Classification of High-Dimension PDFs Using the Hungarian Algorithm

James S. Cope and Paolo Remagnino

Digital Imaging Research Centre, Kingston University, London, UK
{j.cope,p.remagnino}@kingston.ac.uk

Abstract. The Hungarian algorithm can be used to calculate the earth mover's distance, as a measure of the difference between two probability density functions, when the pdfs are described by sets of n points sampled from their distributions. However, information generated by the algorithm about precisely how the pdfs are different is not utilized. In this paper, a method is presented that incorporates this information into a 'bag-of-words' type method, in order to increase the robustness of a classification. This method is applied to an image classification problem, and is found to outperform several existing methods.

1 Introduction

For many machine learning problems, an object of data, e.g. an item to be classified, can be described as a single point within a feature space. Many different methods exist for classifying such objects, from simple methods, such as k-nearest-neighbour, to more sophisticated methods, such as support vector machines. However it is sometimes more appropriate to describe an object as a distribution within a feature space. A number of methods also exist for classifying data of this type. When described using histograms, the difference between two probability density functions (pdfs) can be calculated using bin-by-bin methods, such as the Jeffrey-divergence metric, however these methods encounter problems when the data has a high dimensionality, where a large number of bins makes the calculation expensive, whilst the sparse population of bins causes poor results. The earth mover's distance (EMD) [7] deals this by using signatures, and provides an accurate and intuitive measurement. These 'signatures' are weighted points within the feature space. This is akin to clustering data points drawn from the distribution, and weighting each cluster centroid by the number of points in the cluster. Another method is to use kernel density estimation to estimate a probability density function using points sampled from a distribution, and then to use this estimation to predict the probability of another sampling of points belonging to the same distribution. More recently, 'bag-of-words' methods have enjoyed increasing usage for this problem, particularly in the guise of 'bag-of-visual-words' [8] for image retrieval.

In this paper we utilize information generated in the calculation of the earth mover's distance in order to allow for more robust classification of pdfs, combining this with the strengths of the 'bag-of-words' method.

G.L. Gimel' farb et al. (Eds.): SSPR & SPR 2012, LNCS 7626, pp. 727–733, 2012.

In section 2 we describe the EMD and 'bag-of-words' methods in more detail. In section 3 a new bag-of-words method is described. A comparison of this methods to other common methods, using empirical results, is given in section 4.

2 Background

2.1 The Hungarian Algorithm and the Earth Mover's Distance

The earth mover's distance (EMD) [7] is a measure of the difference between two pdfs. The analogy is that, to reform one mound of earth as another, the effort required would depend on the sum the of distances that each unit of dirt must be moved. Whilst bin-by-bin methods only consider the amount of 'earth' in each location, the EMD considers how far it must be moved. There are two forms of pdf descriptions that allow the EMD to be calculated, histogram binning, and the aforementioned signatures. Since the binning is analogous to using evenly spaced signatures, we need only consider the latter.

Whilst there may be many ways of reforming one pdf into another, the EMD is calculated as being the one that requires the minimum total movement. The standard way of determining this is to model it as the transportation problem. There are a number of methods for solving the transportation problem, but by reforming the data so that each signature has an equal weight, it becomes equivalent to the simpler assignment problem, which can be solved using the Hungarian algorithm [5]. Whilst the original Hungarian algorithm was $O(n^4)$, an $O(n^3)$ version has since been found by Edmond and Karp [4].

The EMD only uses the minimum cost calculated by the Hungarian algorithm, but in our usage here we will also record the corresponding mapping between signatures, as it provides not only a measurement of the difference between the pdfs, but also information about in what way they are different.

2.2 The Bag-of-Words Model

The 'bag-of-words' model was originally used for the retrieval of text documents [9]. The idea was to represent documents as the frequency of occurrence of different words, and to find similar documents by comparing these frequencies. In recent years this concept has been extended to allow for the classification of more general forms of data. Typically, a large number of points are sampled from the training distributions and then a clustering is performed on these. The cluster centroids are used as the 'codewords' in a 'dictionary' used to perform a quantization of the data, by assigning each data-point to its nearest 'codeword'. A set of points from a distribution can then be described as the frequency of occurrence of each 'codeword'. This concept has seen much use recently in the field of computer vision, for tasks such as image retrieval [1,8] and texture analysis [6,10].

3 Methodology

We define the problem as follows. An object, X, is described by a set of n data points, $X = \{\bar{x}_1, \bar{x}_2, ...\bar{x}_n\}$, sampled from a distribution. Each data point $\bar{x}$ is a feature vector, $\bar{x} = [x_1, x_2, ..., x_d]$, where d is the number of features. Given a number of different classes, where class i is described by another set of n data points, $C_i = \{\bar{y}_1, \bar{y}_2, ...\bar{y}_n\}$, drawn from all objects in the training set that belong to the class, we wish to determine the class to which object X most likely belongs. This is calculated using Bayes theorem.

The method involves first generating a set of codewords from the training set, suitable for representing the data. All points in the training and class objects are assigned to their nearest codeword. A mapping is calculated between the data points in each training object and its corresponding class object. For each pair of codewords and each class, the probability is calculated of a mapping having its training object point assigned to the first of these codewords and its class point assigned to the second. For classification, the same codeword assignments and mappings are performed, and the previously calculated probabilities are used to determine the class which the object belongs to.

3.1 Generating a Vocabulary

Within the literature there has been much discussion on the appropriate methods for generating, and ideal size of, the codeword dictionary. The simplest approach is choose points evenly distributed throughout the feature space. The main disadvantage of this is that large portions of the space may not be used, resulting in redundant codewords, whilst other, more useful areas may receive inadequate representation. Another simple method is to use randomly selected points from the training data as the codewords. This largely eradicates the above problems, although using the centroids from a clustering performed on the training data normally provides a better representation. Another approach is to perform a separate clustering for each class and combining the generated codewords. This ensures that each class has some appropriate codewords, but may result in very similar codewords in the combined dictionary. We found that a k-means clustering of the whole training set produces an appropriate dictionary for our method.

There is no consensus on the size of a dictionary, with suggestions varying greatly, but for this method we found that, with objects described using 1024 point, a dictionary of size 256 produced good results, with larger dictionaries providing little or no improvement. We call the i^{th} codeword in the dictionary D_i.

3.2 Producing the Class Models

For each class, a class object is produced by randomly selecting n points from the class's example in the training set. For each training object, a mapping is found from its data points to those its class object using the Hungarian algorithm. This mapping pairs the points in one object to those in the other, such that the

sum of the squared Euclidean distances between paired points is minimised. We define the point in the class object C_i to which point $\bar{x}$ is paired as $M(\bar{x}, C_i)$.

Each point in the training data is assigned to its nearest codeword. For each class i, for each pair of codewords, (D_a, D_b), the conditional probability is calculated of a point $\bar{x}$ in that class's training data being assigned to codeword D_a, given that the corresponding point in the class object has been assigned to D_b. This is calculated as follows:

$$P(\bar{x} \in D_a | M(\bar{x}, C_i) \in D_b) \tag{1}$$

$$= \frac{P(\bar{x} \in D_a, M(\bar{x}, C_i) \in D_b)}{P(M(\bar{x}, C_i) \in D_b)} \tag{2}$$

where

$$P(\bar{x} \in D_a, M(\bar{x}, C_i) \in D_b) = \sum_{\substack{T_{ij} \in D_a \\ M(T_{ij}, C_i) \in D_b}} \frac{1}{|T_i|} \tag{3}$$

$$P(M(\bar{x}, C_i) \in D_b) = \sum_{d=0}^{|D|} P(\bar{x} \in D_d, M(\bar{x}, C_i) \in D_b) \tag{4}$$

where T_{ij} is the j^{th} point, $|T_i|$ is the total number of points in the training data for class i, $|D|$ is the number of codewords, and $\bar{x} \in D_a$ indicates that point $\bar{x}$ has been assigned to codeword D_a (likewise, $M(\bar{x}, C_i) \in D_b$ indicates that the point which $\bar{x}$ is paired with is assigned to codeword D_b).

Equation 3 calculates the probability of a point in D_a being mapped to a point in D_b as the fraction of training points for a class C_i for which this occurs. The probablity of a point, from any codeword, being mapped to one in D_b is then the sum of these for all codewords (equation 4).

3.3 Performing the Classification

To classify an object, we again assign all of the object's data points to determine to their nearest codewords. The object is mapped using the Hungarian algorithm to each of the class objects. We can then determine the class to which the object belongs using a Bayesian classifier.

$$c^* = \arg\max_i P(X|C_i)P(C_i) \tag{5}$$

$$P(C_i) = \frac{|T_i|}{\sum_j |T_j|} \tag{6}$$

$$P(X|C_i) = \prod_{\bar{x} \in X} P(\bar{x} \in D_a | M(\bar{x}, C_i) \in D_b) \tag{7}$$

4 Experiments

In this section the new algorithm is empirically evaluated by comparing it to a selection of other techniques. For these experiments we have 32 different classes, with 16 examples of each, performing a 16-fold cross validation. Each example's object has 1024 data-points. For the first method we use a dictionary of 64 codewords, and for the second method we use 16 clusters for each class.

To test the algorithm we apply it to a visual computing problem, the classification of plant species from images of their leaves. This is a problem which has received much interest recently [2]. For each leaf image in the database, we randomly select 1024 small windows. For each window we calculate 20 features based on the responses from different filters applied to all the pixels in the window. The set of features for each window becomes one of the objects data-points in a 20-dimension feature space.

4.1 Methods for Comparison

Three different methods are used for comparison:

1. Kernel Density Estimation - Kernel density estimation is used to predict the probability density function for each class. This estimate of the pdf is then used to calculate the likelihood of the object belonging to that class.

$$P(X|C_i) = \prod_{\bar{x} \in X} P(\bar{x}|C_i)$$
$$= \prod_{\bar{x} \in X} \sum_{\bar{y} \in C_i} \frac{\phi(||\bar{y} - \bar{x}||)}{|C_i|}$$

   where $\phi(x)$ is a normal distribution function with mean, $\mu = 0$ and standard deviation, $\sigma = 0.1$. This kernel function was used as it appeared to give the best results for the dataset.
2. Earth Mover's Distance - For the we use the pure value calculated by the earth mover's distance instead of utilizing the mapping between objects. Each object is classified as belonging to the class whose object is closest to it according to the EMD metric.
3. Naive-Bayesian Bag-of-Words - For the bag of words method, we use the same codeword dictionary as for the new method, to allow fairer comparison. We use a Naive-Bayes classifier, as it is both one of the most common classifiers used for bag-of-words [3], and is similar to that used in the proposed method.

4.2 Results

Table 1 gives the results for the proposed method, using different numbers of data points, and different dictionary sizes. The overall results of the experiments are given in table 2.

Table 1. Results for the proposed method, varying object and dictionary size (in %)

$\|D\|$	$n = 256$	$n = 512$	$n = 1024$
16	67.97	73.05	75.39
32	75.39	80.66	81.64
64	84.77	85.35	88.09
128	86.13	90.04	90.06
256	90.02	91.02	92.97

Table 2. Overall results, using best parameter values for each method (in %)

	n		
Method	256	512	1024
Proposed Method	90.02	91.02	92.97
Kernel Density Estimation	69.73	73.83	77.73
Earth Mover's Distance	73.83	79.88	85.35
Bag-of-Words	77.15	79.30	80.27

As the results show, the new method both performed far better than the standard bag-of-words method. This is because when the difference between pdfs means that points are assigned to different codewords, the standard method considers only that these points are no longer assigned to the same codeword, whereas the new methods both consider where in the feature space those points may exist, given that particular class. The kernel density estimation and earth mover's distance methods both performed worse than the other methods. These methods both directly compare samplings from distributions, and so are susceptible to noise produced by the sampling. The bag-of-words methods eliminate much of this noise, by quantisation via assignment to codewords.

Given that the EMD must be calculated in performing the new method, it may be possible to improve the results by incorporating the EMD metric. In our experience, however, doing so produced no change in the results. As would be expected, increasing the number of points used to describe objects increases the quality of the classification, but the new method still performs better than the other methods when a smaller number of points are used, making it particularly suitable when larger samplings are not practicle.

5 Discussion

In this paper a new method for the classification of high-dimension probability density functions has been proposed. The method utilizes the Hungarian algorithm to calculate mapping between sets of points sampled from PDFs. This information is incorporated into a 'bag-of-words' type method by calculating the probabilities of a pair of corresponding data-points being assigned to particular pairs of 'codewords'. This allows for more robust classification than the

traditional 'bag-of-words' method. For a visual object recognition problem the algorithm was found to perform significantly better than a number of existing techniques, achieving over 92% accuracy.

References

1. Chen, X., Hu, X., Shen, X.: Spatial Weighting for Bag-of-Visual-Words and Its Application in Content-Based Image Retrieval. In: Theeramunkong, T., Kijsirikul, B., Cercone, N., Ho, T.-B. (eds.) PAKDD 2009. LNCS, vol. 5476, pp. 867–874. Springer, Heidelberg (2009)
2. Cope, J.S., Corney, D.P.A., Clark, J.Y., Remagnino, P., Wilkin, P.: Plant species identification using digital morphometrics: A reviews. Expert Systems with Applications 39, 7562–7573 (2012)
3. Csurka, G., Dance, C.R., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: Workshop on Statistical Learning in Computer Vision, ECCV, pp. 1–22 (2004)
4. Edmonds, J., Karp, R.M.: Theoretical improvements in algorithmic efficiency for network flow problems. Journal of the ACM 19, 248–264 (1972)
5. Kuhn, H.W.: The Hungarian method for the assignment problem. Naval Research Logistics Quarterly 2, 83–97 (1955)
6. Leung, T., Malik, J.: Representing and recognising the visual appearance of materials using three-dimensional textons. International Journal Of Computer Vision 43, 7–27 (2001)
7. Rubner, Y., Guibas, L.J., Tomasi, C.: The earth mover's distance, multi-dimensional scaling, and color-based image retrieval. In: ARPA Image Understanding Workshop, pp. 661–668 (1997)
8. Sivic, J., Zisserman, A.: Google video: A text retrieval approach to object matching in videos. In: International Conference On Computer Vision, vol. 2, pp. 1470–1477 (2003)
9. Sparck-Jones, K., Needham, R.M.: Automatic term classifications and retrieval. Information Storage And Retrieval 4, 91–100 (1968)
10. Varma, M., Zisserman, A.: A statistical approach to material classification using image patch exemplars. IEEE Transactions on Pattern Analysis and Machine Intelligence 31, 2032–2047 (2009)