Unsupervised Image Embedding Using Nonparametric Statistics

Guobiao Mei
University of California, Riverside
gmei@cs.ucr.edu

Christian R. Shelton
University of California, Riverside
cshelton@cs.ucr.edu

Abstract

Embedding images into a low dimensional space has a wide range of applications: visualization, clustering, and pre-processing for supervised learning. Traditional dimension reduction algorithms assume that the examples densely populate the manifold. Image databases tend to break this assumption, having isolated islands of similar images instead. In this work, we propose a novel approach that embeds images into a low dimensional Euclidean space, while preserving local image similarities based on their scale invariant feature transform (SIFT) vectors. We make no neighborhood assumptions in our embedding. Our algorithm can also embed the images in a discrete grid, useful for many visualization tasks. We demonstrate the algorithm on images with known categories and compare our accuracy favorably to those of competing algorithms.

1 Introduction

In this work, we propose a novel approach to embed images into a Euclidean space so that each image lies close to similar other images, while far apart from those with large distinctions. It is essentially a dimension reduction problem for images. Image embeddings can have a wide range of applications. With the abundance of digital photography, many households have thousands of images stored on their home computer without a suitable method for searching them. Graphic visualization of the entire database can be a great aid in allowing easy retrieval of photographs.

General dimension reduction has been extensively studied. Classical techniques such as principal component analysis (PCA) try to find a linear embedding for high dimensional input. Due to the nonlinearity of images, such methods generally do not work very well.

Much of the recent work on image dimension reduction focuses on learning an underlying manifold, and embedding images into the low-dimensional manifold. In [9], the authors proposed Isomap. It is capable of discovering the nonlinear degrees of freedom of the data and computes the globally optimal solution. They also show that asymptotically Isomap is guaranteed to converge to the true underlying manifold structure. [8] proposed another dimension reduction algorithm: locally linear embedding (LLE). By using pairwise distances, LLE can embed inputs into a single global coordinate space without problematic local minimum. The LLE algorithm is also capable of finding the underlying manifold structure for the high dimensional data. [10] proposes semidefinite embedding (SDE). It overcomes certain limitations of the previous methods. All of these algorithms work on general dimension reduction tasks, and are applicable to image data.

The listed algorithms work for image embedding tasks. However, they assume that the set of images forms a single well-sampled manifold. For many image data sets, this is not true: there are islands (for example different object classes) of images, each of which might be well-described by a low dimensional subspace, but there are no images that demonstrate a "path" from one island to the next. Our method explicitly uses SIFT features and makes no assumptions about the underlying geometry of the space. It merely states that images with similar features should be near each other.

Additionally, as we show in Section 2.3, our method can be adapted to non-Euclidean spaces. In particular, we can embed the images into a grid, suitable for visualization. [6] also propose a visualization specifically for images. It is based primarily on PCA on the raw pixel values and has a postprocessing step to attempt to separate the images for better visualization. By contrast our method has a non-linear embedding and can directly avoid overlap without the need for postprocessing.

2 Image Embedding Problem

In [5], we proposed an algorithm that can visualize collaborative data. Unlike [7], in which they use Term Frequency Inverse Document Frequency (TF-

IDF) based scoring algorithm to retrieve relevant images. We use a real-valued Bayesian network to model the embedded positions of the users and items, along with the ratings that relating them. We then employ a Markov chain Monte Carlo (MCMC) algorithm with simulated annealing to find samples that maximize the posterior likelihood. In addition, we propose to use the nonparametric statistic Kendall's τ [3] as a criterion to evaluate the embedding quality. In this work, we adopt the overall structure of [5]. We do not have "users" that have rated the images, but we employ SIFT features in a similar role (see Section 2). We also simplify their optimization method. Instead of trying to maximize the posterior distribution of a complex graphical model, we propose to directly minimize Kendall's τ. Given this as the end target criterion, we think it more natural to optimize it directly instead of using the posterior likelihood of a probabilistic model as a proxy.

2.1 Problem Formulation

Given a set of images $\mathbf{I}$, our task is to embed them into a D-dimensional space. Distances in this embedded space should capture the image similarity: we want to put similar images near each other, and dissimilar ones far apart.

We first extract the SIFT features for all the images (let S_i be the features from image I_i), and then cluster all features from all images together into m groups, $\{K_j\}$. We have found that the end results are fairly stable with respect to the number of clusters and the clustering algorithm. We use k-means to do this clustering. For any image I_i, we then count N_{ij}, the number of features in S_i that belong to cluster K_j. If we divide N_{ij} by the size of S_i, we have a distribution of "membership" to the cluster K_j for image I_i. So we can consider $r_{ij} = N_{ij}/\|S_i\|$ as the fractional "vote" of I_i for K_j. So, following [5], we embed both the images (items) and SIFT clusters (users) in the same space so that SIFT clusters are near images that they like (have many examples of the feature) and are far away from those that they do not like (do not have the feature).

Consider the case where we already have a potential embedding, containing image points and cluster points. Let the images have points $\{I_i\}$ in that space, and the SIFT feature clusters have points $\{K_j\}$. For any particular image I_i, we can compute its distance to all the cluster points, which we denote $d_{ij} = \|I_i - K_j\|$. From the SIFT feature clustering, we also have the membership distribution r_{ij}. Let $\tau(a, b)$ be Kendall's τ between these two sequences a and b. The image embedding problem can be formulated as finding the $\arg\min$ over the embedded points $\mathbf{I} = \{I_i\}$ and $\mathbf{K} = \{K_j\}$ of

$$T(\mathbf{I}, \mathbf{K}) = \frac{1}{n} \sum\nolimits_{i=1}^{n} \tau(\{r_{ij}\}_{j=1}^{m}, \{d_{ij}\}_{j=1}^{m}) \ . \quad (1)$$

That is, we would like the lists of distances and membership distributions to have exactly reverse orderings (images are near SIFT feature they contain and far from those they do not).

2.2 Simulated Annealing

Exact algorithms to minimize the function T are not possible due to its combinatorial nature. Instead, we use simulated annealing [4]. We begin with a random embedding of $\mathbf{I}$ and $\mathbf{K}$. Samples from a multi-variate Gaussian distribution work fine in practice.

At each step, we randomly choose a point, either an image or a cluster, to resample. We drawn the new potential location from a multi-variate Gaussian proposal distribution centered at the old point. We calculate ΔT, the change in T if the point were moved, and accept the change with probability $\min\{\exp(-\frac{1}{\beta}\Delta T), 1\}$. By recording the number of concordant and discordant pairs for each Kendall's τ, this change can be calculated quickly without recomputing all of Equation 1.

We iterate this resampling procedure until convergence. Since our problem is to find the $\arg\min$ of T, as is standard in simulated annealing, we set β initially to 1, and we let it shrink toward zero.

2.3 Grid-based Image Embedding

The final embedding of the images will inevitably involve much overlap if we plot the images in their embedded space. If the embedding is simply for dimension reduction as an initial step of machine learning, this is not a problem. However, it is a problem if visualization is the desired goal.

Unlike many other dimension reduction algorithms, our framework can be easily adapted to a "grid-based" approach. We do this by setting the target embedding space to be a grid, *i.e.* each image can be only placed into one of the embedding grid cells. The proposal distribution for changing an image can be a uniform distribution over all cells, or, more efficiently, a uniform distribution over the neighbors of the image's current cell. If there is already another image that takes the proposed grid position, then the proposed move is to *swap* the two images, otherwise the proposal is to *move* the image position to the new cell.

Everything else in the above algorithm remains the same. We also restrict the SIFT cluster locations to the grid cells. However, we do not require there to be at

Figure 1. One-dimensional embedding for the *shoes* object in the ALOI data set.

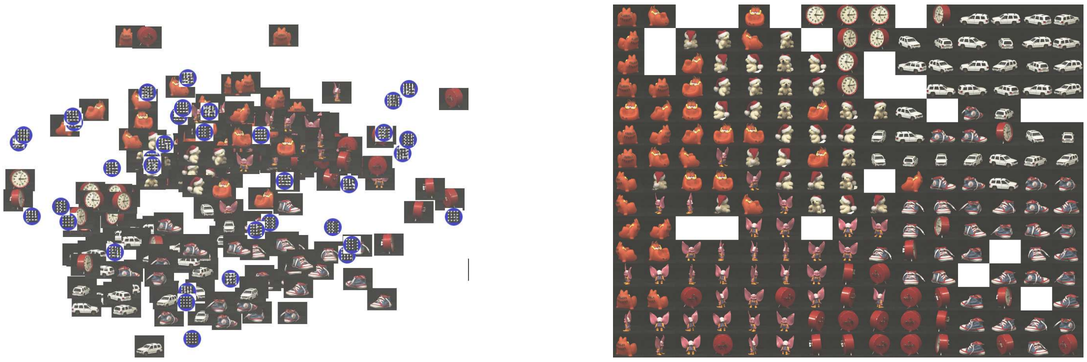

Figure 2. Sample embedding (left) and grid-based results for a subset of ALOI data set.

most one per cell. Rather, the SIFT clusters may coexist on the same cell as we will not be displaying them.

3 Experiments and Results

We tested the embedding algorithms on two data sets. The Amsterdam Library of Object Images (ALOI) [2] contains images of a set of small objects. It has images for 1000 objects, each has 72 images taken from different viewing angles. This data set is noise-free and is relatively easy for unsupervised image embeddings. Caltech101 [1] is another well-known data set containing 101 categories of (more variable) images. It is a harder data set for image dimension reduction.

It is usually difficult to numerically evaluate the embedding results. Fortunately we know the ground-truth category for each image in both data sets we used. We use k-nearest neighbors accuracy (kNNA) to evaluate the embedding:

$$kNNA(I_i) = \frac{\sum_{I_j \in \mathcal{N}_k(I_i)} \text{Category}(I_i)\text{=Category}(I_j)}{k},$$

where $\mathcal{N}_k(I_i)$ is the set of the k nearest neighbors of image I_i in the embedding. The average kNNA for the embedding is just $kNNA = \frac{\sum_i kNNA(I_i)}{\sum_i 1}$.

We compared our simulated annealing embedding (SAE) algorithm with Isomap, LLE, and SDE. Running Isomap, LLE, and SDE with raw pixels produced poor results. Instead, we ran those embedding algorithms using the SIFT distributions as the images' vector representation. This also removes any representation bias. In addition to using Euclidean distance as the standard distance metric, we also tried pairwise Pearson correlations between the SIFT-feature vectors as a distance metric.

3.1 Sample SAE Embeddings

Figure 1 shows a sample embedding of shoe images from the ALOI data set viewed from different angles using SAE. We picked 12 evenly spaced images in the derived embedding from all of the 72 embedded shoe images. It is clear that SAE preserves the pairwise similarity well in terms of the shoe's rotation.

Figure 2 shows a sample two-dimensional Euclidean embedding for the ALOI data set. We randomly chose a subset of 6 objects (categories), each with 36 different images. A small number of images lie far away from the center, so to show clearly, we zoomed in to a smaller region with much higher image density. Note that we also show in the same embedding the positions of the SIFT clusters in circles. The cluster images are generated directly from the gradient intensities specified in the vector. To be specific, the 128-dimensional vector contains 8 gradient values for 16 sub-windows of interest. We plotted these values in corresponding positions in a small image.

Figure 2 shows the result of using a grid-based embedding on the same data sets. These grid-based embeddings provide a user-friendly image grid. Unlike the unconstrained Euclidean embeddings, there is no overlap obscuring some of the images. The images categories still cluster well into different areas of the space.

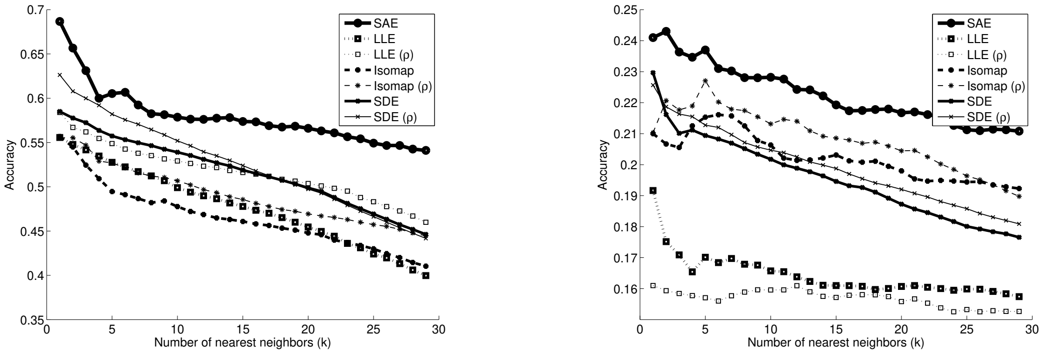

Figure 3. kNNA results for the ALOI (left) and Caltech (right) data sets.

3.2 kNNA Results

For more quantitative results, we ran SAE, Isomap, LLE and SDE embedding algorithms on 10 randomly chosen objects from the ALOI data set, using 30 images for each object. We set $m = 50$ cluster centers for grouping the SIFT features. We stopped the SAE algorithm when the change in T was below 10^{-6}. The embedding dimension D was set to 2. We then calculated the kNNA accuracy of the derived embedding for each value of the number of neighbors k. We ran 10 independent experiments (each with random objects), and reported the average kNNA values for all the algorithms. With the same settings, we ran similar experiments on the Caltech101 data set with 10 randomly chosen categories, each with 30 images. The results are shown in Figure 3. Results of algorithms with pairwise correlation as similarities are also shown in the figure (with label ρ).

4 Conclusions

Our algorithm achieves higher accuracy results than any of the other three algorithms. We believe this is because our algorithm is designed to cluster images with similar properties (as opposed to find parameters of continuous variation). Although our method works for continuous parameter variation (see Figure 1) and the dimensionality reduction algorithms have some success in clustering images (see Figure 3), our algorithm's strength is in embedding heterogeneous sets of images in which there may not be any continuous path of images leading from one member of the data set to another. For example, there is no "axis of variation" that one can vary to generate a natural smooth set of images from a butterfly image to a cellphone image in the CalTech101 data set. The butterflies and cellphones occupy disconnected regions of the space of images of interest. For many applications on data sets like Caltech101, we think this strength is particularly important.

Finally, our method has the advantage of being able to generate grid-based embeddings. For some applications, this is crucial to the utility of the embedding.

References

[1] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. In *CVPR, workshop on Generative-Model Based Vision*, 2004.

[2] J. M. Geusebroek, G. J. Burghouts, and A. W. M. Smeulders. The Amsterdam library of object images. *Int. J. Comput. Vision*, 61(1):103–112, 2005.

[3] M. G. Kendall. *Rank Correlation Methods*. Hafner Publishing Co., New York, 1955.

[4] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[5] G. Mei and C. R. Shelton. Visualization of collaborative data. In *UAI*, pages 341–348, 2006.

[6] B. Moghaddam, Q. Tian, N. Lesh, C. Shen, and T. S. Huan. Visualization and user-modeling for browsing personal photo libraries. *Int. J. Comput. Vision*, 56(1–2):109–130, 2004.

[7] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *CVPR*, volume 2, pages 2161–2168, June 2006.

[8] L. K. Saul and S. T. Roweis. Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *JMLR*, 4:119–155, 2003.

[9] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

[10] K. Q. Weinberger and L. K. Saul. Unsupervised learning of image manifolds by semidefinite programming. *Int. J. Comput. Vision*, 70:77–90, 2006.