Evaluation of dimensionality reduction methods for image auto-annotation

Hideki Nakayama[1]
nakayama@isi.imi.i.u-tokyo.ac.jp
Tatsuya Harada[1,2]
harada@isi.imi.i.u-tokyo.ac.jp
Yasuo Kuniyoshi[1]
kuniyosh@isi.imi.i.u-tokyo.ac.jp

[1] Graduate School of Information Science and Technology
The University of Tokyo
Tokyo, Japan

[2] PRESTO, JST

Abstract

Image auto-annotation is a challenging task in computer vision. The goal of this task is to predict multiple words for generic images automatically. Recent state-of-the-art methods are based on a non-parametric approach that uses several visual features to calculate distances between image samples. While this approach is successful from the viewpoint of annotation accuracy, the computational costs, in terms of both complexity and memory use, tend to be high, since non-parametric methods require many training instances to be stored in memory to compute distances from a query. In this paper, we investigate several linear dimensionality reduction methods for efficient image annotation. Using the additional information provided by multiple labels, we can obtain a small representation preserving (and hopefully improving) the semantic distance of a visual feature. Linear methods are computationally reasonable and are suitable for practical large-scale systems, although only limited comparison of such methods is available in this research field. Extensive experiments and analyses on various datasets and visual features show how these simple methods can be applied effectively to image annotation.

1 Introduction

Image auto-annotation is one of the most important challenges in computer vision. The goal of this task is to predict multiple keywords for generic images captured in real environments. With the proliferation of unlabeled images and videos on the Web and personal devices, automatic annotation methods are becoming more and more important to make such images accessible to users. Owing to the generic purpose of the system, such methods will inevitably require an enormous number of real-world images for training. Fortunately, current technology makes it possible to exploit the immensity of the Internet to obtain data for building these systems. For example, Torralba *et al*. [31] showed that using extensive Web data, despite it being contaminated with strong noise, can drastically improve image recognition accuracy. This pioneering study indicates that the success of image annotation depends on the amount of training data and the scalability of the system.

Hitherto, many approaches have been proposed to solve the image annotation problem, including classification-based [6, 9], region-based [11, 12, 20], graph-based [21, 22], topic

BMVC 2010 doi:10.5244/C.24.94

model [3, 25], and generative image patch modeling [5] approaches. Notably, recent studies have shown that a simple non-parametric nearest neighbor-like approach is quite successful. For example, Makadia *et al*. proposed the joint equal contribution (JEC) method [24], which exploits multiple visual features (*e.g*., color histograms and Haar wavelets) to improve performance. For each feature, a base distance is defined using an appropriate metric in the feature space (*e.g*., χ^2 distance for color histograms and L1 distance for Haar wavelets). Then, all the base distances are concatenated with equal weights to retrieve the nearest neighbors. Despite its simplicity, JEC achieved the best performance as of 2008. Furthermore, Guillaumin *et al*. proposed TagProp [13], which realizes state-of-the-art performance. This method makes use of 15 powerful global and local features, including bag-of-visual-words (BoW) [8] and GIST features [27], amongst others. TagProp differs from JEC in that the weights for the base distances are optimized in the metric learning framework by directly maximizing the log-likelihood of the tag prediction. The success of these methods is thought-provoking, and is somewhat analogous to that of the multiple kernel learning [19] approach in categorization tasks. The key issue here for improving performance is to employ rich visual features with appropriate distance metrics defined in raw feature spaces. This means that, image representation can become rather high-dimensional. For example, TagProp uses 15 features, resulting in more than 37,000 dimensions.

While this approach is successful from the viewpoint of annotation accuracy, its computational costs, in terms of both complexity and memory use, however, tend to be high due to the size of the training datasets. A non-parametric method needs to store all training instances in memory to compute their respective distances from the input queries. This cost becomes prohibitive when high-dimensional features and a large number of training samples are used. Moreover, to realize a practical large-scale system with acceptable response speed, implementation of an approximate nearest neighbor (ANN) search method will be mandatory. With some exceptions [17, 18], most ANN methods have assumed Euclidean distance as the similarity measure in the input feature space [10, 34]. Many recent works have exploited machine learning techniques [32, 34], so that the original Euclidean distance can be approximated by the Hamming distance between small binary codes. Although many of these consider an unsupervised setting, some studies consider propagating supervised label information [30, 32]. The training costs of these methods, however, are generally expensive. Moreover, ANN methods often become ineffective with a high dimension of original features, a problem known as the "curse of dimensionality".

In this study, we focus on a fundamental problem: given a visual feature representation, how far can we go using simple linear dimensionality reduction methods to compress the semantic distance of images? Using the sample labels for supervision, new Euclidean distance metrics are embedded in a small-dimensional subspace. This should assist many ANN methods based on Euclidean distance. Clearly, linear methods have both advantages and limitations. One important merit is the high scalability thereof. Linear methods enable training in time linear to the number of samples. This property is beneficial in an open system, where labeled training data may evolve over time. However, owing to the linear assumption, one would not expect these methods always to work effectively. By ignoring the non-linear metric in the original data manifolds, these methods may exhibit poor performance. Therefore, we also consider embedding these metrics using a kernel machine in an efficient manner. Our objective in this study is to perform extensive comparisons of several methods using various datasets and visual features, and to consider under which circumstances the methods are effective. We also show how they can be applied effectively to image annotation.

2 Compared methods

Suppose we have a p-dimensional image feature $\mathbf{x}$, and a q-dimensional label feature $\mathbf{y}$. The details of the features are described in Section 3.3. Suppose also, that we have N labeled training samples $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$. We let $C = \begin{pmatrix} C_{xx} & C_{xy} \\ C_{yx} & C_{yy} \end{pmatrix}$ denote the sample covariance matrix obtained from the training dataset, where $C_{xx} = \frac{1}{N}\sum_i^N (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T$, $C_{yy} = \frac{1}{N}\sum_i^N (\mathbf{y}_i - \bar{\mathbf{y}})(\mathbf{y}_i - \bar{\mathbf{y}})^T$, $C_{xy} = \frac{1}{N}\sum_i^N (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{y}_i - \bar{\mathbf{y}})^T$, and $C_{yx} = C_{xy}^T$. In the above equations, $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$ denote the sample means. The objective is to obtain a new d-dimensional small vector $\mathbf{r}$ ($d << p$), whose distance metric could be the L2 distance.

2.1 Principal component analysis (PCA, PCAW)

PCA is the most basic unsupervised dimensionality compression method, and is still widely used in various areas of computer vision. It finds a subspace that best preserves the variance of the original feature distribution. The projection matrix of PCA is obtained by the following eigenvalue problem: $C_{xx}A = A\Omega \;\; (A^T A = I_d)$, where Ω is a diagonal matrix with eigenvalues as elements. A compressed latent vector is obtained by $\mathbf{r}_{PCA} = A^T(\mathbf{x} - \bar{\mathbf{x}})$. Also, it is empirically known that whitening principal components (*i.e.*, normalizing the variance of principal components) sometimes results in a better distance metric. A whitened vector is obtained by $\mathbf{r}_{PCAW} = \Omega^{-1/2} A^T(\mathbf{x} - \bar{\mathbf{x}})$.

2.2 Partial least squares analysis (PLS, nPLS)

Partial least squares (PLS) [35] is a common tool for multi-modal dimensionality compression. It finds linear transformations $\mathbf{s} = V_x^T(\mathbf{x} - \bar{\mathbf{x}})$ and $\mathbf{t} = V_y^T(\mathbf{y} - \bar{\mathbf{y}})$ that maximize the covariance between new values $\mathbf{s}$ and $\mathbf{t}$. The projection matrices V_x and V_y are obtained by the following eigenvalue problems:

$$C_{xy}C_{yx}V_x = V_x\Theta \quad (V_x^T V_x = I_d), \tag{1}$$
$$C_{yx}C_{xy}V_y = V_y\Theta \quad (V_y^T V_y = I_d), \tag{2}$$

where, Θ is a diagonal matrix with eigenvalues as elements. A latent vector is obtained by $\mathbf{r}_{PLS} = V_x^T(\mathbf{x} - \bar{\mathbf{x}})$. The result of PLS is strongly influenced by the variances of the original features. Therefore, we also test PLS after normalizing the variances of original feature elements. We refer to this as normalized PLS (nPLS).

Although PLS is a classical method, it has been employed successfully in a state-of-the-art human detection method [29]. The authors compressed 170,000-dimensional features into 20-dimensional latent features without much deterioration in performance, making large-scale training tractable. Whereas the semantic aspect (y-view) in [29] is binary (human or non-human), we have multiple labels for a single image. These labels are expected to provide rich semantic information.

2.3 Canonical correlation analysis (CCA)

CCA was first proposed by Hotelling [16] in 1936, and has hitherto been one of the most basic and important multivariate analysis methods [15]. CCA is closely related to PLS. Whereas PLS finds the projections that maximize the covariance between the two new values, CCA finds those that maximize the correlation. More details can be found in [4]. We obtain

projection matrices U_x and U_y by solving the following eigenvalue problems:

$$C_{xy}C_{yy}^{-1}C_{yx}U_x = C_{xx}U_x\Lambda^2 \quad (U_x^T C_{xx}U_x = I_d), \tag{3}$$

$$C_{yx}C_{xx}^{-1}C_{xy}U_y = C_{yy}U_y\Lambda^2 \quad (U_y^T C_{yy}U_y = I_d), \tag{4}$$

where Λ is the diagonal matrix of the first d canonical correlations. A latent vector is obtained by $\mathbf{r}_{CCA} = U_x^T(\mathbf{x}-\bar{\mathbf{x}})$. Note that d is at most $min\{p,q\}$ in CCA.

CCA has been successfully employed in some previous studies on image annotation [14, 36]. The main focus in these studies, however, is to construct a strong regression model using kernelized CCA (KCCA), rather than dimensionality compression. Our objective is more similar to that of the correlational spectral clustering [2], in which CCA and KCCA are used for unsupervised clustering of weakly coupled image-text documents. The authors showed that the distance between instances can be estimated more accurately in the latent space, although the dimensionality thereof is substantially reduced.

2.4 Canonical contextual distance (CCD)

Nakayama *et al*. proposed the canonical contextual distance [26], which is based on the probabilistic canonical correlation analysis (PCCA)[1] framework. Although both PLS and CCA use semantic information (y-view) for learning projection matrices, the latent variable itself is estimated using the image side (x-view) only. Considering the probabilistic background of CCA, we can use both views for estimating latent variables and computing their distances. Moreover, PCCA automatically weights each dimension of the latent variable according to its effectiveness (canonical correlation), whereas CCA treats them similarly.

2.4.1 Probabilistic canonical correlation analysis

PCCA is represented by a graphical model in Fig. 1. $\mathbf{z}$ in this figure represents a latent variable reflecting sample similarities in terms of both image and labels. In this manner, we can efficiently estimate the latent space using label information with a stochastically rigorous background. PCCA defines the probability density distributions as follows.

$$\begin{aligned} \mathbf{z} &\sim \mathcal{N}(0, I_d), \ \ min\{p,q\} \geq d \geq 1, \\ \mathbf{x} \mid \mathbf{z} &\sim \mathcal{N}(W_x\mathbf{z}+\mu_x, \Psi_x), \ \ W_x \in \mathscr{R}^{p\times d}, \Psi_x \succeq 0, \\ \mathbf{y} \mid \mathbf{z} &\sim \mathcal{N}(W_y\mathbf{z}+\mu_y, \Psi_y), \ \ W_y \in \mathscr{R}^{q\times d}, \Psi_y \succeq 0. \end{aligned} \tag{5}$$

The maximum likelihood solution of this model basically corresponds to the solution of normal CCA: *e.g.*, $\widehat{W}_x = C_{xx}U_xM_x$, $\widehat{W}_y = C_{yy}U_yM_y$, where $M_x, M_y \in \mathscr{R}^{d\times d}$ are arbitrary matrices such that $M_xM_y^T = \Lambda$ and the spectral norms of M_x and M_y are smaller than one.

Using this structure, we can derive the posterior probability of a sample in the latent space. When only an image feature $\mathbf{x}$ of the sample is given, $p(\mathbf{z}|\mathbf{x})$ becomes a Gaussian with mean $\dot{\mathbf{z}}$ and variance Φ_x defined as:

$$\dot{\mathbf{z}} = E(\mathbf{z}|\mathbf{x}) = M_x^T U_x^T(\mathbf{x}-\bar{\mathbf{x}}), \tag{6}$$

$$\Phi_x = var(\mathbf{z}|\mathbf{x}) = I - M_xM_x^T. \tag{7}$$

Similarly, when both an image feature $\mathbf{x}$ and a label feature $\mathbf{y}$ are given, we have,

$$\ddot{\mathbf{z}} = E(\mathbf{z}|\mathbf{x},\mathbf{y}) = \begin{pmatrix} M_x \\ M_y \end{pmatrix}^T \begin{pmatrix} (I-\Lambda^2)^{-1} & -(I-\Lambda^2)^{-1}\Lambda \\ -(I-\Lambda^2)^{-1}\Lambda & (I-\Lambda^2)^{-1} \end{pmatrix} \begin{pmatrix} U_x^T(\mathbf{x}-\bar{\mathbf{x}}) \\ U_y^T(\mathbf{y}-\bar{\mathbf{y}}) \end{pmatrix}, \tag{8}$$

$$\Phi_{xy} = var(\mathbf{z}|\mathbf{x},\mathbf{y}) = I - \begin{pmatrix} M_x \\ M_y \end{pmatrix}^T \begin{pmatrix} (I-\Lambda^2)^{-1} & -(I-\Lambda^2)^{-1}\Lambda \\ -(I-\Lambda^2)^{-1}\Lambda & (I-\Lambda^2)^{-1} \end{pmatrix} \begin{pmatrix} M_x \\ M_y \end{pmatrix}. \tag{9}$$

M_x and M_y have an arbitrary property of scale and rotation. Here, we define them simply using the following diagonal matrices: $M_x = \Lambda^{\beta}$, $M_y = \Lambda^{1-\beta}$ $(0 < \beta < 1)$. β is a parameter to balance the contribution of the image feature and label feature in estimating the latent variable. With this definition, Φ_x and Φ_{xy} are now diagonal.

2.4.2 1-view CCD (CCD1)

As described above, a sample forms a Gaussian in the latent space. [26] defines the similarity measure as the joint probability of the distributions generated by the samples, and uses it for kernel density estimation. Similarly, in this study, we use Kullback-Leibler (KL) divergence between the distributions as a distance measure.

Let us consider the KL divergence between a query $\mathbf{x}_q$ and a training sample $\{\mathbf{x}_t, \mathbf{y}_t\}$ in the latent space. When considering the x-view only (Fig. 2(a)), the divergence becomes:

$$KL(p(\mathbf{z}|\mathbf{x}_q), p(\mathbf{z}|\mathbf{x}_t)) = (\dot{\mathbf{z}}_q - \dot{\mathbf{z}}_t)^T \Phi_x^{-1} (\dot{\mathbf{z}}_q - \dot{\mathbf{z}}_t). \tag{10}$$

This can be computed as the Euclidean distance of $\mathbf{r}_{CCD1} = \Phi_x^{-1/2} \dot{\mathbf{z}}$.

2.4.3 2-view CCD (CCD2)

All training images have corresponding labels that can provide additional information to obtain a better representation and distance metric. Considering both views of the training samples (Fig. 2(b)), the KL divergence becomes:

$$KL(p(\mathbf{z}|\mathbf{x}_q), p(\mathbf{z}|\mathbf{x}_t, \mathbf{y}_t)) = \frac{1}{2} \log \frac{|\Phi_{xy}|}{|\Phi_x|} - \frac{d}{2} + \frac{1}{2} \mathrm{Tr}(\Phi_{xy}^{-1} \Phi_x) + (\dot{\mathbf{z}}_q - \ddot{\mathbf{z}}_t)^T \Phi_{xy}^{-1} (\dot{\mathbf{z}}_q - \ddot{\mathbf{z}}_t). \tag{11}$$

Since the first three terms are constant, this can also be computed as the Euclidean distance, defining $\mathbf{r}^q_{CCA2} = \Phi_{xy}^{-1/2} \dot{\mathbf{z}}_q$ for a query image and $\mathbf{r}^t_{CCA2} = \Phi_{xy}^{-1/2} \ddot{\mathbf{z}}_t$ for a training sample.

2.5 Complexity of PCA, PLS, and CCA

The off-line training phase for the above mentioned methods comprises three steps: 1) calculating covariances, 2) solving eigenvalue problems, and 3) projecting training samples using the learned metric. Table 1 summarizes the training complexity of each method. For fixed features, these methods scale the number of training samples with linear complexity, a property that would be beneficial to large scale problems.

2.6 Embedding non-linear metrics

Although PLS and CCA can perform semantic dimensionality reduction effectively, they have difficulty dealing with specific features that have non-linear distance metrics. In this case, we first embed the non-linear metrics in a Euclidean space via kernel PCA (KPCA) [28]. Suppose a kernel function $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$ is given, where $\phi : \mathbf{x} \to \phi(\mathbf{x})$ denotes the projection that maps an input vector onto a high-dimensional feature space. Using $n (n \leq N)$ training samples, we compute the kernel base vector $\mathbf{k}_x = \big(K(\mathbf{x}, \mathbf{x}_1), ..., K(\mathbf{x}, \mathbf{x}_n)\big)^T$. Using a kernel trick, the solution of KPCA becomes a linear problem on $\mathbf{k}_x$ coordinates. The embedded vector is obtained as $\tilde{\mathbf{x}} = B^T \mathbf{k}_x$, where B is the KPCA projection matrix. We can use $\tilde{\mathbf{x}}$ as the new input for PLS, CCA, and CCD. In our implementation, we use the exponentiated distance function, $K(\mathbf{x}_i, \mathbf{x}_j) = \exp\big(-\frac{1}{2P} \mathrm{dist}(\mathbf{x}_i, \mathbf{x}_j)\big)$ as the kernel function. Here, $\mathrm{dist}(\mathbf{x}_i, \mathbf{x}_j)$ is a base distance, such as the χ^2 or L2 distance, and P is the mean of the base distances in the n training samples.

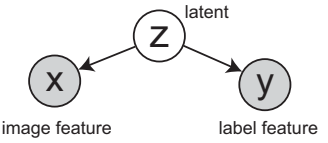

Figure 1: Graphical model of PCCA. **z** represents an unobserved latent variable.

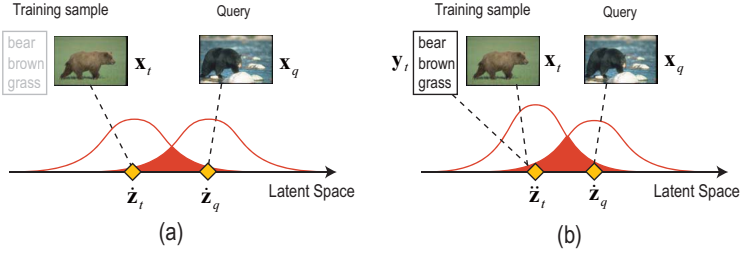

Figure 2: Illustration of canonical contextual distances. Estimation of distance between a query and training sample: (a) from the x-view only (CCD1); and (b) considering both the x and y-views (CCD2).

	PCA	PLS	CCA
1)	$O(Np^2)$	$O(Npq)$	$O(N(p^2+pq+q^2))$
2)	$O(p^3)$	$O(min\{p^2(p+q),(p+q)q^2\})$	$O(p^3+q^3+p^2q+pq^2)$
3)	$O(Npd)$	$O(Npd)$	$O(Npd)^*$

Table 1: Computational complexity for PCA, PLS, and CCA based methods: 1) calculating covariances, 2) solving eigenvalue problems, and 3) projecting training samples using the learned metric. (*)$O(N(p+q)d)$ for CCD2.

Theoretically, the larger n becomes, the better the performance is. In a standard approach, we may use all available training samples for a kernel trick ($n = N$). However, computing the kernel base of a query requires n raw training samples in memory. If n is large, this is computationally as expensive as a brute-force search in a raw feature space, thus, destroying our objective. Therefore, we randomly sample a small number of training samples ($n = 300$) for kernelization, and compute the eigenvalue decomposition of KPCA using all N samples. Our experimental results show that we can obtain satisfactory performance with this setup.

3 Experimental setup

3.1 Benchmarks

We performed extensive experiments using three different datasets and various image features. The **Corel5K** dataset [11] has been the de facto standard for problems of image annotation. This dataset contains 5000 pairs of images and labels. Each image has been manually annotated with an average of 3.4 keywords. 4500 samples are specified as training data, with the remaining 500 samples as the test data. The dictionary contains 260 words. The **IAPR-TC12** dataset consists of 17,665 training samples and 1,962 testing samples. Each sample is annotated with an average of 5.7 words out of 291 candidate words. We followed

the same setup as in [13, 24]. The **NUS-WIDE** dataset [7] is a comparatively large Web image dataset, consisting of 161,789 training samples and 107,859 testing samples downloaded from Flickr. All samples are supervised and labeled with 81 concepts. Note that many images in the dataset are "negative" and have no labels; that is, none of the 81 concepts appear within the images. We randomly sampled 2,000 "positive" images from the testing samples and used these as our testing data.

3.2 Annotation method and evaluation protocol

Since our interest was in the performance of distance metrics for non-parametric image annotation, we simply used the k nearest neighbor method with a brute-force search. The system output the most frequent labels in the k retrieved neighbors. We prioritized a rare label in the training dataset if the numbers of relevant neighbors were equal. With Corel5K and IAPR-TC12, we tested $k = 1, 2, 4, 8, 16, 32$ and took the best performance. Similarly, we tested $k = 50, 100, 150, 200$ using the NUS-WIDE dataset.

For evaluation of annotation, we followed the methodology of previous works [11]. The system annotates test images with 5 words each. These words are then compared with the ground truth. For each label w_i, a denotes the number of images that are correctly annotated by the system, b denotes the number of images with w_i as the ground truth, and c denotes the number of images that the system annotates with w_i. Then, $\mathrm{Recall}(w_i) = a/b$ and $\mathrm{Precision}(w_i) = a/c$. These are averaged over all the testing words to give Mean-Recall (MR) and Mean-Precision (MP). Because of the trade-off of these two scores, we evaluated the total performance using the F-measure $= 2 \times \mathrm{MR} \times \mathrm{MP}/(\mathrm{MR} + \mathrm{MP})$.

3.3 Image and label features

To facilitate a quantitative comparison, we used publicly available feature files in the experiment. For Corel5K and IAPR-TC12, we tested four features: 1) 1000-dimensional densely-sampled SIFT [23] BoW, 2) 100-dimensional densely-sampled Hue [33] BoW, 3) 512- dimensional GIST [27], and 4) 4096-dimensional HSV color histogram. All these features are employed in TagProp [13], and are available on the authors' Web page [1]. For the NUS-WIDE dataset, we tested: 1) 73-dimensional edge histogram, 2) 144-dimensional color correlogram, 3) 225-dimensional grid color moment, and 4) 500-dimensional SIFT BoW. These features are provided by the authors of [7] [2]. To provide baselines, we computed some base distances for each feature (*e.g.*, χ^2 distance, L1 and L2 distance, histogram intersection).

Regarding label features, we used a binary vector indicating the presence of each word. Each element of the vector corresponds to one word. For example, if an image is annotated with "sky", "plane", and "cloud", the label feature becomes $(1,0,0,1,1)^T$, where the dictionary contains "sky", "sea", "mountain", "plane", and "cloud". The inner product of two label features is thus equal to the number of common words in the corresponding labels. Intuitively, this makes the Euclidean assumption on the feature space and application of linear methods reasonable. However, because of the sparsity of the label feature, the covariance matrix C_{yy} may become singular, which in turn may present problems with CCA. In this case, we can add regularization terms to make the eigenvalue problem stable. For example, C_{yy} can be replaced by $C_{yy} + \gamma I$, where γ is a small positive number.

[1] http://lear.inrialpes.fr/data

[2] http://lms.comp.nus.edu.sg/research/NUS-WIDE.htm

4 Experimental results

We first report the results for the Corel5K and IAPR-TC12 datasets. In these small datasets, the dimension of visual features is too large for CCA, which requires the inverse of the covariance. Therefore, with the exception of Hue BoW, we initially compressed visual features into 200-dimensional vectors using PCA, before computing CCA, CCD1, and CCD2. For embedding non-linear metrics, we exploited the first 200 principal components of KPCA, and used these as the new visual features. We placed the χ^2 distance into a kernel for SIFT BoW, Hue BoW, and the color histogram, and the L1 distance for GIST (see Section 2.6).

Figure 3 gives a comparison of the annotation accuracy (F-measure). In many cases, nPLS and CCD exhibit superior performance, and achieve comparable or better performance than with the original L2 distance, using the first 10 or 20 dimensions only. However, in terms of accuracy, it is difficult for simple linear methods to compete with the original domain-specific metric. This is especially true with Hue BoW and the color histogram. In such a case, KPCA embedding works effectively and substantially improves the performance, although a small fraction of training samples was used for kernelization (n=300). Another observation is that the performance of the CCA family is often ordered CCD2 > CCD1 > CCA, which indicates the importance of considering the y-view explicitly for distance computation. While simple CCA is not always effective and is sometimes outperformed by PCA or PCAW, we observe that CCD2 consistently outperforms these methods.

Next, we summarize the results for the NUS-WIDE dataset in Fig. 4. Since the provided features are normalized, we only investigate L1 and L2 as baseline distances, except for BoW. In this larger dataset, the effect of semantic dimensionality reduction seems to be more profound. In particular, CCD shows a substantial improvement over the original distances in many cases, using only a dozen or so dimensions. However, unlike in the previous experiment, CCD1 and CCD2 perform almost equally. Although NUS-WIDE is a relatively large dataset, the label feature in this experiment consists of only 81 basic concepts. Our hypothesis is that, while this label feature is effective in the dimensionality reduction phase, it is too weak to contribute to the actual distance computation in the latent space.

Finally, we report actual computation times using the NUS-WIDE dataset. The training time for each method is summarized in Table 2. Target dimensionality is set to $d = 20$, and we use an 8-core Xeon 3.20 GHz machine for computation. PLS and CCD can be computed with moderate additional time from PCA. This is especially true when the dimension of the visual feature is much larger than the vocabulary size ($p >> q$), and is well explainable in terms of the analysis in Section 2.5. For example, PLS works faster than PCA in 500-dimensional BoW.

5 Conclusion

We investigated and compared several linear dimensionality reduction methods for non-parametric image annotation. Obtaining powerful small codes in a scalable manner is a crucial issue in implementing large-scale image annotation systems. Linear methods enable training in linear time and are suitable for this purpose. Overall, we can balance the trade-off between computational efficiency and annotation accuracy by selecting the dimensions of the latent features. Using the semantic information provided by multiple labels, we can obtain a small-dimensional latent subspace reflecting the semantic distance of instances. Moreover, the superior performance of 2-view CCD indicates the importance of using label information explicitly in actual distance computation.

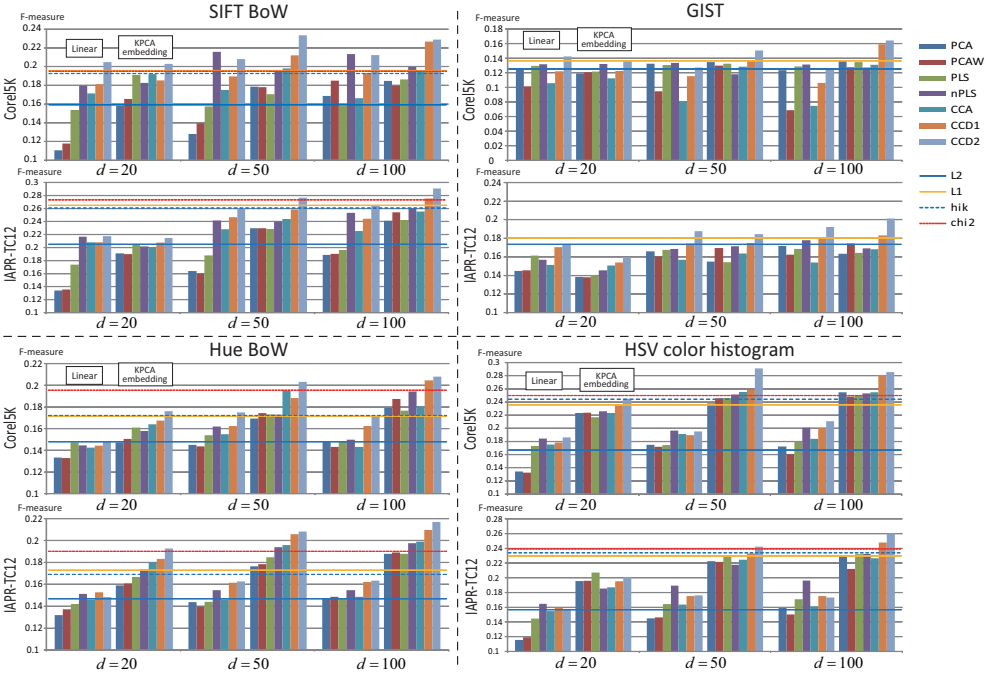

Figure 3: Comparison of annotation performance (F-measure) with designated dimensions d. For each entry, the left set of bars corresponds to normal linear methods, while the right set corresponds to those with KPCA embedding. Top left: SIFT BoW (1000-dim). Top Right: GIST (512-dim). Bottom left: Hue BoW (100-dim). Bottom right: HSV color histogram (4096-dim).

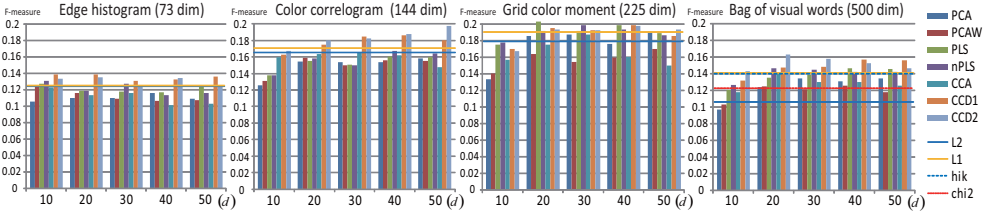

Figure 4: Results for the NUS-WIDE dataset (F-measure). Methods are compared with different features in the designated dimensionality (d).

	NUS-WIDE (161,789 samples, 81 words)			
	EDH (73 dim)	Cor. (144 dim)	C.mom. (225 dim)	BoW (500 dim)
PCA (PCAW)	1.2	2.0	3.4	8.0
PLS	1.9	2.6	3.6	6.7
nPLS	3.5	5.2	7.4	14.6
CCA (CCD)	2.1	3.0	4.5	10.1

Table 2: Computation times for training the system using the NUS-WIDE dataset with each method [s]. We found that the differences in execution time between PCA and PCAW, and between CCA and CCD are negligible for a small d.

Reported methods are particularly effective for generic features whose optimal non-linear metric is unknown. When such metrics are known, we can improve the performance at a moderate cost, by means of a kernel trick, using a small number of samples.

In future work, we aim to extend these methods by incorporating unsupervised ANN methods to further compress visual features and build rapid annotation systems.

Acknowledgment

We are deeply grateful to the authors of [13] and [7] for kindly providing the extensive visual features that made this study possible.

References

[1] F. R. Bach and M. I. Jordan. A probabilistic interpretation of canonical correlation analysis. Technical Report 688, Department of Statistics, University of California, Berkeley, 2005.

[2] M. B. Blaschko and C. H. Lampert. Correlational spectral clustering. In *Proc. IEEE CVPR*, 2008.

[3] D. M. Blei and M. I. Jordan. Modeling annotated data. In *Proc. ACM SIGIR*, pages 127–134, 2003.

[4] M. Borga, T. Landelius, and H. Knutsson. A unified approach to PCA, PLS, MLR and CCA. Report LiTH-ISY-R-1992, ISY, 1997.

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

[6] E. Y. Chang, K. Goh, G. Sychay, and G. Wu. CBSA: content-based soft annotation for multimodal image retrieval using Bayes point machines. *IEEE Trans. Circuits and Systems for Video Technology*, 13(1):26–38, 2003.

[7] 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 *Proc. ACM CIVR*, 2009.

[8] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Proc. ECCV Workshop on Statistical Learning in Computer Vision*, pages 1–22, 2004.

[9] C. Cusano, G. Ciocca, and R. Schettini. Image annotation using SVM. In *Proc. SPIE Conference on Internet Imaging IV*, volume SPIE, 2004.

[10] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proc. the Symposium on Computational Geometry*, pages 253–262, 2004.

[11] P. Duygulu, K. Barnard, N. de Freitas, and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *Proc. ECCV*, pages 97–112, 2002.

[12] S. Feng, R. Manmatha, and V. Lavrenko. Multiple bernoulli relevance models for image and video annotation. In *Proc. IEEE CVPR*, volume 2, pages 1002–1009, 2004.

[13] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid. Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In *Proc. IEEE ICCV*, pages 309–316, 2009.

[14] D. R. Hardoon, C. Saunders, S. Szedmak, and J. Shawe-Taylor. A correlation approach for automatic image annotation. In *Proc. ADMA*, 2006.

[15] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor. Canonical correlation analysis; an overview with application to learning methods. *Neural Computation*, 16(12):2639–2664, 2004.

[16] H. Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.

[17] B. Kulis and K. Grauman. Kernelized locality-sensitive hashing for scalable image search. In *Proc. IEEE ICCV*, pages 2130–2137, 2009.

[18] B. Kulis, P. Jain, and K. Grauman. Fast similarity search for learned metrics. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 31(12):2143–2157, 2009.

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

[20] V. Lavrenko, R. Manmatha, and J. Jeon. A model for learning the semantics of pictures. In *Proc. NIPS*, 2003.

[21] J. Liu, M. Li, Q. Liu, H. Lu, and S. Ma. Image annotation via graph learning. *Pattern Recognition*, 42(2):218–228, 2009.

[22] J. Liu, M. Li, W.-Y. Ma, Q. Liu, and H. Lu. An adaptive graph model for automatic image annotation. In *Proc. ACM MIR*, pages 61–70, 2006.

[23] D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. IEEE ICCV*, pages 1150–1157, 1999.

[24] A. Makadia, V. Pavlovic, and S. Kumar. A new baseline for image annotation. In *Proc. ECCV*, pages 316–329, 2008.

[25] F. Monay and D. Gatica-Perez. PLSA-based image auto-annotation: constraining the latent space. In *Proc. ACM Multimedia*, 2004.

[26] H. Nakayama, T. Harada, and Y. Kuniyoshi. Canonical contextual distance for large-scale image annotation and retrieval. In *Proc. ACM Multimedia workshop on Large-Scale Multimedia Retrieval and Mining*, pages 3–10, 2009.

[27] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

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

[29] W. R. Schwartz, A. Kembhavi, D. Harwood, and L. S. Davis. Human detection using partial least squares analysis. In *Proc. IEEE ICCV*, pages 24–31, 2009.

[30] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter sensitive hashing. In *Proc. IEEE ICCV*, pages 750–757, 2003.

[31] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: A large dataset for non-parametric object and scene recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 30(11):1958–1970, 2008.

[32] A. Torralba, R. Fergus, and Y. Weiss. Small codes and large image databases for recognition. In *Proc. IEEE CVPR*, 2008.

[33] J. van de Weijer and C. Schmid. Coloring local feature extraction. In *Proc. ECCV*, pages 334–348, 2006.

[34] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *Proc. NIPS*, 2008.

[35] H. Wold. Partial least squares. In S. Kotz and N. Johnson, editors, *Encyclopedia of Statistical Sciences*, volume 6, pages 581–591. John Wiley & Sons, 1985.

[36] O. Yakhnenko and V. Honavar. Multiple label prediction for image annotation with multiple kernel correlation models. In *Proc. IEEE CVPR workshop on Visual Context Learning*, 2009.