostly to the binary constraints. As described before, all DML algorithms require converting image annotations into binary constraints, which does not make full use of the annotation information. To verify this point, we run RKML with similarity measure $s_{i,j}$ computed from the binary constraints that are generated for the baseline DML algorithms, and denote this method by RKMLH. We observe in Table 2 that RKMLH performs significantly worse than RMKL which directly uses the real-valued similarity measures, confirming the significance of using real-valued similarities for DML in automatic image annotation. More results of RKMLH and a discuss of fairness of binary constraints can be found in the supplementary document.

AP@t(%)	t=1	t=4	t=7	t=10
RKML	55 ± 1.1	41 ± 0.6	33 ± 0.5	28 ± 0.4
RKMLH	49 ± 1.1	36 ± 0.7	29 ± 0.7	24 ± 0.5
RLML	52 ± 1.3	38 ± 0.8	31 ± 0.5	26 ± 0.4

Table 2. Comparison of various extensions of RKML for the top t annotated tags on the IAPR TC12. RKMLH runs RKML using binary constraints, and RLML is the linear version of RKML.

Comparison to linear DML algorithms. We compare our RKML to seven state-of-the-art *linear* DMLs, including Relevant component analysis (*RCA*) [1], Discriminative component analysis (*DCA*) [16], Large margin nearest neighbor classifier (*LMNN*) [30], Local Fisher discriminant analysis (*LFDA*) [27], Information theoretic based metric learning (*ITML*) [7], Probabilistic RCA (*pRCA*) [32], and Logistic discriminant-based metric learning (*LDML*) [13].

Figure 3 shows the average annotation precision for the linear DML baselines. Similar to KML, we observe that even the best linear DML algorithm is only slightly better than the Euclidean distance, while RKML significantly outperforms all linear DML baselines. Again, we believe that the failure of linear DML is likely due to the binary constraints generated from image annotations. Since none of the baseline algorithms, neither linear nor nonlinear DML, is able to significantly outperform the Euclidean distance, it remains unclear if kernel DML is advantageous to a linear DML. To examine this point, we implement the linear version of RKML, denoted by RLML. Table 2 shows the performance of RLML on IAPR TC12. It is clear that RKML significantly outperforms its linear counterpart RLML, verifying the advantage of using kernel in DML. More results for RLML can be found in the supplementary document.

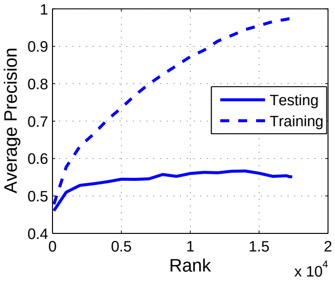


Figure 2. Average precision for the first tag predicted by RMKL using different values of rank r on *IAPR TC12* data. To make the overfitting effect clearer, we turn off the Nyström approximation in this experiment.

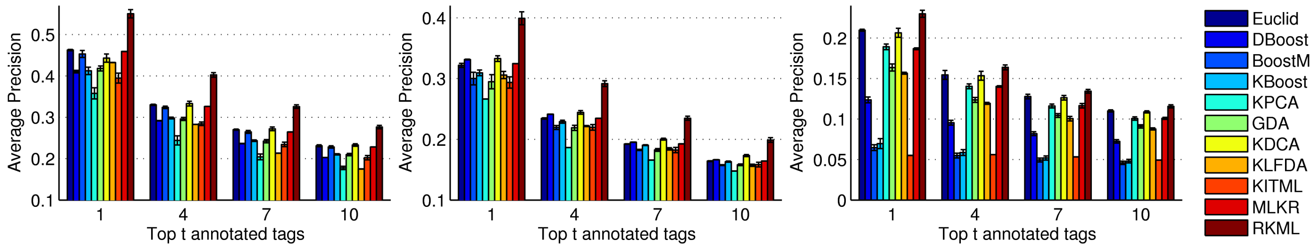
Sensitivity to parameters. We finally examine the role of rank r in the proposed algorithm by evaluating the prediction accuracy with varied r on the IAPRTC 12 dataset for both training and testing images (Figure 2). To make it clear, we turn off the Nyström approximation used by RMKL in this experiment. We observe that while the average accuracy of test images initially improves significantly with increasing rank r, it becomes saturated after certain rank. On the other hand, the prediction accuracy of training data increases almost linearly with respect to the rank, and becomes almost 1 for very large r, a clear indication of overfitting training data. We also examine the sensitivity of the other parameters used by the proposed algorithm (*i.e.*, m', the number of retained eigenvectors of $\tilde{Y}$, and n_s, the number of sampled images used for Nyström approximation). Detailed results of examining parameters m' and n_s can be found in the supplementary document. Overall, we found that our algorithm is insensitive to the values of these parameters over a wide range.

Besides, we also provide the analysis of different semantic similarity measures in the supplementary document.

4.3. Comparison with State-of-the-art Image Annotation Methods

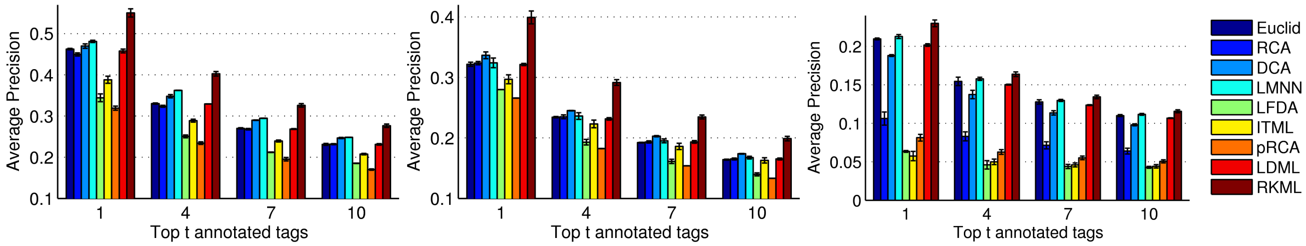
Additionally, we compare RKML algorithm to several state-of-the-art image annotation models including: (1) Two versions of the TagProp method [12], using either rank-based weights (*TP-R*) or distance-based weights (*TP-D*), (2) TagRelevance (*tRel*) [20] based on the idea of neighbor voting, (3) 1-vs-1 SVM classification, using either linear (*SVML*) or RBF kernel (*SVMK*) classifiers[5]. We include *Pop* as a comparison reference which simply ranks tags based on their occurring frequency in the training set.

[5] SVM was unable to perform over *Flickr 1M* due to its large size.



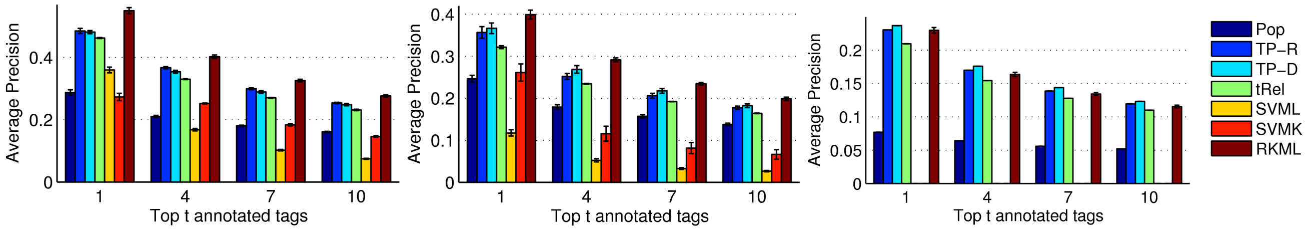
(a) *IAPR TC12* (b) *ESP Game* (c) *Flickr 1M*

Figure 1. Average precision for the top t annotated tags using nonlinear distance metrics.



(a) *IAPR TC12* (b) *ESP Game* (c) *Flickr 1M*

Figure 3. Average precision for the top t annotated tags using linear distance metrics.



(a) *IAPR TC12* (b) *ESP Game* (c) *Flickr 1M*

Figure 4. Annotation performance with different annotation models. SVM method is not included in (c) due to its high computational cost.

TIME	DCA	LMNN	ITML	LDML	DBoost	BoostM	KPCA	GDA	KDCA	KLFDA	KITML	MLKR	RKML
IAPR TC12	1.5e4	1.4e4	4.2e4	4.2e5	1.7e4	1.1e6	2.8e4	4.8e4	2.2e4	8.8e4	5.3e4	2.2e3	4.6e2
ESP Game	2.3e4	1.7e4	5.8e4	5.5e5	4.3e4	1.2e6	3.3e4	5.4e4	3.7e4	3.2e5	6.8e4	3.5e4	1.3e3
Flickr 1M	8.1e4	6.0e4	3.0e4	5.2e5	1.2e4	3.2e5	7.3e3	3.3e4	1.3e5	1.0e5	3.7e6	7.9e3	3.4e3

Table 3. Comparison of running time (s) for several different metric learning algorithms.

Figure 4 shows the comparison of average precision obtained by different image annotation models. It is not surprising to observe that most annotation methods significantly outperform Pop, while the proposed RMKL method outperforms all the state-of-the-art image annotation methods on IAPR TC12 and ESP Game datasets, and only performs slightly worse than TP-D on the Flickr 1M dataset.

TIME	TP-R	TP-D	tRel	SVML	SVMK	RKML
IAPR TC12	9.1e2	4.6e2	1.0e1	2.5e3	4.0e5	4.8e2
ESP Game	2.7e2	1.5e2	1.5e1	1.6e2	8.9e4	1.3e3
Flickr 1M	1.6e5	9.9e4	5.7e3	-	-	3.4e3

Table 4. Running time (s) for image annotation. SVM methods Flickr 1M are not included due to their high computational costs.

4.4. Efficiency Evaluation

Table 3 summarizes the running time of different DML algorithms. We observe that RKML is significantly more efficient than any DML baseline. Table 4 compares the efficiency of different baselines for annotation, where the running time includes the time for both learning a distance metric and predicting image tags. We observe that compared to the other annotation methods, the proposed RKML algorithm is particularly efficient for large datasets (*i.e.*, Flickr 1M), making it suitable for large-scale image annotation.

	Ground	Euclid	DCA	LMNN	LDML	DBoost	BoostM	KBoost	KLDA	KPCA	KDCA	KLFDA	MLKR	TP-R	TP-D	RKML
	fog	**mountain**	**mountain**	**mountain**	**mountain**	**mountain**	**mountain**	**mountain**	**mountain**	tree	**mountain**	**mountain**	**mountain**	**mountain**	**mountain**	**mountain**
	front	**wall**	**wall**	**wall**	**wall**	**wall**	**tourist**	**wall**	**range**	sky	**wall**	**range**	**wall**	man	man	**terrace**
	mountain	**terrace**	tree	**fog**	**terrace**	**terrace**	**front**	**terrace**	**wall**	**front**	**terrace**	**terrace**	**terrace**	**fog**	**wall**	**wall**
	range	**fog**	cloud	**terrace**	woman	cloud	**wall**	cloud	cloud	man	**fog**	**wall**	cloud	**wall**	tree	**range**
	ruin	**range**	**terrace**	**tourist**	cloud	**fog**	woman	**range**	sky	**wall**	**range**	grey	man	tree	people	**ruin**
	terrace	cloud	**tourist**	woman	**range**	**range**	group	sky	**terrace**	house	cloud	**ruin**	woman	**terrace**	woman	**fog**
	tourist	tree	**fog**	tree	**fog**	**ruin**	**range**	tree	**fog**	**mountain**	hill	sky	**range**	slope	**front**	slope
	wall	**ruin**	people	forest	man	sky	man	man	people	people	man	**fog**	tree	summit	**tourist**	sky
	building	**sky**	**sky**	**sky**	**sky**	**sky**	people	**sky**	**sky**	**sky**	**sky**	**sky**	**sky**	**sky**	**sky**	**meadow**
	front	**tree**	**tree**	**tree**	**tree**	sea	**sky**	**tree**	cloud	**tree**	**tree**	**tree**	**tree**	**tree**	**tree**	**sky**
	hill	cloud	**building**	**meadow**	cloud	cloud	man	cloud	house	**wall**	sea	**building**	cloud	**wall**	house	**tree**
	meadow	**building**	man	**building**	**building**	beach	mountain	**building**	**tree**	**front**	beach	people	mountain	**ruin**	cloud	**building**
	ruin	house	house	cloud	house	rock	**tree**	bush	**hill**	mountain	bush	square	house	slope	**front**	**hill**
	sky	**hill**	**front**	bush	people	**meadow**	**front**	**meadow**	mountain	people	cloud	column	man	**meadow**	**meadow**	**wall**
	tree	people	cloud	landscape	bush	**tree**	bush	sea	**building**	man	**meadow**	flag	**meadow**	**building**	man	terrace
	wall	bush	**meadow**	**ruin**	**hill**	coast	rock	house	sea	house	house	**front**	people	house	people	**front**
	bike	**road**	man	sky	**road**	tree	man	sky	tree	tree	sky	sky	**road**	**landscape**	tree	**road**
	cycling	man	wall	bush	man	sky	sky	snow	sky	sky	snow	tree	man	man	sky	sky
	cyclist	**cyclist**	desert	man	**cyclist**	**short**	**rock**	tree	meadow	front	**cycling**	**landscape**	**cyclist**	**grass**	man	**landscape**
	helmet	**jersey**	front	**road**	**helmet**	**jersey**	people	building	man	wall	**cyclist**	**rock**	**helmet**	sea	front	**cyclist**
	jersey	**short**	sky	tree	**jersey**	meadow	tree	front	**cyclist**	man	man	bush	**jersey**	tree	**road**	**short**
	landscape	**bike**	floor	**bike**	**short**	sock	bush	people	**landscape**	people	**short**	building	**short**	cactus	wall	**bike**
	mountain	**cycling**	**road**	car	**cycling**	lawn	**landscape**	cloud	**road**	house	**bike**	cloud	sky	**road**	bush	**cycling**
	road	**helmet**	tree	**cycling**	sky	man	cliff	man	rock	**mountain**	front	front	**cycling**	sky	meadow	**jersey**
	short	car	tourist	**cyclist**	**bike**	spectator	front	street	cloud	woman	**helmet**	**grass**	**bike**	**rock**	people	**helmet**
	door	building	**wall**	building	building	front	building	**house**	building	**sky**	front	**house**	building	**house**	**house**	**door**
	house	front	table	street	table	**house**	**tree**	building	front	**tree**	**house**	**sky**	**house**	**window**	**window**	**house**
	palm	**house**	woman	balcony	**house**	**window**	**sky**	**window**	**window**	front	building	**tree**	table	street	street	**sky**
	roof	table	front	people	front	building	**house**	front	**house**	people	**window**	hill	**wall**	**sky**	**sky**	**window**
	sky	**window**	**window**	square	**wall**	**wall**	street	**door**	**sky**	**house**	**door**	landscape	front	**door**	**tree**	**palm**
	tree	square	classroom	**tree**	woman	**sky**	people	balcony	**wall**	man	**wall**	meadow	man	**tree**	**door**	**tree**
	window	woman	man	**window**	man	column	tower	entrance	**door**	mountain	flag	**roof**	**window**	**palm**	**palm**	building
	wall	**door**	building	front	square	entrance	car	**wall**	column	building	man	snow	woman	man	tile	street
	car	**sky**	**tree**	**tree**	people	**sky**	**sky**	man	**sky**	**sky**	people	fog	people	**tree**	people	**sky**
	fence	people	building	building	**sky**	front	building	sea	people	**tree**	man	**sky**	**sky**	**sky**	**tree**	**spectator**
	grandstand	**tree**	front	**sky**	**tree**	building	**tree**	**sky**	cloud	wall	**tree**	wall	**tree**	front	**sky**	**tree**
	house	man	people	front	**house**	people	people	woman	boat	front	**sky**	man	man	building	man	**fence**
	sky	woman	man	**car**	front	square	man	**tree**	man	man	**fence**	mountain	front	cloud	front	front
	palm	**house**	**sky**	**house**	man	tower	**house**	beach	sea	cloud	woman	slope	**house**	river	**house**	**car**
	spectator	**car**	**car**	**meadow**	square	**tree**	front	cloud	**tree**	mountain	bank	beach	woman	boat	building	**grandstand**
	tree	building	**fence**	**palm**	woman	man	**car**	water	beach	**house**	**car**	bed	square	people	woman	people
	bed	**wall**	**wall**	**wall**	**wall**	**bed**	**wall**	**wall**	**wall**	sky	**wall**	**bed**	**wall**	woman	**wall**	**wall**
	blanket	table	table	woman	table	**wall**	**room**	**room**	**window**	tree	**room**	**wall**	table	**wall**	woman	**bed**
	curtain	**room**	**room**	door	**room**	**room**	**bed**	**bed**	**room**	mountain	**bed**	**room**	**room**	**front**	table	**room**
	front	**window**	woman	table	**front**	**curtain**	table	table	**bed**	cloud	table	table	**front**	door	man	**window**
	room	**curtain**	**front**	man	**window**	**window**	**window**	**window**	**curtain**	**wall**	**curtain**	**wood**	man	man	**room**	**curtain**
	wall	woman	**window**	**room**	**bed**	table	**wood**	**wood**	table	rock	**window**	bedcover	woman	table	**front**	**wood**
	window	**bed**	**bed**	**bed**	woman	**wood**	**curtain**	**curtain**	**wood**	**front**	**wood**	**bedside**	**window**	house	**window**	table
	wood	door	door	building	**curtain**	lamp	lamp	door	**front**	house	door	**curtain**	**bed**	**room**	**bed**	**front**
	building	**sky**	**tree**	**tree**	**sky**	**sky**	**sky**	**sky**	**sky**	**tree**	**sky**	**sky**	**sky**	**sky**	**sky**	**sky**
	cloud	**cloud**	man	**road**	**front**	**cloud**	**tree**	**tree**	**cloud**	man	**tree**	mountain	**front**	**cloud**	**tree**	**tree**
	front	**front**	car	**front**	**cloud**	sea	mountain	**building**	**tree**	wall	cyclist	**tree**	**tree**	**front**	**cloud**	**cloud**
	hill	**tree**	cyclist	man	**tree**	man	**hill**	sea	mountain	**sky**	**front**	desert	**cloud**	**tree**	man	**building**
	meadow	man	cycling	mountain	**road**	landscape	tourist	beach	beach	woman	man	grey	**road**	car	**front**	**meadow**
	monument	**road**	short	**sky**	man	**meadow**	**front**	house	man	**front**	mountain	**hill**	man	park	mountain	**hill**
	sky	mountain	**building**	car	mountain	beach	house	**front**	house	house	people	landscape	mountain	man	**road**	mountain
	tree	car	**sky**	**cloud**	people	**tree**	landscape	city	**meadow**	mountain	**road**	snow	**hill**	shop	house	**front**

Table 5. Examples of annotation results generated by 14 baselines and the proposed RKML. The annotated tags are ranked based on the estimated relevance score in descending order, and the correct ones are highlighted in blue bold font. Note the ground truth annotations in the 2-nd column do not always include all relevant tags (*e.g.*, "people" for the 5-th image), and sometimes contain polysemes (*e.g.*, "palm" for the 4-th and 5-th images) and controversial tags (*e.g.*, "front").

5. Conclusions and Future Work

In this paper, we propose a robust and efficient method for kernel metric learning (KML). The proposed method addresses (i) high computational cost by avoiding the projection into PSD cone, (ii) limitation of binary constraints in tags by adopting a real-valued similarity measure, as well as (iii) the overfitting problem by appropriately regularizing the learned kernel metric. Experiments with large-scale image annotation demonstrate the effectiveness and efficiency of the proposed algorithm by comparing it to the state-of-the-art approaches for DML and image annotation. In the future, we plan to improve the annotation performance by developing a more robust semantic similarity measure.

6. Acknowledgement

This work was partially supported by Army Research Office (W911NF-11-1-0383).

References

[1] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning a mahalanobis metric from equivalence constraints. *JMLR*, 6:937–965, 2005.

[2] G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural Computation*, 12(10):2385–2404, 2000.

[3] G. Carneiro, A. B. Chan, P. J. Moreno, and N. Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. *PAMI*, 29(3):394–410, 2007.

[4] G. Chechik, V. Sharma, U. Shalit, and S. Bengio. Large scale online learning of image similarity through ranking. *JMLR*, 11:1109–1135, 2010.

[5] J. Chen, Z. Zhao, J. Ye, and H. Liu. Nonlinear adaptive distance metric learning for clustering. In *KDD*, 2007.

[6] R. Chitta, R. Jin, and A. K. Jain. Efficient kernel clustering using random fourier features. In *ICDM*, 2012.

[7] J. Davis, B. Kulis, P. Jain, S. Sra, and I. Dhillon. Information-theoretic metric learning. In *ICML*, 2007.

[8] P. Drineas and M. W. Mahoney. On the nyström method for approximating a gram matrix for improved kernel-based learning. *JMLR*, 6:2153–2175, 2005.

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

[10] S. L. Feng, R. Manmatha, and V. Lavrenko. Multiple bernoulli relevance models for image and video annotation. In *CVPR*, 2004.

[11] S. Gao, Z. Wang, L.-T. Chia, and I. W.-H. Tsang. Automatic image tagging via category label and web data. In *ACM Multimedia*, 2010.

[12] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid. TagProp: discriminative metric learning in nearest neighbor models for image auto-annotation. In *ICCV*, 2009.

[13] M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? metric learning approaches for face identification. In *ICCV*, 2009.

[14] T. Hertz, A.-B. Hillel, and D. Weinshall. Boosting margin based distance functions for clustering. In *ICML*, 2004.

[15] T. Hertz, A.-B. Hillel, and D. Weinshall. Learning a kernel function for classification with small training samples. In *ICML*, 2006.

[16] S. Hoi, W. Liu, M. Lyu, and W. Ma. Learning distance metrics with contextual constraints for image retrieval. In *CVPR*, 2006.

[17] C. Ji, X. Zhou, L. Lin, and W. Yang. Labeling images by integrating sparse multiple distance learning and semantic context modeling. In *ECCV*, 2012.

[18] R. Jin, S. Wang, and Y. Zhou. Regularized distance metric learning:theory and algorithm. In *NIPS*. 2009.

[19] Landauer. *Handbook of Latent Semantic Analysis*. Lawrence Erlbaum Associates, 2007.

[20] X. Li, C. G. Snoek, and M. Worring. Learning social tag relevance by neighbor voting. *IEEE Transactions on Multimedia*, 11(7):1310–1322, 2009.

[21] A. Makadia, V. Pavlovic, and S. Kumar. A new baseline for image annotation. In *ECCV*, 2008.

[22] T. Mensink, J. J. Verbeek, and G. Csurka. Learning structured prediction models for interactive image labeling. In *CVPR*, 2011.

[23] C. A. Micchelli, Y. Xu, and H. Zhang. Universal kernels. *JMLR*, 6:2651–2667, 2006.

[24] B. Schölkopf and A. J. Smola. *Learning with kernels: support vector machines, regularization, optimization and beyond*. MIT Press, 2002.

[25] B. Schölkopf, A. J. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[26] C. Shen, J. Kim, L. Wang, and A. van den Hengel. Positive semidefinite metric learning with boosting. In *NIPS*. 2009.

[27] M. Sugiyama. Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis. *JMLR*, 8:1027–1061, 2007.

[28] L. Torresani and K.-c. Lee. Large margin component analysis. In *NIPS*, 2006.

[29] X.-J. Wang, L. Zhang, F. Jing, and W.-Y. Ma. Annosearch: Image auto-annotation by search. In *CVPR*, 2006.

[30] K. Weinberger, J. Blitzer, and L. Saul. Distance metric learning for large margin nearest neighbor classification. In *NIPS*, 2006.

[31] K. Weinberger and G. Tesauro. Metric learning for kernel regression. In *Artificial Intelligence and Statistics*, 2007.

[32] 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.

[33] L. Wu, R. Jin, and A. K. Jain. Tag completion for image retrieval. *PAMI*, 35(3):716–727, 2013.

[34] P. Wu, S. C.-H. Hoi, P. Zhao, and Y. He. Mining social images with distance metric learning for automated image tagging. In *WSDM*, 2011.

[35] L. Yang and R. Jin. Distance metric learning: A comprehensive survey. Technical report, Michigan State Univ., 2009.