Semi-supervised Learning based on Bayesian Networks and Optimization for Interactive Image Retrieval

Mai Yang[1+], Jian Guan[1+], Guoping Qiu[1#] and Kin-Man Lam[2]

[1] School of Computer Science and Information Technology, The University of Nottingham, Nottigham, NG8 1BB, UK

[2] Department of Electronic and Information Engineering, The Hong Kong Polytechnic University

Abstract

In this paper, we present a novel interactive image retrieval technique using semi-supervised learning. Recently, Guan and Qiu [8, 9] have shown that by constructing a Bayesian Network where the nodes represent the (continuous) class membership scores and arcs represent the dependence relations of the data points, the (semi-supervised) classification problem can be formulated as a quadratic optimization problem; and by using the labeled data as linear constraints, the optimization problem yields a large, sparse system of linear equations which can be solved very efficiently using standard methods. In this work, we show that this semi-supervised learning method can be naturally adopted as a computational tool to incorporate users feedbacks for interactive image retrieval. We present experimental results to show the effectiveness of our new interactive image retrieval method. We also show that semi-supervised learning can have advantages over supervised and unsupervised learning in image retrieval applications.

1 Introduction

In many machine vision applications, it is often very difficult or maybe even impossible to develop fully automatic solutions. For example, despite much research effort, a fully automatic solution to the longstanding image segmentation problem is still an unattainable goal. Other examples where a fully automatic solution is difficult include content-based image retrieval (CBIR).

Humans have remarkable abilities in distinguishing different image regions or separating different classes of objects. Furthermore, users may have different intentions in different application scenarios. Therefore, it is both necessary and helpful to incorporate high-level knowledge and human intentions into the computational algorithms. Interactive approaches, which provide semi-automatic solutions, put the users in the computational loop and allow users to supply constraints to the computational algorithms interactively, may offer a more realistic solution paradigm for

---

[+] These two authors have contributed equally to this work.

[#] Corresponding author (`qiu@cs.nott.ac.uk`)

many computer vision problems. Examples of such solutions include interactive image segmentation [4, 5, 8, 9] and relevance feedback in CBIR [10].

One of the important challenges to developing successful vision algorithms is to effectively model high level knowledge and to incorporate the users intentions in the computational algorithms. Traditionally, this is often achieved by integrating statistically learned prior knowledge into numerous computational algorithms through techniques such as Bayesian Inference [11]. Although the Bayesian approach has been successfully used in the literature, the resulting combinatorial optimization problems have been often solved by inexact and inefficient computational methods. In this paper, we present a new computationally simple optimization-based framework for directly incorporating priors (user inputs which provide both high level knowledge and user intentions) into the computational algorithm for interactive content-based image retrieval.

2 Semi-supervised Learning based on Bayesian Networks

Consider a given dataset consists of N samples $\{x_1, x_2, \ldots, x_N\}$. For simplicity, we consider the case where these samples come from two classes. For each sample, we assume there is an associated membership score, $\{(x_1, \alpha_1), (x_2, \alpha_2), \ldots, (x_N, \alpha_N)\}$, where $0 \le \alpha_i \le 1$ is interpreted as follows: if $\alpha_i = 1$, then $x_i \in$ class 1; if $\alpha_i = 0$, then $x_i \in$ class 0. The smaller α_i is, the more likely $x_i \in$ class 0; conversely, the larger α_i is, the more likely $x_i \in$ class 1. Now consider the case where some of the samples are labeled, i.e., $\alpha_i = y_i$, for $i = 1, 2, \ldots L$, where $y_i \in \{0, 1\}$ is the class label of x_i. The rest $\alpha_{L+1}, \alpha_{L+2}, \ldots, \alpha_N$ are unknown. Our task is to assign membership scores to those unlabeled data.

For a given metric in the feature space, $d(x_i, x_j)$, which measures the similarity of data x_i and x_j, our classification model makes a basic assumption that in the feature space, similar examples should be classified similarly[1]. However, unlike traditionally approaches that use the labeled data only to construct a classifier for classifying unlabelled data, we take an approach sometimes referred to as semi-supervised learning [1] by exploiting both labeled and unlabeled data in the construction of the classifier. Even though a fair amount of work has already been done [1 - 3], semi-supervised learning is a fairly new area. In [8, 9], two of us (Guan and Qiu) have proposed a semi-supervised learning based approach for the image matting problem, where they have established a link between semi-supervised learning and Bayesian Network, and shown that semi-supervised learning can be formally formulated as a linearly constrained quadratic optimization problem which can be solved very efficiently using linear methods. We here adopt this method for general data classification.

2.1 Construct a Bayesian Network

Let α_i defined above be the probability that a certain data point x_i belongs to class 1. Let all data points that can affect the state of x_i form a set S_i and call it the *neighborhood* of x_i. In a Bayesian Inference framework, we can write:

$$a_i = pr(x_i \in class_1) = \sum_{j \in S_i} pr(x_i \in class_1 \mid x_j \in class_1) pr(x_j \in class_1) \quad (1)$$

Define $w_{ij} = pr(x_i \in class_1 \mid x_j \in class_1)$, then we have

[1] Most classification algorithms are based on such assumption. However, this assumption could be invalid in some inappropriate spaces or for some inappropriate metrics. In these cases, the problem becomes more complicated and is beyond the scope of this paper

$$a_i = \sum_{j \in S_i} w_{ij} a_j \qquad \text{and} \qquad \sum_{j \in S_i} w_{ij} = 1 \tag{2}$$

By making a mild assumption that the given data follow the Gibbs distribution [11], the conditional probabilities can be defined as follows

$$w_{ij} = \frac{1}{Z} e^{-\beta d(x_i, x_j)} \qquad \text{where} \qquad Z = \sum_{j \in S_i} e^{-\beta d(x_i, x_j)} \tag{3}$$

where β is a positive constant, $d(x_i, x_j)$ is a metric function, and $w_{ij} = 0$ for $j \notin S_j$.

These definitions can be interpreted in a Bayesian Network framework where the nodes represent α_i's, and the arcs connecting two nodes x_i and x_j represent w_{ij}'s. The task of classifying the data is therefore to compute the states of the nodes of the Bayesian Network, α_i's.

2.2 Classification Cost Function

To make the 2nd equality of (1) holds, one need to collect all data points that will have an influence on a given data point. Since the given data set cannot be infinite and it is impossible and computationally unacceptable to find all data that will have an influence on a given data x_i, we cannot make the equality of (1) hold exactly. The best we can do is to make the quantities on both sides of equation (1) as close as possible, or equivalently, use the expression of (2), to make $\sum_i |a_i - \sum_j w_{ij} \alpha_j|$ as small as possible. Therefore, the classification problem can be formulated as the following quadratic optimization problem:

$$\alpha = \arg\min \left\{ \sum_i \left(\alpha_i - \sum_{j \in S_i} w_{i,j} \alpha_j \right)^2 \right\} \tag{4}$$

It is not difficult to see that optimizing the cost function in (4) means that, if a sample and its neighbors are similar, the optimization will favor them to have similar membership scores. Conversely, if a sample and its neighbors are different, the optimization will favor them to have different membership scores. Therefore, the optimization formulation satisfies the nearest neighbor assumption that similar samples should be classified similarly, i.e., assigned similar membership scores.

Unlike totally unsupervised classification such as k-NN, our data is partially labeled. It is important to note that even though a given sample (both labeled and unlabelled) is only linked to its neighbors, all samples are linked by a fully connected graph (it is always possible to construct a fully connected graph and we will present some possible methods in Section 4. Optimizing the cost function (4) is therefore a global process in the following sense. The labeled membership scores through their connection weights influence the membership scores of the unlabeled data. Furthermore, the membership scores of the unlabeled data are not only affected by the labeled data, but also influenced by other unlabelled data. In this way, we have brought both the labeled and the unlabeled data into the construction of the classification model.

2.3 Solving the Optimization Problem

To solve the optimization problem in (4), we use the labeled data membership scores as constraints and solve for the unknown membership scores. For the labeled data points, we have $\alpha_i = y_i$, for $i = 1, 2, \ldots L$, where $y_i \in \{0, 1\}$. Because the cost function is quadratic and the constraints are linear, the optimization problem has a unique global minimum. It is straightforward that the optimization problem yields a large, spares linear system of equations, which can be solved efficiently using a number of standard solvers [12]. Therefore, the formulation of the classification problem in an optimization framework has yielded simple and efficient computational solutions.

2.4 Relation to Previous Work

The optimization problem of (4) can be viewed as belonging to a family of recently proposed algorithms that optimize a cost function of the form in (5) to solve problems in nonlinear dimensionality reduction [16], image matting [6, 9] and segmentation [8, 13], colorization [7], and data classification [1 - 3].

$$E(\alpha) = \alpha^T (I - W)^T (I - W)\alpha \tag{5}$$

The locally linear embedding (LLE) framework for nonlinear dimensionality reduction [16] maps high dimensional inputs to low dimensional outputs by minimizing a cost function of the form of (5). LLE performs the optimization by solving a sparse eigenvalue problem. The normalized cut algorithm for image segmentation also solves a similar optimization problem as (5). More recently, the same framework was extended to grouping with bias where labeled data are used as grouping constraints [13].

Recently, interactive image editing have attracted significant interests [4 - 9]. The interactive image matting method of [6] solves a similar optimization problem of the form of (5) using random walk. The work in [8, 9] formulates interactive image segmentation and matting as an optimization problem of the form of (5) and solves a large, sparse linear system of equations. Similarly, [7] solves an optimization problem by solving a large, sparse system of linear equations for colorizing black and white photographs.

In terms of both application and the form of the cost function, our current method is more closely related to a class of semi-supervised learning methods based on graph cut [1 - 3]. Whilst most of these methods formulate the cost functions that are in one form or another similar to (5), perhaps that work that is most similar to ours is that of [3]. Whilst other graph cut based semi-supervised learning performs binary classification, the method of [3], similar to what we are doing here, performs a "soft classification".

Whilst our current method is related to the above (non-exhaustive list) methods in the literature, there are some differences, which are worth mentioning. Firstly, we formulate the problem from different intuitions. Whilst most of these reviewed methods "choose" a cost function of the form (5), we derive our solution formally in a Bayesian Network framework, which may offer new interpretations of and insights into this type of now increasingly popular techniques. Secondly, we set out in the outset that we want to assign a continuous membership score for each unlabelled data and formulate our cost function to satisfy the premise that similar data (based on some metric) should be classified similarly. Specifically, we want to assign similar data points with similar membership scores. Thirdly, we formulate the problem as a linearly constrained quadratic optimization problem which guarantees to have a unique global minimum. We also propose to use the labeled data as linear constraints and to obtain our solution by solving a large, sparse system of linear equations, which can be implemented using a number of efficient standard solutions. Therefore, we have formulated the problem such that it can be solved using simple standard methods. Fourthly, as will become clear in the next section, whilst others have hard classification in mind, we want to exploit the continuous membership scores directly. Fifthly, the formulation of equation (4) can be easily extended to perform semi-supervised multi-class classification as has been shown in [8].

2.5 Connection to Linear Neighborhood Embedding

In [17], two of us (Qiu and Guan) developed a method termed Linear Neighborhood Embedding (LNE) for colorizing black and white images. It is interesting to note that the computational technique of LNE is very similar to that developed here, except the way in which the neighborhood weights, w's, are calculated.

Following the definition at the beginning of this section, if we make a different yet well-known assumption that data points lie on the same low-dimensional manifold should share the same membership score, the classification problem can be modeled and solved using LNE as follow. For a given data point x_i, we can find its neighbors, S_i, according to some metric. Based on the manifold assumption, x_i can be linearly reconstructed using its neighbors. Using the computational technique detailed in [16, 17], we can calculate the linear reconstruction weights by minimizing the reconstruction error, i.e.

$$w = \arg\min \left\{ \sum_i \left(x_i - \sum_{j \in S_i} w_{i,j} x_j \right)^2 \right\} \tag{6}$$

Given these neighborhood weights, we embed them into the membership score space and result in the same quadratic optimization problem as (4). Figure 1 shows the effectiveness of LNE in solving semi-supervised data classification problem for a toy dataset similar to that used in [18]. We will address the behavior of LNE on data classification, e.g. how the classification performance affected by the penalty used for solving (6) (see [16] for more detail) and the neighborhood size in another paper. We are also working towards unifying these 2 methods.

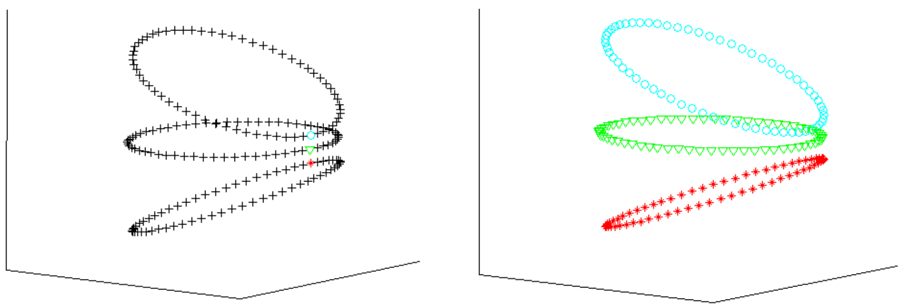


Figure 1: Left: Data points sampled from 3 circles in 3-D space, where the black '+' symbols are un-classified points and the 'o', '*', 'Δ' represent 3 labeled data points for 3 different classes respectively. Note that the distance between adjacent circles is smaller than the distance between two neighboring points on the same circle. **Right:** Classification result using LNE.

3 Interactive Image Retrieval using Optimization based Semi-supervised Learning

Content-based image retrieval is an area of considerable interest in the research community [10]. In CBIR, a querying example is first used to retrieve images that are "similar" in some way to the querying example. However, computational similarities, often measured as some metric distances between low-level features of images, and the perceived similarities, which are high-level concepts, often mismatch. In order to retrieve images that the users are looking for, we have to incorporate users intentions into the computational algorithms. One way to include users input in the computational algorithm is to use relevance feedback [10], where users interact with the computer to supply the algorithms with positive and negative examples to refine or retrain the algorithms so that results that better match the users intentions can be retrieved. These semi-automatic approaches, which put users in the retrieval loop, may offer more realistic solutions to the CBIR problem. Obviously, one important technical problem is how to capture and model humans' intentions in the computational algorithms. In this section, we propose to use the optimization based semi-supervised learning approach developed in Section 2 to perform this task.

With the technique of Section 2, it is relatively straightforward to perform relevance feedback for CBIR. The procedure can be described as follows:

Step 1: Find an initial querying example and perform example based image retrieval using one of many traditional approaches, such as ranking the images' similarity to the querying image according to their low level features distances [10].

Step 2: From the first N returned images, users identify L_1 positive examples (setting their membership scores α_i =1, i =1, …, L_1), and L_2 negative examples (setting their membership scores α_i =0, i = L_1+1, …, $L_1 + L_2$). These $L = L_1 + L_1$ samples are used as labeled samples.

Step 3: Perform semi-supervised learning on the N returned images based on (4) and using the L user labeled samples as constraints to solve the optimization problem by performing solutions to a large, sparse system of linear equations.

Step 4: Rank the computed unknowns α_i, i = L_1+ L_2 +1, L_1+ L_2 +2, …, N, in decreasing order and return the image with the highest membership score first and image with the lowest membership score last.

Step 5: If the desired image is found, then stop, if not, then label more examples and repeat Steps 3 & 4.

Note that the membership scores themselves may provide important information for the users to strategically pick effective samples to perform relevance feedback. The strategy is as follows: (i) label the unwanted images with the largest membership scores as the negative examples; (ii) label the wanted images with the lowest membership scores as the positive examples.

4 Experimental Results

We have implemented the optimization based semi-supervised learning method for interactive image retrieval in MATLAB. We performed our experiment using a color photograph database often used in CBIR literature [14]. The database is a subset of the popular Coral color photo collection and contains 1000 images divided into 10 categories each containing 100 images.

To represent the images, we used a scheme as illustrated in Figure 2. We first used sampling windows of various sizes and orientations to sample the image at random locations. Let $SW(m, n, \theta, x, y)$ be the sampling window of size $m \times n$ pixels, orientation θ and centered at (x, y) co-ordinate of the image $I(u, v)$ which will sample blocks $B(m, n, \theta)$ that are of $m \times n$ pixels and oriented at θ. These blocks are then first rotated by $-\theta$ and then scaled to a uniform size of $l \times l$ (4×4 in all experiments). In the experiments, we used sampling window sizes ranging from 1 pixel to the size of the image and 8 orientations $\theta = 45k$, k = 0, 1, 2, …7. For each image, we randomly choose the block sizes and orientations and sample 8000 blocks at random locations. We then used the coloured pattern appearance model (CPAM) [19] to compile a 64-d CPAM histogram (32 achromatic patterns and 32 chromatic patterns) for the 8000 blocks to represent the image content.

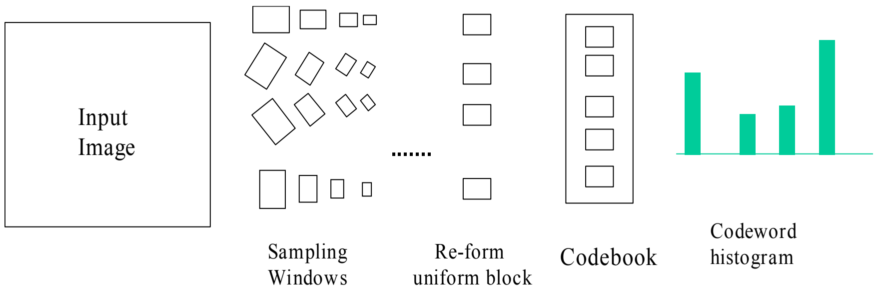


Figure 2: Derive image representation features using windows of various sizes and orientations to sample the image at random locations. The randomly sampled blocks are then re-formed into a uniform size and orientation. A codebook is designed to encode the uniform-sized blocks to compile a histogram of the blocks to represent the image.

For our semi-supervised learning approach, we used the Euclidean distance as the metric to construct the graph. Take each image as a vertex. We define the neighborhood S_i of a certain image i by finding K_1 nearest neighbor images from the unlabelled set and K_2 nearest neighbor images from the labeled set. The weighted edge between vertex i and j, $w_{i,j}$ was computed as (3) and the scaling constant β was set as the inverse of the variance of all feature variables in S_i. Note due to the normalization, $w_{i,j} \neq w_{ji}$, that is, the graph we construct here is a directed one.

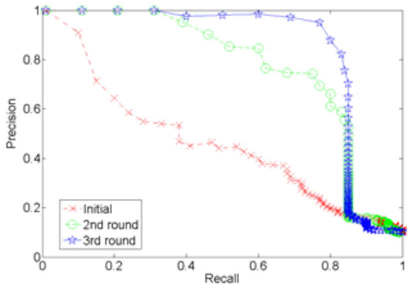
The *labeled neighbors* serve an important role for the robustness of our method. Unlike in many other applications, e.g. in image segmentation/matting [5, 6, 8, 9], where the graph is guaranteed to be fully connected because the pixels are geometrically connected to each other, thus each pixel is connected to a labeled one. In our case for CBIR, the graph constructed without the idea of *labeled neighborhood* could have many isolated components. If an isolated component is not connected to a labeled vertex, it is easy to understand that the decision made on these set of vertices is arbitrary because no prior knowledge is introduced. A more formal explanation is that the Laplacian matrix for such a graph cannot be inverted because it is rank deficient. Note in all experiments we set K_2=1.

4.1 Interactive Image Retrieval Experiments

We first performed relevance feedback experiments using the features derived according to Figure 2 and the procedures as described in Section 3. We use the standard Recall/Precision Curve (RPC) to measure the performance. Figure 3 shows the RPC's of querying the database using two randomly selected initial querying images. It is seen that using our semi-supervised learning as a computational tool for relevance feedback can dramatically improve the retrieval results.

4.2 Comparison with K-NN

The semi-supervised learning and the classical K nearest neighbor classifier are closely related. In this experiment, we compare the performances of our optimization-based semi-supervised learning and the K-NN classifier for the 1000 image database. We performed leave L out experiments. For each category, we left L image out as testing data, and used the 100-L as labeled positive samples and all images in other categories as labeled negative sample. For each category, we repeated the experiments several times by leaving different images out as testing samples. We measure the average classification errors over all experiments and results are shown in Table 1 for various values of L and a neighborhood size of 4 used both in the semi-supervised learning and for the K-NN classification. Table 2 shows the classification error rates for L = 25 and various neighborhood sizes for the K-NN and semi-supervised learning classifiers. It is seen that with the semi-supervised learning, classification performance can be improved significantly over unsupervised K-NN. Note both methods used the same neighbors in the classification.



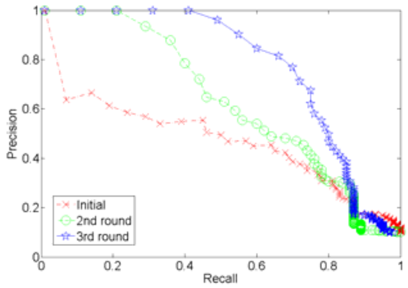


Figure 3: Recall Precision Curve (RPC) of interactive image retrieval using the optimization based semi-supervised learning. The initial round is based on the distance between the querying image and the database images. Relevance feedback for the 2nd round and 3rd round of interactions is based on user selection of 4 positive and 4 negative samples as labeled data for the semi-supervised learning. **Left:** results for a query for the "horse" category. **Right:** results for a query for the "bus" category.

# Testing samples, L	K-NN Error rate	Semi-Supervised Error rate
10	47.3%	34.1%
20	51.2%	36.5%
25	53.1%	37%
50	61.9%	43.5%
70	74.9%	52.3%

Table 1: Average classification errors for the K-NN and the semi-supervised learning for various numbers of labeled data and a fixed neighborhood size of 4.

# Neighbors, K_1	K-NN, Error rate	Semi-Supervised Error rate
4	53.1%	37%
8	52.9%	37.3%
16	58.7%	37.1%
32	66.5%	37%
48	70.4%	37.3%

Table 2: Average classification error rates for L = 25 and various neighborhood sizes for the K-NN classifier and the semi-supervised learning classifier. It is intersting to note that the semi-supervised learning is insensitive to the neighborhood sizes.

4.3 Comparison with Support Vector Machines

In this experiment, we try to evaluate how the semi-supervised learning method of this paper compared against a popular supervised machine learning method, the Support Vector Machines. In this experiment, we used a face/non-face image database. Our database consists of 2245 face and 6241 non-face image patterns of various resolutions. We first scaled all the images to 32 x 32 blocks. We then reduce the dimensionality of each image to 128-d using principal component analysis (PCA). We have performed two sets of experiments. In the first experiment, we used 1734 face and 4447 non-face images as labeled data to train a semi-supervised classifier and a support vector machine, and tested the classifiers on the rest (511 face and 1794 non-face) images. In the second experiment, we reversed the training testing samples used in the first experiment. We used 511 face and 1794 non-face as labeled data and tested on 1734 face and 4447 non-face images. For the SVM, we used an implementation downloaded from [15] and have chosen the Gaussian kennel to perform the experiments. We again used the recall precision curves (RPC's) to compare the performances and also plotted the receiver operating curve (ROC) for the methods. Results are shown in Figure 4. These results show that in this particular application, the semi-supervised learning outperforms Support Vector Machines.

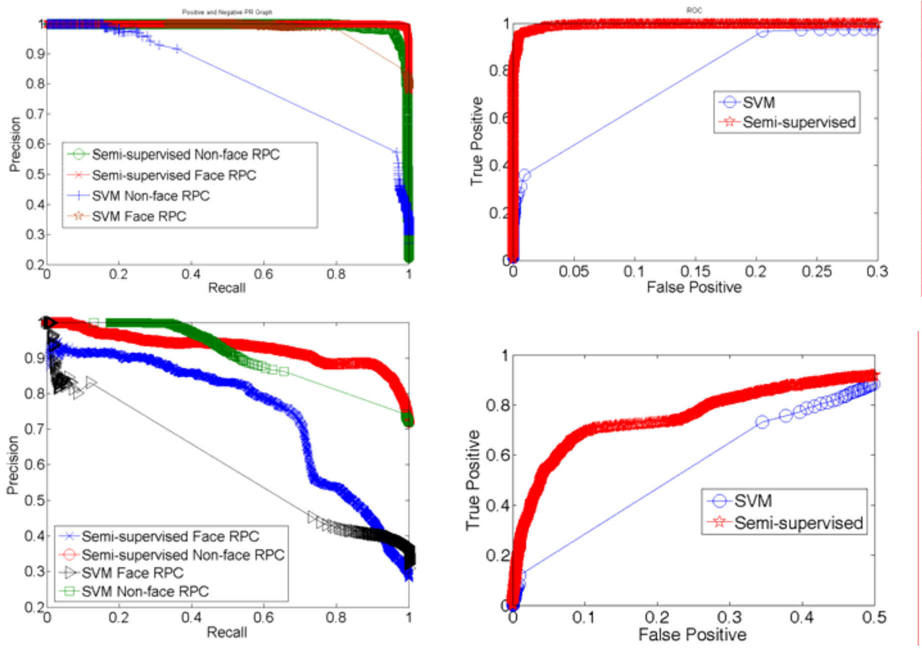


Figure 4: The Recall Precision Curves (RPC's) and the Receiver Operating Curves (ROC's) for the support vector machines and our optimization based semi-supervised learning. Top row is the results of the first experiment and the bottom row is the results of the second experiment. Left column: RPC's. Right column: ROC's. For the semi-supervised learning, a neighborhood size of 4 was used.

Another advantage of our optimization based semi-supervised learning over SVM is that our method is computationally very simple. We implemented our method using Matlab's direct least square solver without any optimization and the SVM used was the SVM-KM Toolbox [15]. The tests were run on an Intel Pentium 4 1.8GHz PC with 2GB RAM. Our method was on average about 30% ~ 50% faster than this version of SVM. With state-of-the-art solvers, e.g. multigrid, our method can be implemented more efficiently with linear complexity.

5 Concluding Remarks and Future Work

In this paper, we have shown that the optimization based semi-supervised learning method developed in our previous work [8, 9] can be naturally adopted as a computational tool to incorporate user feedbacks for interactive image retrieval. We have presented experimental results which demonstrate that our new interactive image retrieval method works effectively. We have also presented experimental results which provided some evidence to show that semi-supervised learning can outperform traditional classifiers such as K-NN and SVM in some applications.

Semi-supervised learning is still a very young discipline, more work is needed to develop the field and to gain deeper understanding how it relates to and compares with established classification methods such as K-NN and SVM.

In the past, very little work has been done to apply this new learning paradigm to content based image retrieval. Because of its ability to incorporate priors into the computational process, we believe semi-supervised learning is naturally suited for interactive image retrieval. This paper has provided some initial evidence of this. However, this is just the beginning and there is much work to be done. In our future work, we will continue developing interactive image retrieval methods using semi-supervised learning algorithms. We will also investigate how our optimization based semi-supervised learning relates to other similar methods and extend these methods to solving problems in computer vision, pattern recognition and multimedia classification and management.

References

[1] X. Zhu, *Semi-supervised learning literature survey*, Computer Science TR 1530, University of Wisconsin – Madison, February, 2006

[2] A. Blum, J. Lafferty, M. R. Rwebangira, and R. Reddy, *Semi-supervised learning using randomized mincuts*, ICML-04, 21st International Conference on Machine Learning, 2004

[3] X. Zhu, Z. Ghahramani and J. Lafferty, *Semi-supervised learning using Gaussian fields and harmonic functions*, ICML-03, 20th International Conference on Machine Learning, 2003

[4] Y. Boykov and V. Kolmogorov, *An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision*, IEEE Transactions on PAMI, Vol. 26, 1124-1137, 2004

[5] C. Rother, V. Kolmogorov, and A. Blake, *GrabCut: interactive foreground extraction using iterated graph cuts*, ACM Trans. Graph. 23(3): 309-314, 2004

[6] L. Grady, T. Schiwietz, S. Aharon and R. Westermann, *Random Walks for Interactive Alpha-Matting*, Proceedings of VIIP 2005, September, Benidorm, Spain pp. 423-429, ACTA Press

[7] A. Levin, D. Lischinski, and Y. Weiss, *Colorization using Optimization*, SIGGRAPH, ACM Transactions on Graphics, 689-694, 2004

[8] J. Guan and G. Qiu, *Interactive image segmentation using optimization with statistical priors*, International Workshop on The Representation and Use of Prior Knowledge in Vision (In conjunction with ECCV 2006), May 13, 2006, Graz, Austria

[9] J. Guan and G. Qiu, *Interactive Image Matting using Optimization: A Bayesian Network Approach*, Pacific Graphics 2006, Taipei, October 11-13, 2006

[10] A. W. M. Smeulders et al, *Content-based image retrieval at the end of the early years*, IEEE Trans PAMI, vol. 22, pp. 1349 - 1380, 2000

[11] D. Mackay, *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press, 2003

[12] W. H. Press, S. A. Teukolsky, et al, *Numerical Recipes in C++ - The Art of Scientific Computing*, Cambridge University Press, 2002

[13] J. Shi and J. Malik, *Normalized cuts and image segmentation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 22, 888–905, 2000

[14] James Z. Wang, Jia Li, Gio Wiederhold, *SIMPLIcity: Semantics-sensitive Integrated Matching for Picture Libraries*, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol 23, no.9, pp. 947-963, 2001. Database at: http://wang.ist.psu.edu/docs/related/

[15] S. Canu, Y. Grandvalet, V. Guigue and A. Rakotomamonjy, *SVM and Kernel Methods Matlab Toolbox*, Perception Systèmes et Information, INSA de Rouen, Rouen, France, 2005, http://asi.insa-rouen.fr/~arakotom/toolbox/

[16] L. K. Saul and S. T. Roweis, *Think Globally, Fit Locally: Unsupervised Learning of Low Dimensional Manifolds*, The Journal of Machine Learning Research, 4 119-155, 2003

[17] G. Qiu and J. Guan, *Color by Linear Neighborhood Embedding*, ICIP2005, IEEE International Conference on Image Processing, Genova, Italy, September 11 - 14, 2005

[18] J. Lim, J. Ho, M-H Yang, K-C Lee and D. Kriegman, *Image Clustering with Metric, Local Linear Structure and Affine Symmetry*, ECCV 2004, Prague, May 11-14, 2004

[19] G Qiu, "Indexing chromatic and achromatic patterns for content-based colour image retrieval", Pattern Recognition, vol. 35, pp. 1675 – 1686, August, 2002