Image Classification Using a Bigram Model

Rong Jin
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
1-412-268-4050
rong+@cs.cmu.edu

Alexander G. Hauptmann
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
1-412-268-1448
alex+@cs.cmu.edu

Rong Yan
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
1-412-268-9515
rongyan@cs.cmu.edu

ABSTRACT
Feature representation and classification algorithm are two important aspects for the task of image classification. Previous studies focused on using color histogram and/or texture as the features for representing an image. Support vector machines (SVM) have been among the most successful classification algorithms for image classification. In this paper, we examine a new type of representational feature, namely the 'bigram' feature, which computes a distribution for pixel pairs. Unlike color histogram based features, which treats each pixel as independent from others, the 'bigram' feature scheme is able to take the correlations between pairs of pixels into account. In experiments over six different image categories, the 'bigram' feature scheme appeared to be a better representation for image classification and achieved a better classification accuracy than either color histogram features, texture features or color correlograms.

Keywords
image classification, paired feature representation, bigram features, correlograms

1. INTRODUCTION
Image classification has been an interesting task for the computer vision and image processing community. The basic problem of image classification is to find an appropriate category for an image given a predefined set of categories. Many problems can be treated as a variant of the image classification problem. For example, the problem of object detection can be viewed as a special case of image classification where each type of object is treated as a different image category.

Recently, some machine learning techniques have been applied in order to solve the problem of image classification. The basic idea is that, by looking at positive and negative examples for an image category, the machine learning algorithm will learn a model from those examples and apply the learned model to predict the category for an unseen image. To accomplish it, we need to solve two issues: the first one is the representation issue, i.e. how to represent an image with a set of features and the second issue is the issue of classification, i.e. how to learn a model from the examples for an image category and then use the learned model to determine the appropriate category for an image.

For the choice of classification algorithm, support vector machines (SVM) [1] have been quite popular for the image classification task. This can be attributed to two reasons: First, to determine the category for an image, usually a large number of features are required. The SVM algorithm is able to take a large number of features for an image and combine them together to predict the category for that image. Unlike many other learning algorithms that tend to over-fit the training data when the number of features is large, SVM is able to alleviate the problem of over-fitting by maximizing the classification margin [1]. Secondly, the SVM algorithm allows for complicated nonlinear functions to be used as its kernel functions as long as they satisfies the Mercer condition [1], which can be very helpful in terms of exploiting the correlation between different features, while many other machine learning algorithms simply assume each feature is independent from the others.

The other important issue for the image classification problem is how to represent an image through a set of extracted features. Previous studies have focused on color histogram and texture features because of the belief that the image perception in human beings is strongly influenced by the color and texture distribution of the image [2, 3]. The color histogram feature scheme simply collects the number of pixels for every color bin and uses them as the features to represent the images. As a result of this simple process, the context of the pixel is not taken into account. More generally, the color histogram feature scheme doesn't take any correlation between different pixels into consideration. However, experiments in using color histograms as representation features for image classification have shown reasonably good performance [2], which appears to be consistent with the belief that color is a dominating component in determining human being's perception of an image [6]. In addition to color histogram features, texture features have been introduced to improve the classification accuracy. Other researchers found that the introduction of texture features significantly reduced image classification errors [3]. Since texture features are able to describe some form of localized pattern while the color histogram feature scheme pays no attention to the context of pixels, it is obvious that the introduction of texture features for an image can compensate what is missed in the color histogram features and therefore improve the ability of systems to predict the classification category for an image.

In this paper, we explore the correlation between pixels in a different way than through the notion of texture. Unlike the color histogram feature scheme, which treats every single pixel independently and only computes a distribution at the level of single pixels, we propose a new feature scheme, which we call the 'bigram' feature scheme that is able to compute the distribution of ***pixel pairs***. By counting the distribution for pixel pairs instead of features for single pixels, we are able to consider one type of correlation between pixels, i.e. the linking between pairs of pixels.

With the incorporation of this correlation into the representations for images, we may be able to improve the prediction accuracy for image classification.

To examine the effectiveness of our 'bigram' feature scheme, we conducted an experiment in image classification over six different image categories. By comparing the 'bigram' feature scheme to both simple color histogram features and texture features, we found that the 'bigram' feature scheme appears to be a good representation for images and achieved better classification accuracy than either the color histogram or the texture feature representation. To further demonstrate that this is not an accidental result, we alternatively used either support vector machine or maximum entropy as the classification algorithm and found that the results from both of these two classification algorithms indicate that the 'bigram' feature scheme is better than the other two representations.

The rest of paper is arranged as follows: Section 2 discusses the 'bigram' feature scheme for image classification. The experiment of comparing the 'bigram' feature scheme to color histogram and texture feature representation is presented in Sections 3 and 4. Section 5 draws the conclusions and also discusses potential future work.

2. THE 'BIGRAM' FEATURE SCHEME

As already pointed out in the introduction section, the problem with color histograms is that they treat each pixel as an independent unit and therefore the correlation between pixels is ignored. However, pixels exist in their context and without context these pixels may be completely meaningless. The texture features give a description of localized pattern and therefore are able to bring in some context for pixels. As a result, the introduction of texture features to color histogram feature scheme significantly boosts classification accuracy.

In this paper, we suggest a simpler way to explore the correlation between pixels. The basic idea to compute a distribution for pixel pairs instead of considering only a single pixel. By counting the distribution for pixel pairs, we are able to exploit the coupling between two pixels instead of treating each pixel independently.

One motivation for this work comes from work in speech recognition. In speech recognition, n-gram language models have long been used successfully to discard unlikely word sequences. The simplest case is a unigram model, which simply favors more frequent words over rare words. A slightly more sophisticated model is the bigram model, which is derived from the premise that to predict a word in a sequence, we should favor those words that are used frequently *immediately following the previous word* instead of frequent words used in any environment. Experiments in speech recognition have shown that bigram language models can achieve dramatic improvements in recognition over simple unigram models. The reason is that the probability of a particular word being spoken given a first word is very different from the probability of a word found in the language overall. E.g. while 'the' is a very frequent word in English, the probability that 'the' immediately follows 'the' in some text passage is extremely low. In other words, a bigram language model takes advantage of the correlation between the two neighboring words while a unigram treats each word as independently generated.

The other motivation of using a 'bigram' feature scheme to replace simple color histograms is that the 'bigram' feature scheme has higher representational power than color histograms. Consider two black white images with one image having the top half painted by black color and bottom half by white color and the other image having both black and white colors randomly painted throughout the whole image. For the simple color histogram scheme, it is impossible to distinguish these two cases since both of these two images have half of the pixels painted by both black and white colors respectively. Since the 'bigram' feature scheme is about a distribution for pixel pairs, it can tell one of these images from the other. The 'bigram' feature scheme will find for the first image, i.e. the image with the top half painted by black and the other half by white color, that it is much more likely to find a white pixel instead of a black one around a white pixel and vice versa. For the second image, i.e. the image with white and black color randomly distributed over the whole image, the 'bigram' feature scheme will find that it is equally likely to have a black pixel and a while pixel around either a white pixel and a black pixel. Furthermore, since different shape and texture can result in different distributions for pixel pairs, the 'bigram' feature scheme is endowed with the ability to represent different shapes and textures, at least to some extent.

Now, we need to discuss what kind of distribution is appropriate for pixel pairs. Formally, the color histogram can be viewed as a joint probability distribution between location L and color C P(L,C), i.e. the probability of finding a pixel with color C around location L. By assuming probability P(L,C) as the underlying generation model for an image, we also assume that each pixel is created independently without consulting the existence of other pixels.

For a 'bigram' feature scheme, we need to compute a distribution for pixel pairs. More formally, we would like compute the joint probability between two pixels P(C1, L1, C2, L2), i.e. probability of find a pixel pair with one pixel having color C1 at location L1 and the other having color C2 at location L2. One problem with using this joint probability is that it squares the number of features for color histogram and may results in too many features. In the following, we will discuss how to reduce the number of features for the 'bigram' feature scheme.

One dimension of reducing the number of features is to simplify the joint probability P(C1, L1, C2, L2). Instead of describing the pixel pair with their absolute location L1 and L2, we can to describe it using the relative position between the two pixels by assuming that the probability distribution has some sort of translation invariance. Let vector $\vec{r}$ stands for the vector from pixel 1 to pixel 2. Then, instead of computing P(C1, L1, C2, L2), we can compute P(C1,C2, $\vec{r}$), i.e. the probability of finding a pixel pair with one pixel having color C1 and the other having color C2 and the two pixels are separated by a vector $\vec{r}$. Since a vector can be represented as a combination of length and an angle, we can further write the probability as P(C1, C2, d, θ) where d and θ are the length and angle of the vector $\vec{r}$ respectively. To further reduce the number of parameters, we can decouple the correlation between the angle θ and the distance d and have two separated distributions for distance and angle. Therefore, the distribution for pixel pairs will be represented by the two sets of distributions P(C1,C2, d) and P(C1,C2, θ).

The other dimension of parameter reduction is to reduce the number of different distances d and angles θ. To accomplish that, similar to JPEG coding, we use a macro-block instead of single pixels where each block is 10 by 10 pixels and the dominant color within each block is assigned as the color for that block. The distance is quantized into 10 different bins and the angle is quantized into 8 different directions.

3. EXPERIMENTAL DESIGN

The goal of this experiment is to see the effectiveness of the 'bigram' feature scheme for image classification task. The image collections used in this experiment are extracted from the COREL photo CD library [7]. It consists of six different categories with a total of 423 images. The description of each category is presented in Table 1.

Table 1. Image categories of the 423 images collection

Category Name	Number of Images
Birds	73
Buildings & residence apartments	74
Fast foods	95
Fishes	40
Fruits & vegetables	38
Skies & Clouds	103

As seen from table 1, the six categories used in the experiment are more abstract than the 14 image categories used in previous studies [2, 3]. As already described in other literature [4], the classification accuracy for abstract categories will be considerably lower than the accuracy for concrete categories.

The first classification algorithm used in our experiments was the support vector machine (SVM), which has been used in many studies on image classification. The basic idea of SVM is to find the decision boundary $\vec{w}$ that is able to separate the positive examples from the negative examples. More precisely, we would like all the positive examples $\vec{x}_p$ to be above the boundary, i.e. $\vec{w} \cdot \vec{x}_p > 0$ and all negative examples $\vec{x}_n$ to be underneath the boundary, i.e. $\vec{w} \cdot \vec{x}_n \le 0$. Since there may be many different classification boundaries satisfying these conditions, the SVM algorithm always chooses the one with the maximum margin. Furthermore, in cases that are not linearly separable, we can either replace the dot product between the classification boundary $\vec{w}$ and the instance $\vec{x}$ with a different kernel function K($\vec{w}$, $\vec{x}$) or allow some difficult training examples to be incorrectly classified.

To avoid the suspicion that the 'bigram' feature scheme may be only work with the SVM algorithm, we also repeated the experiment using a maximum entropy model (ME) [5]. Having similar strengths and advantages as the SVM algorithm, the ME model is able to combine a large number of pieces of evidence for class prediction without severely over-fitting the training data. Furthermore, unlike the SVM algorithm which originally only targets the binary classification problem (*is this instance a member of the class or not*), and requires extra treatment to convert the problem of multiple classes into a binary classification problem, the ME model is essentially a classification algorithm for the multiple class problems. The basic form expressing the ME model is $P(y \mid \vec{x}) = \exp(\vec{\lambda}_y \cdot \vec{x}) / \sum_{y'} \exp(\vec{\lambda}_{y'} \cdot \vec{x})$, where $\vec{\lambda}_y$ stands for the weight vector for class y. As indicated by the previous expression, the input feature vector $\vec{x}$ is first weighted by the vector $\vec{\lambda}_y$ and combined together for the prediction of class label y through an exponential form. To find the optimal weight vectors for the exponential form, we use the standard maximum likelihood estimation (MLE) approach to find a set of optimal weight vectors, which can accurately predict the class labels for the training data. The search for optimal weight vectors is accomplished using a conjugate gradient (CG) algorithm. More details about the maximum entropy model and the search algorithm for the optimal weights can be found in [5].

To test the effectiveness of our 'bigram' feature scheme, we compared it to the color histogram features and texture features as well as correlograms. In the implementation of our color histogram, every image is divided equally into 3×3 blocks and each block is represented as a vector of distribution over 16 color bins. In total, we obtained 144 features for every image. In terms of color representation, we used the gray-scale level of the pixels. The reason for choosing gray scale level instead of RGB or HSV color space is to enable a comparison to the texture features, which only use gray levels and the 'bigram' features where only the lightness of pixels is used. For the texture features, we use a program based on the convolution of the image with various Gabor Filters [8]. N filters are generated at angles from 0 to pi-(1/N). A histogram is computed for each filter. In our implementation, 6 angles are used and each filter output is quantized into 15 bins.

In the realization of our 'bigram' feature scheme, as described in Section 2, we compute the distributions P(C1,C2,d) and P(C1,C2,θ). Distance has been equally divided into 10 bins and the angle has been equally divided into 8 different directions. Again, the color representation for the 'bigram' feature scheme is the gray level of the pixels, because the extraction of texture features is based on the gray level of pixels and choosing gray level as the color representation for the 'bigram' feature scheme enables a fair comparison to the texture features. In order to avoid having too many features, we only use 5 different histogram bins for gray levels in the 'bigram' feature scheme. The correlogram scheme only used the P(C1,C2,d) distribution and not the angle.

4. RESULTS AND DISCUSSION

Table 2. Average errors in image classification with five folder cross validation for 2 classification techniques (Support Vector Machine and Maximum Entropy) and 4 representation schemes (Color Histogram, Texture, Correlograms and Bigrams).

	SVM	ME
Color Histogram	64%	61%
Texture	45%	50%
Correlogram	50.1%	53.2%
Bigram	42%	45%

The five folder cross validation is performed over all the six image categories and the averaged classification errors are listed in Table 2.

As indicated by Table 2, the results for both SVM algorithm and ME model indicate that texture feature scheme is better than the color histogram feature scheme and the 'bigram' feature scheme is better than the texture scheme. Notice that in our experiment, for the purpose of comparison, we used only the gray level as the color representation for color histograms, which results in a classification error of over 60%. This is different from most published experiments where either the RGB or HSV color space is used. Further experiments on our data with color histograms using the Munsell color space achieved much better performance than the one listed in Table 2. However, this would not be a fair comparison to the other representation schemes, since both texture features and 'bigram' features only use gray level as their color representation. Consistent with published reports, correlograms, which also included knowledge of the distance between colors, performed better than color histograms, but were far inferior to the bigram feature scheme, which represented both the distance and the angle between colors.

Considering the simplicity of the 'bigram' feature scheme versus the complexity of the texture extraction where a set of special filters is used, we were quite surprised by the fact that the 'bigram' feature scheme performs better than the texture features with a classification error rate 42% vs. 45% for the SVM algorithm and 45% vs. 50% for the ME algorithm. As already discussed in the previous section, in order to make the comparison to the texture features fair, we intentionally chose gray level as the limited color representation for the 'bigram' feature scheme. However, in general, the 'bigram' feature scheme permits the use of any color space for the color representation. In contrast, the extraction of texture features relies on the lightness contrast of pixels and it would be quite difficult to incorporate the true color into the texture features.

5. CONCLUSION AND FUTURE WORKS

In this paper, we proposed a new feature representation scheme, called the 'bigram' feature scheme, for image classification. Compared to color histogram features, the 'bigram' feature scheme computes a distribution for pixel pairs and therefore is able to capture the correlation between two pixels. In an experiment over six different categories of images, we compared the 'bigram' feature scheme to color histogram features and to texture features and found that our 'bigram' feature scheme outperforms both color histogram and texture in terms of averaged classification errors. Therefore, we conclude that the 'bigram' feature scheme is a good representation for the image classification task.

In this paper, for the purpose of fair comparison, we only used the gray level in the 'bigram' feature scheme. In future work, we plan to extend the 'bigram' feature scheme to the true color representations, such as RGB or HSV, where we expect even better classification accuracy.

Finally, it seems likely that a combination of representation schemes may be most effective for many classification tasks. The best representations schemes will probably combine several generic representation schemes, such as the ones discussed in this paper together with specialized derived features (e.g. lines, faces, etc) for specific classification tasks.

6. ACKNOWLEDGEMENTS

This work was partially supported by National Science Foundation under Cooperative Agreement No. IRI-9817496, and by the Advanced Research and Development Activity (ARDA) under contract number MDA908-00-C-0037.

7. REFERENCES

[1] C. J. C. Burges, A Tutorial on Support Vector Machine for Pattern Recognition, Knowledge Discovery and Data Mining, 2(2), 1998.

[2] O. Chapelle, P. Halffiner and V. N. Vapnik, Support Vector Machine for Histogram based Image Classification, IEEE Transaction on Neutral Network, Vol. 10, 1999.

[3] K. Goh, E. Y. Chang, K. T. Cheng, SVM Binary Classifier Ensembles for Image Classifier, CIKM 2001.

[4] O. Teytaud and D. Sarrut, Kernel based Image Classification, Lecture Notes in Computer Science, http://www.citeseer.nj.nec.com/496638.html, 2001

[5] S. D. Pietra, V. D. Pietra and J. Lafferty, Inducing Features of Random Fields, IEEE Transaction on Pattern Analysis and Machine Intelligence, 19(4), 1997

[6] Del Bimbo, A., Visual Information Retrieval, Morgan Kaufmann Publishers, San Francisco, California, 1999

[7] Corel Corporation (1999). Corel clipart & photos, http://www.*corel.com/*

[8] I. Fogel and D. Sagi. Gabor filters as texture discriminator. Biological Cybernetics, 61:103113, 1989