Efficient Large-Scale Image Annotation by Probabilistic Collaborative Multi-Label Propagation

Xiangyu Chen[†‡], Yadong Mu[§], Shuicheng Yan[§], and Tat-Seng Chua[†‡]

[†] NUS Graduate School for Integrative Sciences and Engineering,
[‡] School of Computing,
[§] Department of Electrical and Computer Engineering
National University of Singapore, Singapore 117417
{chenxiangyu, elemy, eleyans, chuats}@nus.edu.sg

ABSTRACT

Annotating large-scale image corpus requires huge amount of human efforts and is thus generally unaffordable, which directly motivates recent development of semi-supervised or active annotation methods. In this paper we revisit this notoriously challenging problem and develop a novel multi-label propagation scheme, whereby both the efficacy and accuracy of large-scale image annotation are further enhanced. Our investigation starts from a survey of previous graph propagation based annotation approaches, wherein we analyze their main drawbacks when scaling up to large-scale datasets and handling multi-label setting. Our proposed scheme outperforms the state-of-the-art algorithms by making the following contributions. 1) Unlike previous approaches that propagate over individual label independently, our proposed large-scale multi-label propagation (LSMP) scheme encodes the tag information of an image as a unit label confidence vector, which naturally imposes inter-label constraints and manipulates labels interactively. It then utilizes the probabilistic Kullback-Leibler divergence for problem formulation on multi-label propagation. 2) We perform the multi-label propagation on the so-called hashing-based ℓ_1-graph, which is efficiently derived with Locality Sensitive Hashing approach followed by sparse ℓ_1-graph construction within the individual hashing buckets. 3) An efficient and convergency provable iterative procedure is presented for problem optimization. Extensive experiments on NUS-WIDE dataset (both lite version with $56k$ images and full version with 270k images) well validate the effectiveness and scalability of the proposed approach.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing-indexing methods

General Terms

Algorithms, Performance, Experimentation

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

Keywords

Image Annotation, Collaborative Multi-label Propagation

1. INTRODUCTION

For many applications like image annotation, especially in large-scale setting, annotating training data is often very time-consuming and tedious. Semi-supervised learning (SSL) lends itself as an effective technique, through which users only need to annotate a small amount of image data, and other unlabeled data can work together with these labeled data for learning and inference. In this paper we are particularly interested in efficient graph-based *multi-label* propagation in *large-scale* setting.

It is known that graph is a natural representation for label propagation, wherein each vertex corresponds to a unique image and any edge connecting the two vertices indicates certain relations between the images. Unlike generative modeling methods, graph modeling focuses on non-parametric local structure discovery, rather than a priori probabilistic assumptions. For the transduction task on partially labeled data (known as semi-supervised learning in literature), graph-based methods usually demonstrate the state-of-the-art performance than other SSL algorithms [24].

Generally, there are three crucial subtasks in graph-based algorithms: 1) graph construction; 2) the choice of loss function; and 3) the choice of regularization term. As argued in [23], graph construction is supposed to be more dominating than the other two factors in terms of performance. Unfortunately, it is also the area that is most inadequately studied. In Section 2.2, we propose a novel hashing-based scheme for efficient large-scale graph construction. The solutions to the last two subtasks may affect the final accuracy as well as the proper optimization strategy (thus the convergence speed). As reported in [8], early work on semi-supervised learning can only handle $10^2 \sim 10^4$ unlabeled samples. Consequently, a large number of recent endeavors has been devoted to the scalability to large-scale datasets.

Several recent large scale algorithms (e.g. [11, 6]) plug graph Laplacian based regularizers into *transductive support vector machines* (TSVM) to obtain better transduction capability. The work in [11] solves a graph transduction problem with $650,000$ samples. The whole objective function is optimized via the stochastic gradient descent. While the method in [6] suggests a training method using the *concave-convex procedure* (CCCP), which brings scalability improvement on large-scale dataset. The work in [19]

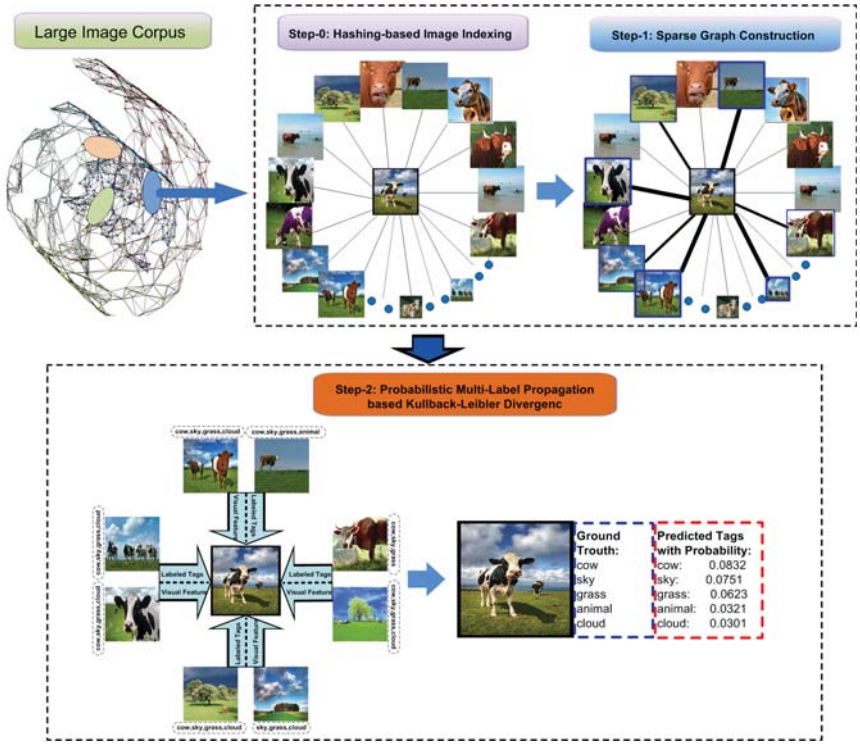

Figure 1: Flowchart of our proposed scheme for multi-label propagation. Step-0 and step-1 are the proposed hashing-based l_1-graph construction scheme, which perform neighborhood selection and weight computation respectively; Step-2 is the probabilistic multi-label propagation based Kullback-Leibler divergence.

solves the largest graph-based problem to date, where there are about $900,000$ samples (including both labeled and unlabeled data). By using a sparsified manifold regularizer and formulating as a center-constrained minimum enclosing ball problem, this method produces sparse solutions with low time and space complexities and can be efficiently solved by the *core vector machine* (CVM).

The seminal work in [17] is most similar to our work in this paper. Unlike previous approaches, this method models the multi-class label confidence vector as a probabilistic distribution, and utilizes the Kullback-Leibler (KL) divergence to gauge the pairwise discrepancy. The underlying philosophy is that such soft regularization term will be less vulnerable to noisy annotation or outliers. Here we adopt the same representation and distance measure, yet in a different scenario (i.e. multi-label image annotation), thus demanding new solution.

Several algorithms were recently proposed to exploit the inter-relations among different labels [12]. For example, Qi et al. [15] proposed a unified Correlative Multi-Label (CML) framework to simultaneously classify labels and model correlations between them. Chen et al. [4] formulated this problem as a sylvester equation, which is similar to [22]. They first constructed two graphs at the sample level and category level associated with a quadratic energy function respectively, and then obtain the labels of the unlabeled images by minimizing the combination of the two energy functions. Liu et. al. [13] utilized constrained nonnegative matrix factorization (CNMF) to optimize the consistency between image similarity and label similarity. Unfortunately, most of the aforementioned algorithms are of high complexity and unsuitable to scale up to the large-scale datasets.

Most existing work in the line of graph-based label propagation suffer (or partially suffer) from these disadvantages: 1) they consider each tag independently when handling multi-label propagation problem, 2) the derived labels for one image are not rankable, and 3) the graph construction process is time-consuming. And most recent large-scale algorithms focus on the single label case, but the scalability to large number of labels is unclear. To address the above issues, we proposed a new large-scale graph-based multi-label propagation approach by minimizing the Kullback-Leibler divergence of the image-wise label confidence vector and its propagated version via the so-called hashing-based ℓ_1-graph, which is efficiently derived with Locality Sensitive Hashing approach followed by sparse ℓ_1-graph construction within the individual hashing buckets. Finally, an efficient and convergency provable iterative procedure is presented for problem optimization. The major contributions of our proposed scheme can be summarized as follows:

- We propose a probabilistic collaborative multi-label propagation formulation for large-scale image annotation, which is founded on Kullback-Leibler divergence based label similarity measurement and scalable ℓ_1-graph construction.

- We also propose a novel hashing-based scheme for efficient large-scale graph construction. *Locality sensitive hashing* [10, 1, 14] is utilized to speed up the candidate selection of similar neighbors for one image, which makes the ℓ_1-graph construction process scalable.

The remainder of this paper is organized as follows. In Section 2, we elaborate on the proposed probabilistic collaborative multi-label propagation (LSMP) algorithm. Section 3 presents analysis on algorithmic complexity and convergence properties. Experimental results on both middle-scale and large-scale image datasets are reported in Section 4. Section 5 concludes this work along with future work discussion.

2. OUR PROPOSED SCHEME

2.1 Scheme Overview

Our proposed large-scale multi-label propagation framework includes three concatenating parts: 1) An efficient k-nearest-neighbor (k-NN) search based on *locality sensitive hashing* (LSH) approach; 2) sparse ℓ_1-graph construction within hashing buckets; and 3) multi-label propagation based on Kullback-Leibler divergence. Figure 1 gives an illustration of the algorithmic pipeline.

2.2 Hashing-based ℓ_1-Graph Construction

The first step of the proposed framework is the construction of an directed weighted graph $\mathcal{G} =< V,\ E >$, where the cardinality of the node set V is $m = l + u$ (denote the labeled and unlabeled data respectively), and the edge set $E \subseteq V \times V$ describes the graph topology. Let V_l and V_u be the sets of labeled and unlabeled vertices respectively. $\mathcal{G}$ can be equivalently represented by a weight matrix $\mathbf{W} = \{w_{ij}\} \in \mathbb{R}^{m \times m}$. To efficiently handle the large-scale data, we enforce the constructed graph to be sparse. The weight between two nodes w_{ij} is nonzero only when $j \in \mathcal{N}_i$, where $\mathcal{N}_i$ denotes the local neighborhood of the i-th image. The graph construction can thus be decomposed into two sub-problems: 1) how to determine the neighborhood of a datum; and 2) how to compute the edge weight w_{ij}.

2.2.1 Neighborhood Selection

For the first problem, the conventional strategies in previous work can be roughly divided into two categories:

- k-nearest-neighbor based neighborhood: w_{ij} is nonzero only if x_j is among the k-nearest neighbors to the i-th datum. Obviously, graphs constructed in this way may ensure a constant vertex degree, avoiding over-dense sub-graphs and isolated vertices.
- ϵ-ball neighborhood: given a pre-specified distance measure between two nodes $d_{\mathcal{G}}(x_i,\ x_j)$ and a threshold ϵ. Any vertex x_j that satisfies $d_{\mathcal{G}}(x_i,\ x_j) \leq \epsilon$ will be incorporated in the neighborhood of the vertex x_i, resulting in nonzero w_{ij}. It is easy to observe that the weight matrix of the constructed graph is symmetric. However, for some vertices beyond a distance from the others, there is probably no edge connecting to other vertices.

Although dominating the graph-based learning literature, the above two schemes are both computation-intensive on large-scale dataset, since a linear scan is required to process a single sample and the overall complexity is $\mathcal{O}(n^2)$ (n is the number of all samples). For a typical image data set to annotate, there are $10^4 \sim 10^5$ images, from each of which high-dimensional features are extracted. A naive implementation based on either of these two schemes usually takes several days to accomplish graph construction, which is definitely unaffordable in terms of efficacy. Instead, in our implementation we use the *locality-sensitive hashing* (LSH) to enhance the efficacy on large-scale data sets.

The basic idea of LSH is to store proximal samples into the same bucket, which greatly saves the retrieval time at the expense of additional storage of hash bits. LSH is a recently proposed hashing algorithm family. The most attractive property of LSH is the theoretic guarantee that the collision probability of two samples (i.e., projected into the same bucket) is proportional to their similarity in feature space. The most popular LSH approach relies on random projection followed by a threshold-based binarization. Formally, given a random projection direction v, the whole dataset is splitted into two half-spaces, according to the rule $h(x_i) = \text{Boolean}(v^T x_i > 0)$. The hash table typically consists of k independent bits, namely the final hash bits are obtained via sequential concatenation $H(x_i) = \langle h_1(x_i), \ldots, h_k(x_i) \rangle$. In the retrieval phase, the k-NN candidate set can be safely confined to be the buckets whose Hamming distances to the query sample are below a pre-specified small threshold. Prior investigation at the theoretic aspect reveals that a sub-linear retrieval complexity is feasible by the LSH method, which is a crucial acceleration for the scenario of large-scale image search. Note that in our implementation, LSH is run for multiple times in all the experiments, and the neighborhoods are the combined to avoid the case of isolated sub-graphs.

2.2.2 Weight Computation

A proper inter-sample similarity definition is the core for graph-based label propagation. The message transmitted from the neighboring vertices with higher weights will be much stronger than the others. Generally, the more similar a sample is to another sample, the stronger the interaction (thus larger weight) exists between them. Below are some popular ways to calculate the pairwise weights:

- *Unweighted k-NN similarity*: The similarity w_{ij} between x_i and x_j is 1 if x_j is among the k-NN of x_i; otherwise 0. For undirected graph, the weight matrix is symmetric and therefore $w_{ij} = w_{ji}$ is enforced.
- *Exponentially weighted similarity*: For all chosen k-NN neighbors, their weights are determined as below:

$$w_{ij} = \exp\left(-\frac{d_{\mathcal{G}}(x_i, x_j)}{\sigma^2}\right), \tag{1}$$

  where $d_{\mathcal{G}}(x_i, x_j)$ is the ground truth distance and σ is a free parameter to control the decay rate.
- *Weighted linear neighborhood similarity* [16, 20]: In this scheme sample x_i is assumed to be linearly reconstructed from its k-NN. The weights are obtained via solving the following optimization problem:

$$\min_{w_{ij}} \parallel x_i - \sum_{j \in \mathcal{N}_i} w_{ij} x_j \parallel^2 . \tag{2}$$

  Typically additional constraints are given to w_{ij}. For example, in [20], the constraints $w_{ij} \geq 0$ and $\sum_j w_{ij} = 1$ are imposed.

In our implementation, we adopt a scheme similar to the idea in [16, 20], based on the linear reconstruction assumption. Moreover, prior work [18] reveals that minimizing the

ℓ_1 norm over the weights is able to suppress the noise contained in data. The constructed graph is non-parametric and is comparably more robust than the other graph construction strategies. Meanwhile, the graph constructed by datum-wise one-vs-all sparse reconstruction of samples can remove considerable label-unrelated links between those semantically unrelated samples to reduce the incorrect information for label propagation.

Suppose we have an over-determined system of linear equations:

$$\begin{bmatrix} x_{i_1} & x_{i_2} & \cdots & x_{i_k} \end{bmatrix} \times \mathbf{w}_i = x_i, \tag{3}$$

where x_i is the feature vector of the i-th image to be reconstructed, $\mathbf{w}_i$ is the vector of the unknown reconstruction coefficients. Let $X \in \mathbb{R}^{d \times k}$ be a data matrix, each column of which corresponds to the feature vector of one of its k-NN. In practice, there are probably noises in the features, and a natural way to recover these elements and provide a robust estimation of $\mathbf{w}_i$ is to formulate $x_i = X\mathbf{w}_i + \xi$, where $\xi \in \mathbb{R}^d$ is the sparse noise term. We can then solve the following l_1-norm minimization problem with respect to both reconstruction coefficients and feature noise:

$$\begin{aligned} \arg_{w,\ \xi} \min \quad & \| \xi \|_1 \\ s.t. \quad & x_i = X\mathbf{w}_i + \xi, \\ & \mathbf{w}_i \geq \mathbf{0}, \quad \| \mathbf{w}_i \|_1 = 1. \end{aligned} \tag{4}$$

This optimization problem is convex and can be transformed into a general linear programming problem. There exists a globally optimal solution, and the optimization can be solved efficiently using many available l_1-norm optimization toolboxes like ℓ_1-MAGIC [3].

2.3 Problem Formulation

Let $M_l = \{x_i, r_i\}_{i=1}^{l}$ be the set of labeled images, where x_i is the feature vector of the i-th image and r_i is a multi-label vector (its entry is set to be 1 if it is assigned with the corresponding label, otherwise 0). Let $M_u = \{x_i\}_{i=l+1}^{l+u}$ be the set of unlabeled images, and $M = \{M_l, M_u\}$ is the entire data set. The graph-based multi-label propagation is intrinsically a transductive learning process, which propagates the labels of M_l to M_u.

For each x_i, we define the probability measure p_i over the measurable space $(Y, \mathcal{Y})$. Here $\mathcal{Y}$ is the σ-field of measurable subsets of Y and $Y \subset \mathbb{N}$ (the set of natural numbers) is the space of classifier outputs. $|Y| = 2$ yields binary classification while $|Y| > 2$ implies multi-label. In this paper, we focus on the multi-label case. Hereafter, we use p_i and r_i for the i-th image, both of which are subject to the multinomial distributions, and $p_i(y)$ is the probability that x_i belongs to class y. As mentioned above, $\{r_j, j \in V_l\}$ encodes the supervision information of the labeled data. If it is assigned a unique label by the annotator, r_j becomes the so-called "one-hot" vector (only the corresponding entry is 1, the rest is 0). In case being associated with multiple labels, r_j is represented to be a probabilistic distribution with multiple non-zero entries.

We propose the following criterion to guide the propagation of the supervision information, which is based on the concept of KL divergence defined on two distributions:

$$D_1(p) = \sum_{i=1}^{l} D_{KL}\big(r_i \parallel p_i\big) + \mu \sum_{i=1}^{m} D_{KL}\big(p_i \parallel \sum_{j \in N(i)} w_{ij} p_j\big), \tag{5}$$

and the optimal solution $p^* = \arg_p \min D_1(p)$.

Here $D_{KL}(r_i \parallel p_i)$ denotes the KL divergence between r_i and p_i, whose formal definition for the discrete case is expressed as $D_{KL}(r_i \parallel p_i) = \sum_y r_i(y) \log \frac{r_i(y)}{p_i(y)}$. The first term in $D_1(p)$ trigger a heavy penalty if the estimated value p_i deviates from the pre-specified r_i. Note that unlike most traditional approaches, there is no constraint for the rigid equivalence between p_i and r_i. Such a relaxation is able to mitigate the bad effect of noisy annotations. The second term of D_1 stems from the assumption that p_i can be linearly reconstructed from the estimations of its neighbors, thus penalizing the inconsistency between the p_i and its neighborhood estimation. Unlike previous works [20] using squared-error (optimal under a Gaussian loss assumption), the adopted KL-based loss penalizes *relative error* rather than *absolute error* in the squared-error case. In other words, they can be regarded as the regularization terms from prior supervision and local coherence respectively. μ is a free parameter to balance these two terms.

If $\mu, w_{ij} \geq 0$, then $D_1(p)$ is convex (the proof is given in Appendix I). Since no closed-form solution is feasible, standard numerical optimization approaches such as interior point methods (IPM) or method of multipliers (MOM) can be used to solve the problem. However, most of these approaches guarantee global optima yet are tricky to implement (e.g., an implementation of MOM to solve this problem would have seven extraneous parameters) [17]. Instead, we utilize a simple alternating minimization method in this work.

Alternating minimization is an effective strategy to optimize functions of the form $f(x,\ y)$ where x, y are two sets of variables. In many cases, simultaneous optimizing over x and y is computationally intractable or unstable, while optimizing over one set of variables with the other fixed is relatively easier. Formally, a typical alternating minimization loops over two sub-problems, i.e., $x^{(t)} = \arg_x \min f(x,\ y^{(t-1)})$ and $y^{(t)} = \arg_y \min f(x^{(t)}, y)$. An example for alternating optimization is the well-known Expectation-Maximization (EM) algorithm. Note that D_1 in Equation (5) is not amenable to alternating optimization. We further propose a modified version by introducing a new group of variables $\{q_i\}$, which is shown as below:

$$\begin{aligned} D_2(p, q) \quad = \quad & \sum_{i=1}^{l} D_{KL}(r_i \parallel q_i) + \mu \sum_{i=1}^{m} D_{KL}(p_i \parallel \sum_{j \in \mathcal{N}(i)} w_{ij} q_j) \\ & + \eta \sum_{i=1}^{m} D_{KL}(p_i \parallel q_i). \end{aligned} \tag{6}$$

In the above, a third measure q_i is introduced to decouple the original term $\mu \sum_{i=1}^{m} D_{KL}\big(p_i \parallel \sum_{j \in N(i)} w_{ij} p_j\big)$. q_i can actually be regarded as a relaxed version of p_i. To enforce consistency between them, the third term $\sum_{i=1}^{m} D_{KL}(p_i \parallel q_i)$ is incorporated.

2.4 Part I: Optimize p_i with q_i Fixed

With $\{q_i,\ i = 1 \ldots m\}$ fixed, the optimization problem is reduced to the following form:

$$\begin{aligned} p^* \quad = \quad & \arg_p \min D_2(p, q) \\ s.t. \quad & \sum_y p_i(y) = 1,\ p_i \geq \mathbf{0},\ \ \forall\ i. \end{aligned} \tag{7}$$

The above constrained optimization problem can be easily

transformed into an unconstrained one using the Lagrange multiplier:

$$p^* = \arg_p \min D_2(p,q) + \sum_{i=1}^{m} \lambda_i (1 - \sum_y p_i(y)). \quad (8)$$

For brevity, let $\mathcal{L}_p \triangleq D_2(p,q) + \sum_{i=1}^{m} \lambda_i (1 - \sum_y p_i(y))$. Recall that any locally optimal solutions should be subject to the zero first-order derivative, i.e.,

$$\begin{aligned} \frac{\partial \mathcal{L}_p}{\partial p_i(y)} &= \mu\big(\log p_i(y) + 1 - \log \sum_{j \in \mathcal{N}(i)} w_{ij} q_j(y)\big) \\ &\quad + \eta\big(\log p_i(y) + 1 - \log q_i(y)\big) - \lambda_i \\ &= 0. \end{aligned} \quad (9)$$

From Equation (9), it is easily verified that (let $\gamma = \mu + \eta$):

$$p_i(y) = \exp\left(\frac{\mu \log \sum_{j \in \mathcal{N}(i)} w_{ij} q_j(y) + \eta \log q_i(y) - \gamma + \lambda_i}{\gamma}\right).$$

Recall that λ_i is the Lagrange coefficient for the i-th sample and unknown. Based on the fact $\sum_y p_i(y) = 1$, λ_i can be eliminated and finally we obtain the updating rule:

$$p_i(y) = \frac{\exp\left(\frac{\mu}{\gamma} \log \sum_{j \in \mathcal{N}(i)} w_{ij} q_j(y)) + \frac{\eta}{\gamma} \log q_i(y)\right)}{\sum_y \exp\left(\frac{\mu}{\gamma} \log \sum_{j \in \mathcal{N}(i)} w_{ij} q_j(y) + \frac{\eta}{\gamma} \log q_i(y)\right)}. \quad (10)$$

2.5 Part II: Optimize q_i with p_i Fixed

The other step of the proposed alternating optimization is to update q_i with p_i fixed. Unfortunately, it proves that the same trick used in subsection 2.4 cannot be applied to the optimization of q_i, due to the highly non-linear term $\log\left(\sum_{j \in \mathcal{N}_i} w_{ij} q_j(y)\right)$. To ensure that q_i is still a valid probability vector after updating, we set the updating rule as:

$$q_i^{new} = q_i^{old} + U\boldsymbol{h}, \quad (11)$$

where the column vector of matrix $U \in \mathbb{R}^{d \times (d-1)}$ is constrained to be summed 0. Denote $\boldsymbol{e}$ to be a column vector with its all entries equal to 1, then we have $\boldsymbol{e}^T U = \boldsymbol{0}$. An alternative view of this relationship is that U is the complementary subspace of the one spanned by $\frac{1}{\sqrt{n}}\boldsymbol{e}$, thus $UU^T = I - \frac{1}{n}\boldsymbol{e}\boldsymbol{e}^T$ also holds.

Vector $\boldsymbol{h}$ in each iteration should be carefully chosen so that the updated value of q_i^{new} results in a non-trivial decrease of the overall objective function. Denote $\mathcal{L}_q \triangleq D_2(p,q)$ and the value of q_i at the t-th iteration as $q_i^{(t)}$, we have

$$\nabla \mathcal{L}_h(q_i^{(t)}) \triangleq \frac{\partial \mathcal{L}_q(q_i^{(t)} + U^T\boldsymbol{h})}{\partial \boldsymbol{h}} = U^T \frac{\partial \mathcal{L}_q}{\partial q_i}\Big|_{q_i = q_i^{(t)}}. \quad (12)$$

Note that in each iteration $\boldsymbol{h}$ is typically initialized as 0, thus $h = -\alpha \nabla \mathcal{L}_h(q_i^{(t)})$ is a candidate descent direction (α is a parameter to control the step size). By substituting it into Equation (11), we obtain the following updating rule:

$$\begin{aligned} q_i^{(t+1)} &= q_i^{(t)} - \alpha UU^T \frac{\partial \mathcal{L}_q}{\partial q_i}\Big|_{q_i = q_i^{(t)}} \\ &= q_i^{(t)} - \alpha(I - \frac{1}{n}\boldsymbol{e}\boldsymbol{e}^T)\frac{\partial \mathcal{L}_q}{\partial q_i}\Big|_{q_i = q_i^{(t)}}. \end{aligned} \quad (13)$$

Algorithm 1 Probabilistic Collaborative Multi-Label Propagation

1: **Input:** An directed weighted sparse graph $\mathcal{G} =< V, E >$ of the whole image dataset $M = \{M_l, M_u\}$, where $M_l = \{x_i, r_i\}_{i=1}^{l}$ is the labeled image set and $M_u = \{x_i\}_{i=l+1}^{l+u}$ is the set of unlabeled images. x_i is the feature vector of the i-th image and r_i is a multi-label confidence vector for x_i.
2: **Output:** The convergent probability measures p_i and q_i.
3: **Initialization:** Randomly initialize $\{p_i \geq 0, \sum_y p_i(y) = 1\}$ and $\{q_i \geq 0, \sum_y q_i(y) = 1\}$.
4: **for** p_i and q_i are not convergent **do**
5: **Optimize** p_i **with** q_i **Fixed:**
$$p_i(y) = \frac{\exp\left(\frac{\mu}{\gamma} \log \sum_{j \in \mathcal{N}(i)} w_{ij} q_j(y)) + \frac{\eta}{\gamma} \log q_i(y)\right)}{\sum_y \exp\left(\frac{\mu}{\gamma} \log \sum_{j \in \mathcal{N}(i)} w_{ij} q_j(y) + \frac{\eta}{\gamma} \log q_i(y)\right)}.$$
6: **Optimize** q_i **with** p_i **Fixed:**
$q_i^{(t+1)} = q_i^{(t)} - \alpha(I - \frac{1}{n}\boldsymbol{e}\boldsymbol{e}^T)\frac{\partial \mathcal{L}_q}{\partial q_i}$, where α lies in the range defined in Equation (16).
7: **end for**

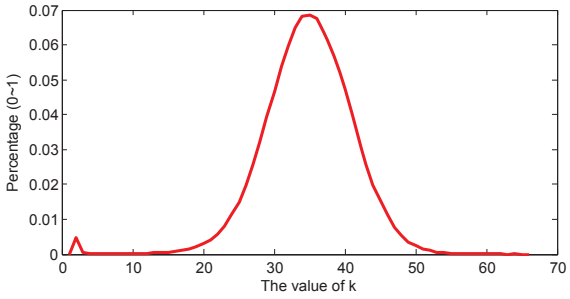

Figure 2: The distribution of the number of nearest neighbors (denote as k) in our proposed LSMP.

In this way, the pursuit of the descent direction with respect to q_i is transformed into an equivalent problem taking $\boldsymbol{h}$ as variable, which is further solved by calculating $\frac{\partial \mathcal{L}_q}{\partial q_i}$. For completeness, we list the concrete value of an entry of $\frac{\partial \mathcal{L}_q}{\partial q_i}$:

$$\frac{\partial \mathcal{L}_q}{\partial q_i(y)} = -\frac{r_i(y)}{q_i(y)} - \mu \sum_{\forall k:\ i \in \mathcal{N}_k} \frac{w_{ki} p_k(y)}{\sum_{j \in \mathcal{N}_k} w_{kj} q_j(y)} - \eta \frac{p_i(y)}{q_i(y)}. \quad (14)$$

One practical issue is the feasible region of parameter α. An arbitrary α probably cannot ensure that the updated $p_i^{(t+1)}$ in Equation (13) stays within the range $[0,1]$. A proper value of α should ensure:

$$\boldsymbol{0} \leq q_i - \alpha UU^T \frac{\partial \mathcal{L}_q}{\partial q_i}\Big|_{q_i = q_i^{(t)}} \leq \boldsymbol{1}. \quad (15)$$

Denote $\boldsymbol{v} = UU^T \frac{\partial \mathcal{L}_q}{\partial q_i}\big|_{q_i = q_i^{(t)}}$. It is easy to verify that

$$0 \leq \alpha \leq \min\left\{\max\left\{\frac{q_i(y)}{\boldsymbol{v}(y)}, \frac{q_i(y) - 1}{\boldsymbol{v}(y)}, \epsilon\right\}\right\}. \quad (16)$$

In practice, α can be adaptively determined from $q_i^{(t)}$. The whole process of optimization is illustrated in Algorithm 1. The resultant p_i is adopted to infer the image tags, as it connects both r_i and q_i.

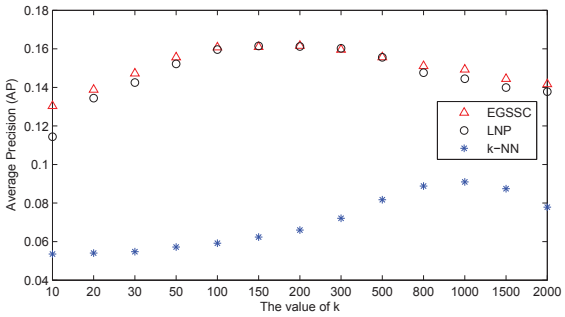

Figure 3: The performance of three baseline algorithms with respect to the number of nearest neighbors (denote as k).

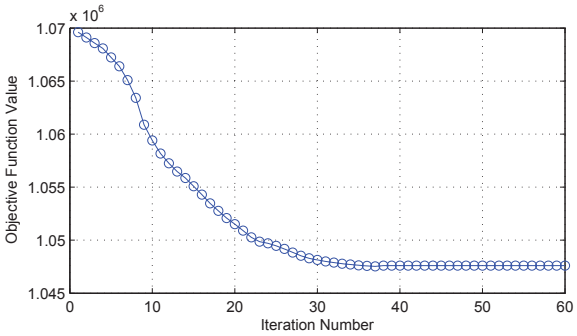

Figure 4: Convergence curve of our proposed Algorithm on NUS-WIDE dataset.

3. ALGORITHMIC ANALYSIS

3.1 Computational Complexity

Overall speaking, the computational complexity of the proposed algorithm consists of two components: the cost of hashing-based ℓ_1-graph construction, and the cost of KL-based label propagation. The efficacy of traditional graph construction as in [21, 18] hinges on the complexity of k-NN retrieval, which is typically $\mathcal{O}(n^2)$ (n is the number of images) for a naive linear-scan implementation. Our proposed LSH-based scheme guarantees a sublinear complexity by aggregating visually similar images into the same buckets, greatly reducing the cardinality of the set of candidate neighbors. Formally, recent work points out the lower bound of LSH is only slightly high than $\mathcal{O}(n\log(n))$, which drastically reduces the computational overhead of graph construction compared with traditional $\mathcal{O}(n^2)$ complexity.

On the other hand, for our proposed KL-guided label propagation procedure, it has $\mathcal{O}(n\ k\ l)$ computation in each iteration, where k denotes the averaged number of nearest neighbors for a graph vertex and l is the total number of labels. Actually, most label propagation methods based on local confidence exchange have the same complexity. The consumed time in real calculation mainly hinges on the value of k. In Figure 2 we plot the distribution of k obtained via the proposed ℓ_1-regularized weight computation, which reaches its peek value around $k = 35$. This small k value indicates that ℓ_1 penalty term is able to select much compacter reconstruction basis for a vertex. In contrast, to obtain nearly optimal performance, previous works usually take $k > 100$ (see Figure 3). In implementation, we find that the subtle reduce of k results in a drastic reduce of the running time (see more details in the experimental section).

3.2 Algorithmic Convergence

The above two updating procedures are iterated until converged. For the experiments on NUS-WIDE dataset, generally about 50 iterations are required for the convergency of the solution. An exemplar convergency curve is shown in Figure 4.

4. EXPERIMENTS

To validate the effectiveness of our proposed approach on large-scale multi-label datasets, we conduct extensive experiments on the real-world image dataset NUS-WIDE [5], which contains 269,648 images accompanied with totally 5,018 unique tags. Images in this dataset are crawled from the photo sharing website Flickr by using its public API. The underlying image diversity and complexity make it a good testbed for large-scale image annotation experiments. Moreover, a subset of NUS-WIDE (known as NUS-WIDE-Lite) obtained after noisy tag removal is also publicly available. We provide quantitative study on both the lite dataset and the full NUS-WIDE dataset, with an emphasis on the comparison with five state-of-the-art related algorithms in terms of accuracy and computational cost.

4.1 Datasets

NUS-WIDE [5]: The dataset contains 269,648 images and the associated 5,018 tags. For evaluation, we construct two image pools from the whole dataset: the pool of labeled images is comprised of 161,789 images whilst the rest are used for the pool of unlabeled images. For each image, an 81-D label vector is maintained to indicate its relationship to 81 distinct concepts (tightly related to tags yet relatively high-level). Moreover, to testify the performance stability of various algorithms, we vary the percentage of labeled images selected from the labeled image pool (in implementation it is varying from 10% to 100% increased by a step of 10%. We introduce the variable $\tau \in [0,\ 1]$ for it). The sampled labeled images are then amalgamated with the whole set of unlabeled images (107,859 in all). We extract multiple types of local visual features from the images (225-D block-wise color moments, 128-D wavelet texture and 75-D edge direction histogram).

NUS-WIDE-Lite: As stated above, this dataset is a lite version of the whole NUS-WIDE database. It consists of 55,615 images randomly selected from the NUS-WIDE dataset. And the labels of each image are also like those of NUS-WIDE, an 81-D label vector is set to indicate its relationship to 81 distinct concepts. As done on NUS-WIDE, three types of local visual features are also extracted for this dataset. We randomly select about half of the images as labeled and the rest to be unlabeled. Again, we use the same sampling strategy on the labeled set to perform the stability test.

4.2 Evaluation Criteria and Baselines

In the experiments, five baseline algorithms as shown in Table 1 are evaluated for comparative study. Amongst them, the *support vector machines* (SVM) is originally developed to solve binary-class or multi-class classification problem. Here we use its multi-class version by adopting the one-vs-one method. The selected baselines includes several state-of-the-art algorithms for semi-supervised learning. The *lin-*

Table 1: The Baseline Algorithms.

Name	Methods
KNN	k-Nearest Neighbors [9]
SVM	Support Vector Machine [6]
LNP	Linear Neighborhood Propagation [20]
EGSSC	Entropic Graph Semi-Supervised Classification [17]
SGSSL	Sparse Graph-based Semi-supervised Learning [18]

ear neighborhood propagation (LNP) [20] bases on a linear-construction criterion to calculate the edge weights of the graph, and disseminates the supervision information by a local propagation and updating process. The EGSSC [17] is an entopic graph-regularized semi-supervised classification method, which is based on minimizing a Kullback-Leibler divergence on the graph built from k-NN Gaussian similarity as introduced in Sub-section 2.2.1 and 2.2.2. The SGSSL [18] is a sparse graph-based method for semi-supervised learning by harnessing the labeled and unlabeled data simultaneously, which considers each label independently.

The criteria to compare the performance include *Average Precision* (AP) for each label (or concept) and *Mean Average Precision* (MAP) for all labels. The former is a well-known gauge widely used in the field of image retrieval, whilst the latter is developed to handle the multi-class or multi-label cases. For example, in our application MAP is obtained by averaging the APs on 81 concepts. All experiments are conducted on a common desktop PC equipped with Intel dual-core CPU (frequency: 3.0 GHz) and 32G bytes physical memory.

For the experiments on NUS-WIDE-Lite, the proposed method is compared with all the five baseline algorithms. While on the NUS-WIDE, the results from SGSSL is not reported due to its incapability to handle dataset in such large scale.

4.3 Experiment-I: NUS-WIDE-LITE (56k)

In this experiment, we compare the proposed algorithm with five baseline algorithms. The results with varying numbers of labeled images (controlled by the parameter τ) are presented in Figure 5. Below are the parameters and the adopted values for each method: for KNN, there is only one parameter k for tuning, which stands for the number of nearest neighbors and is trivially set as 500. For SVM algorithm, we adopt the RBF kernel. For its two parameters γ and C, we set $\gamma = 0.6$ and $C = 1$ in experiments after fine tuning. For LNP algorithm, one parameter α is adjusted, which is the fraction of label information that each image receives from its neighbors. The optimal value is $\alpha = 0.95$ in our experiments. There are three parameters μ, ν and β in EGSSC, where μ and ν are used for weighting the Kullback-Leibler divergence term and Shannon entropy term respectively and β ensures the convergence of the two similar probability measures. The optimal values are set as $\mu = 0.1$, $\nu = 1$ and $\beta = 2$ here. For our proposed algorithm, we set $\mu = 10$ and $\eta = 5$. MAP of these six methods is illustrated in Figure 6.

Our observations from Figure 5 are described as follows:

- Our proposed algorithm LSMP outperforms the other baseline algorithms significantly when selecting different proportions of labeled set. For example, with 10 percent of labeled images selected, LSMP has an improvement 16.6% over SGSSL, 58.5% over EGSSC, 107.6% over LNP, 137.2% over SVM, and 154.5% over KNN. The improvement is supposed to stem from the fact that our proposed algorithm encodes the label information of each image as a unit confidence vector, which imposes extra inter-label constraints. In contrast, other methods either consider the visual similarity graph only, or considers each label independently.

- With the increasing number of labeled images, the performances of all algorithms consistently increase. When $\tau \leq 0.6$, the algorithm SGSSL outperforms the other two state-of-art algorithms LNP and EGSSC significantly. However, when $\tau > 0.6$, the improvement of SGSSL over the others is lower. The proposed method keeps higher MAP value than other five methods over all values of τ.

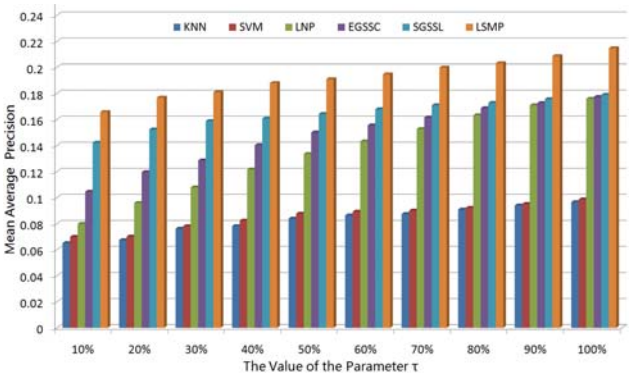

Figure 5: The results of the comparison of LSMP and the five baselines with varying parameter τ on NUS-WIDE-Lite dataset.

Recall that the proposed algorithm is a probabilistic collaborative multi-label propagation algorithm, wherein $p_i(y)$ expresses the probability for the i-th image to be associated with the y-th label. A direct application for this probabilistic implication is the tag ranking task. Some exemplar results of tag ranking are shown in Figure 7.

4.4 Experiment-II: NUS-WIDE (270k)

In this experiment, we compare the proposed LSMP algorithm with four state-of-the-art algorithms on the large-scale NUS-WIDE dataset for multi-label image annotation. As in previous experiments, we modulate the parameter τ to vary the percentage of the labeled images used in the experiments and carefully tune the optimal parameters in each method for fair comparison. For KNN, the optimal value is $k = 1000$. For SVM algorithm, we set $\lambda = 0.8$ and $C = 2$. For LNP method, the optimal value is $\alpha = 0.98$. In the experiment of EGSSC, the best values are $\mu = 0.5$, $\nu = 1$ and $\beta = 1$. For our proposed LSMP algorithm, $\mu = 15$ and $\eta = 8$. The results of all algorithms are shown in Figure 8 and the results with respect to each individual concept are presented in Figure 9. From Figure 9, we can observe that

- On the large-scale real-world image dataset, the proposed algorithm outperforms other algorithms significantly at all values of τ. For example, when $\tau = 0.1$, LSMP has an improvement 53.5% over EGSSC, 112.6% over LNP, 197.2% over SVM, and 220.5% over

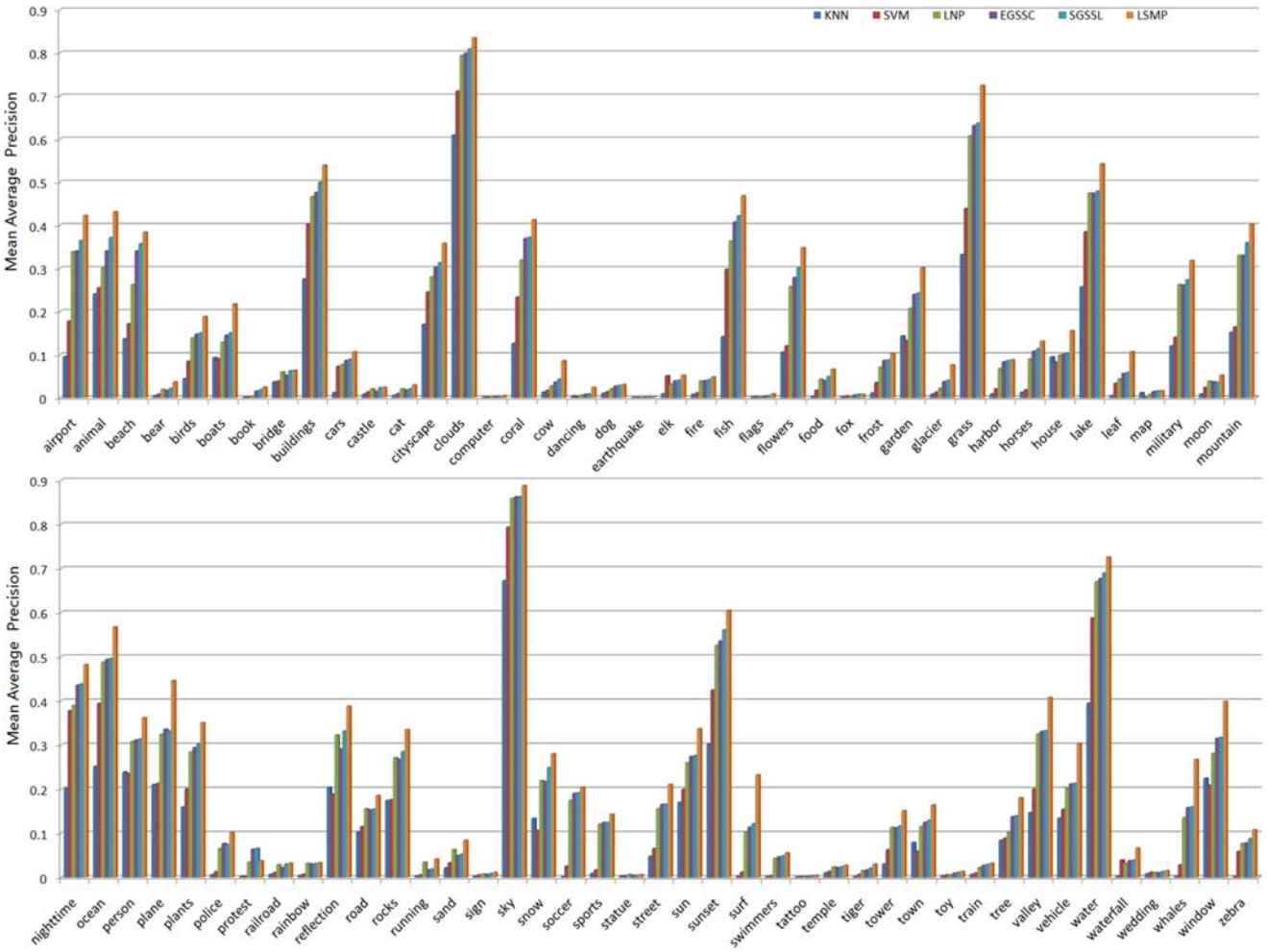

Figure 6: The comparison of APs for the 81 concepts using six methods with $\tau = 1$.

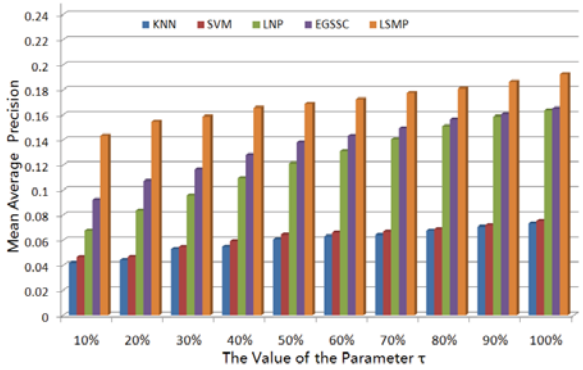

Figure 8: The results of the comparison of LSMP and the four baselines with varying parameter τ on NUS-WIDE.

KNN. Compared with the performance on NUS-WIDE-Lite, the best performance of LSMP in NUS-WIDE is 0.193, which is smaller than the MAP value in the Lite version. The performance degradation is primarily attributed to the increase of data scale (the size of labeled image pool in NUS-WIDE is 170K, while for the Lite version it is only 27K).

- With the increasing parameter τ, the performances of all algorithms also increase. When $\tau \leq 0.6$, the algorithm EGSSC outperforms LNP significantly, but for $\tau > 0.6$, the improvement of EGSSC than LNP is negligible. The proposed method LSMP also keeps higher MAP value than all baselines over all feasible values of τ similar to the case on NUS-WIDE-LITE, which validates the robustness of our proposed algorithm.

We also provide the recorded running time for different algorithms on NUS-WIDE, as shown in Table 2. A salient efficacy improvement can be observed from our proposed method.

5. CONCLUSION

In this paper we propose and validate an efficient large-scale image annotation method. Our contributions lie in both the hashing-accelerated ℓ_1-graph construction, and KL-divergence oriented soft loss function and regularization term in graph-based modeling. The optimization framework utilizes the inter-label relationship and finally returns a probabilistic label vector for each image, which is more robust to noises and can be used for tag ranking. The proposed algorithm is experimented on several publicly-available image benchmarks built for multi-label annotation, including the

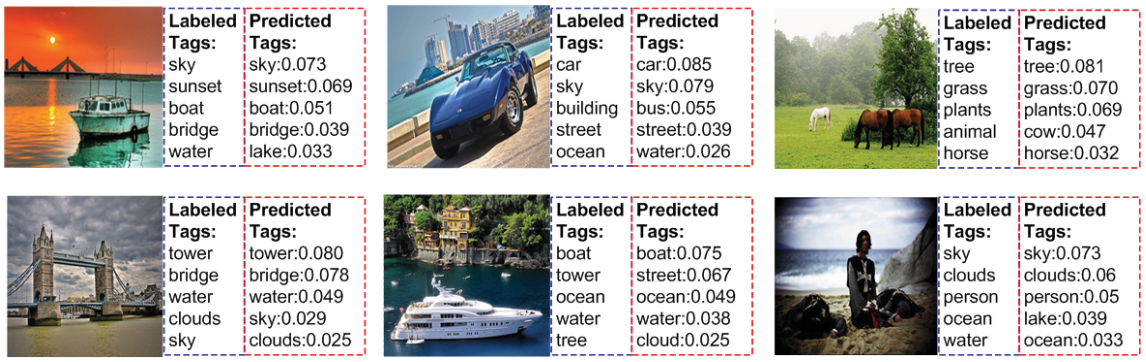

Figure 7: The tags ranking results of LSMP in NUS-WIDE-LITE.

Table 2: Executing time (unit: hours) comparison of different algorithmson the NUS-WIDE dataset.

Algorithms	Graph Construction Time	Label Estimation Time	Total Time
KNN	143.6	0.7	144.3
SVM	0	132.5	132.5
LNP	143.6	0.2	143.8
EGSSC	143.6	2.4	146
LSMP	31.4	0.3	31.7

ever-known largest NUS-WIDE data set. We shows its superiority in terms of both accuracy and efficacy. Our future work will follow two directions: 1) extend the image annotation datasets to web-scale and further testify the scalability of our proposed method; 2) develop more elegant algorithms for KL-based label propagation which shows better convergent speed.

6. ACKNOWLEDGMENTS

This research was supported by Research Grants NRF2007IDM-IDM002-047 on NRF/IDM Program and by AcRF Tier-1 Grant of R-263-000-464-112, Singapore.

7. REFERENCES

[1] A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Commun. ACM*, 51(1):117–122, February 2008.

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

[3] E. J. Candès, J. K. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, February 2006.

[4] G. Chen, Y. Song, F. Wang, and C. Zhang. Semi-supervised multi-label learning by solving a sylvester equation. In *SIAM International Conference on Data Mining*, 2008.

[5] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y.-T. Zheng. Nus-wide: A real-world web image database from national university of singapore. In *CIVR*, July 2009.

[6] R. Collobert, F. H. Sinz, J. Weston, and L. Bottou. Large scale transductive svms. *Journal of Machine Learning Research*, 7:1687–1712, September 2006.

[7] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley Series in Telecommunications, 1991.

[8] O. Delalleau, Y. Bengio, and N. Le Roux. Efficient non-parametric function induction in semi-supervised learning. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, pages 96–103, 2005.

[9] R. Duda, D. Stork, and P. Hart. *Pattern Classification*. JOHN WILEY, 2000.

[10] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the Symposium on Theory Computing*, 1998.

[11] M. Karlen, J. Weston, A. Erkan, and R. Collobert. Large-scale manifold transduction. In *ICML*, 2008.

[12] D. Liu, X.-S. Hua, L. Yang, M. Wang, and H. jiang Zhang. Tag ranking. In *WWW*, 2009.

[13] Y. Liu, R. Jin, and L. Yang. Semi-supervised multi-label learning by constrained non-negative matrix factorization. In *AAAI*, 2006.

[14] Y. Mu, J. Shen, and S. Yan. Weakly-supervised hashing in kernel space. In *CVPR*, 2010.

[15] G.-J. Qi, X.-S. Hua, Y. Rui, J. Tang, T. Mei, and H.-J. Zhang. Correlative multi-label video annotation. In *MM*, 2007.

[16] S.T.Roweis and L.K.Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

[17] A. Subramanya and J. Bilmes. Entropic graph regularization in non-parametric semi-supervised classification. In *NIPS*, 2009.

[18] J. Tang, S. Yan, R. Hong, G.-J. Qi, and T.-S. Chua. Inferring semantic concepts from community-contributed images and noisy tags. In *MM*, 2009.

[19] I. W. Tsang and J. T. Kwok. Large-scale sparsified manifold regularization. In *NIPS*, 2006.

[20] F. Wang and C. Zhang. Label propagation through linear neighborhoods. In *ICML*, June 2006.

[21] J. Yuan, J. Li, and B. Zhang. Exploiting spatial

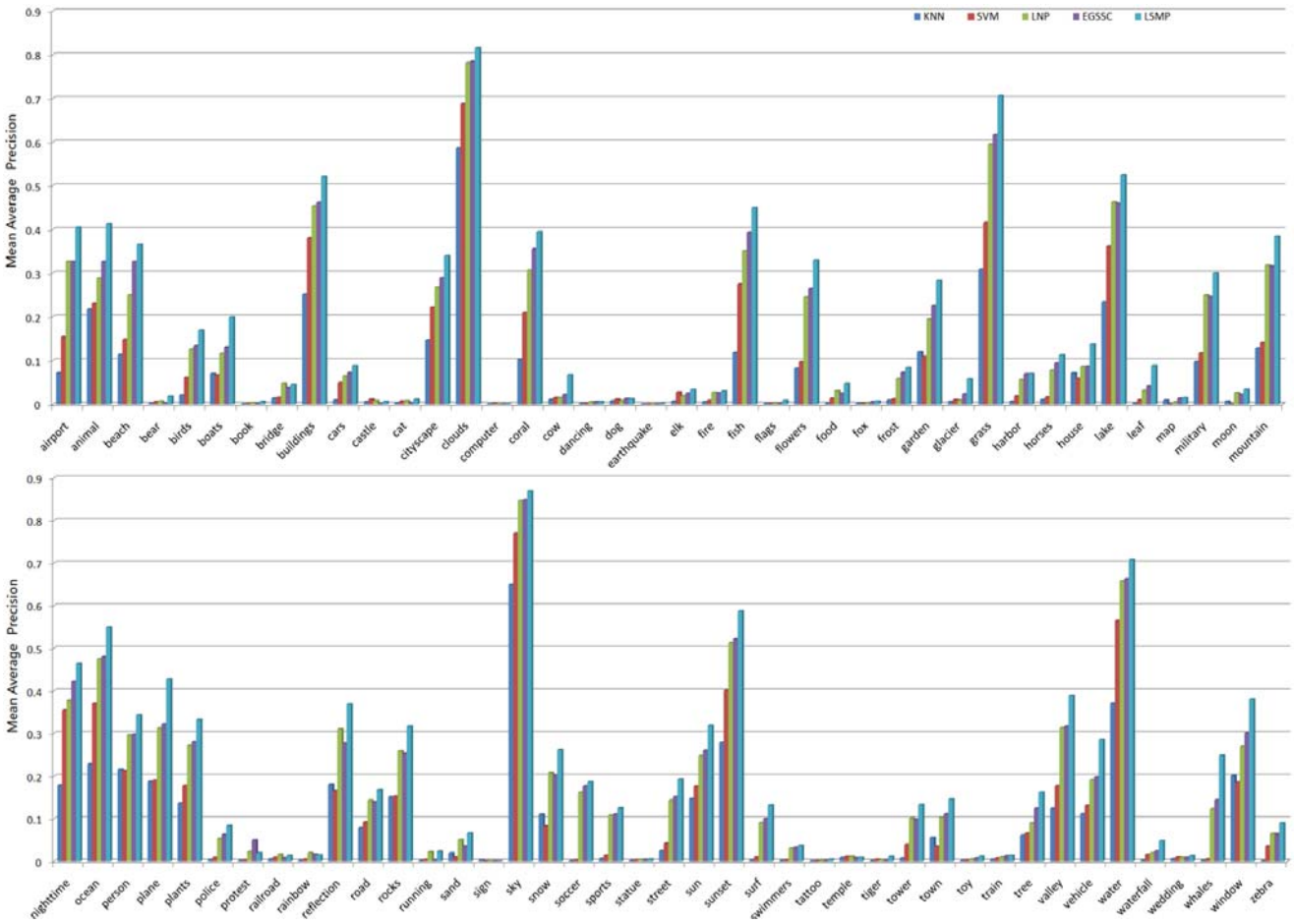

Figure 9: The comparison of APs for the 81 concepts with $\tau = 1.0$ on NUS-WIDE.

context constraints for automatic image region annotation. In *MM*, 2007.

[22] Z.-J. Zha, T. Mei, J. Wang, Z. Wang, and X.-S. Hua. Graph-based semi-supervised learning with multiple labels. *Journal of Visual Communication and Image Representation*, 20(2):97–103, February 2009.

[23] X. Zhu. *Semi-supervised learning with graphs*. Carnegie Mellon University, 2005.

[24] X. Zhu. *Semi-Supervised Learning Literature Survey*. Carnegie Mellon University, 2006.

APPENDIX: Convexity of $D_1(p)$ and $D_2(p, q)$

PROOF: The convexity of $D_1(p)$ is obvious if $D_{KL}(r_i \parallel p_i)$ and $D_{KL}(p_i \parallel \sum_{j\in N(i)} w_{ij}p_j)$ prove convex. Consequently, to justify the convexity of $D_1(p)$, first we elaborate on the convexity of KL divergence defined on two probability mass functions, which has already been studied in the fields of both information theory [7] and convex optimization [2].

Specifically, for $D_{KL}(p \parallel q)$ defined on two pairs of probability mass functions (p_1, q_1) and (p_2, q_2), the convexity of D_{KL} equivalently implies the following fact:

$$D_{KL}(\lambda p_1 + (1-\lambda)p_2 \parallel \lambda q_1 + (1-\lambda)q_2) \le \lambda D_{KL}(p_1 \parallel q_1) + (1-\lambda)D_{KL}(p_2 \parallel q_2), \quad (17)$$

where $\lambda \in [0, 1]$. The correctness of the above inequality is clear by applying the log-sum inequality [7], i.e.,

$$\left(\sum_{i=1}^{n} a_i\right) \log \frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} b_i} \le \sum_{i=1}^{n} a_i \log \frac{a_i}{b_i},$$

on both the left and right sides of the following inequality:

$$D_{KL}(\lambda p_1 + (1-\lambda)p_2 \parallel \lambda q_1 + (1-\lambda)q_2) = \sum_y (\lambda p_1(y) + (1-\lambda)p_2(y)) \log \frac{\lambda p_1(y) + (1-\lambda)p_2(y)}{\lambda q_1(y) + (1-\lambda)q_2(y)}.$$

It is easily verified that

$$D_{KL}(\lambda p_1 + (1-\lambda)p_2 \parallel \lambda q_1 + (1-\lambda)q_2) \le \sum_y \lambda p_1(y) \log \frac{\lambda p_1(y)}{\lambda q_1(y)} + \sum_y (1-\lambda)p_2(y) \log \frac{(1-\lambda)p_2(y)}{(1-\lambda)q_2(y)} = \lambda D_{KL}(p_1 \parallel q_1) + (1-\lambda)D_{KL}(p_2 \parallel q_2). \quad (18)$$

Thus $D_{KL}(r_i \parallel p_i)$ is convex.

And likewise the convexity of $D_{KL}(p_i \parallel \sum_{j\in N(i)} w_{ij}p_j)$ can be justified, observing that $\sum_{j\in N^k(i)} w_{ij}p_j$ is a convex, linear combination of several variables. Hence $D_1(p)$ is convex.

Using the similar tricks above, $D_2(p, q)$ is also demonstrated to be convex.