Sparse Multiscale Local Binary Patterns

Yogesh Raja and Shaogang Gong
Vision Lab, Department of Computer Science
Queen Mary, University of London
Mile End Road, London E1 4NS, U.K.
{jpmetal,sgg}@dcs.qmul.ac.uk

Abstract

In a Local Binary Pattern (LBP) representation, circular point features are taken in their entirety as predicates and restricted to uniform patterns with limited scales of small numbers of features in order to avoid large bin complexity. Such a design cannot fully exploit the discriminative capacity of the features available. To address the problem, this paper proposes (1) a pairwise-coupled reformulation of LBP-type classification which involves selecting single-point features for each pair of classes across multiple scales to form compact, contextually-relevant multiscale predicates known as Multiscale Selected Local Binary Features (MSLBF), and (2) a novel binary feature selection procedure, known as Binary Histogram Intersection Minimisation (BHIM) designed to choose features with minimal redundancy. Experiments show the advantages of MSLBF over traditional LBP representation and of BHIM over feature selection schemes such as AdaBoost.

1 Introduction

Local binary patterns (LBPs) have been used extensively for texture discrimination [6, 8], demonstrating excellent results and good robustness against rotation and global illumination changes. They have also been used successfully for texture segmentation [9] and recognition of facial identity [1] and expression [2, 11]. Currently, most LBP formulations involve the use of "uniform" patterns, which have been experimentally observed to correlate well with real-world structures [8]. These circular pattern features are collected from fixed circular neighbourhoods, each containing a relatively small number of interpolated samples. However, such N-bit "uniform" patterns restrict the domain of attention to the corresponding subspaces of the N-dimensional binary densities formed from the fixed circular neighbourhoods.

Uniform patterns are extracted for each scale of a multipredicate LBP operator [6] and the corresponding reduced-bin histograms appended together as a complete class descriptor (with non-uniform patterns collected in a single bin). When combined with rotation invariant transformations to reduce bin numbers further, $LBP^{riu2}_{P,R}$ provides very good classification results. However, this approach decouples the statistics between scales and suffers from restrictions in the size of the feature pool. Treating all predicates jointly would be computationally intractable as well as requiring infeasibly large training sets for reliable estimation. Additionally, the feature pools are often restricted to 3 scales with

BMVC 2006 doi:10.5244/C.20.82

at most 24 samples. Some work has addressed these issues, such as improvements in reducing complexity [6] and selection of most relevant complete patterns [11].

In this paper, an alternative and more effective method is proposed for generating LBP-type predicates, known as Multiscale Selected Local Binary Features (MSLBF). These MSLBF predicates are used in a pairwise-coupling [5] approach with multiple binary classifiers, one for each pair of classes, along with a scoring procedure to perform multiclass discrimination. Each classifier is a joint density generated from individual bit features selected from across scales in the training data. Different pairs of classes are modelled with features specifically relevant to those pairs.

A feature selection method is required for creating MSLBF predicates. A novel feature selection algorithm called Binary Histogram Intersection Minimisation (BHIM) is introduced. This is a relatively computationally inexpensive greedy feed-forward algorithm which is shown to often find less redundant feature sets from two-class binary data than two other fast algorithms. The selected features can be converted to a decimal value in the same way as circularly-arranged features in traditional LBP methods. Contrary to previous approaches, MSLBF codes need not derive from spatially continuous binary features. In principle, feature pools are generated in the same manner as for previous LBP methods, along with the same rotation invariant mechanisms if required. MSLBF classification and BHIM feature selection are presented in Section 2.

While training a model can take hours depending on the number of classes, the classification process generally requires around a second per input image, even with hundreds of model comparisons. Experiments are described in Section 3 with a suite of the Outex database [7] for textures and the ORL face database [10] where a direct comparison is made between a traditional LBP classifier and several MSLBF classifiers on the same data. It is shown that MSLBF models generated from multipredicate sample data outperformed traditional LBP models with very few features per pair of classes.

Comparisons are also made between BHIM, Conditional Mutual Information Maximisation (CMIM) [3] and AdaBoost [4] for effectiveness in feature selection. CMIM is designed to efficiently choose features which maximise their mutual information with the class variable while minimising their redundancy. AdaBoost is a widely known and developed method for generating strong classifiers by combining a set of weak learners found through a process of dynamically weighting training samples at each step. It is shown that BHIM was consistently more effective than these two alternatives in choosing compact, minimally redundant feature sets from binary training data without prohibitive computational expense. Conclusions are drawn in Section 4.

2 Multiscale Selected Local Binary Features

Multiscale Selected Local Binary Features (MSLBF) predicates are proposed as an improvement to traditional LBP models for LBP-type classification. Feature selection is used to generate these predicates which comprise individual point features from across multiple circular features at different scales. The selected features are treated as binary strings and converted to decimal values to represent individual samples, as with previous LBP methods. The resulting decimal histograms are used for classification by intersecting them with input histograms. Rather than a single multiclass classifier, there are multiple binary classifiers, one for each pair of classes. Each class is scored for an input according

to individual binary classifications. This form of reduction is known as "pairwise coupling" [5]. Here, a simple approach is taken to combining classifier outputs by updating the score of classes according to individual matches and assigning the input to the class with the highest score.

2.1 Binary Histogram Intersection Minimisation

Binary Histogram Intersection Minimisation, when provided with two binary data sets, attempts to find K bits from the total feature pool whose joint distributions for each of the two models are strongly divergent. More precisely, given two datasets P and Q constructed from random variables x_f corresponding to binary features indexed by f, $1 \leq f \leq F$, the objective of the algorithm is to find a set $B = \{b_1, b_2, ..., b_K\}$ where the b_ks are the indices of the selected features from the feature pool. Histogram intersection is employed in the scoring of features. These are only ever computed with two-bin binary histograms and so the "histogram distance" $\{1 - HI(p,q)\}$ given normalised histograms p and q for a binary feature x from datasets P and Q is computed more simply as $|p(x=1) - q(x=1)|$. Each b_k is selected as follows:

$$b_1 = \underset{f}{arg\,max} \ \left| p(x_f = 1) - q(x_f = 1) \right| \tag{1}$$

$$b_{k+1} = \underset{f \notin B'}{arg\,max} \sum_{\mathbf{x} \leftarrow \{0,1\}^k} S(\mathbf{x}, B', f) \tag{2}$$

where $B' = \{b_1, b_2, ..., b_k\}$, $k < K$ is the partial set of features selected so far and

$$S(\mathbf{x}, B', f) = p(\mathbf{x}_{B'}).q(\mathbf{x}_{B'}).\left| p(x_f = 1 | \mathbf{x}_{B'}) - q(x_f = 1 | \mathbf{x}_{B'}) \right| \tag{3}$$

The terms $p(\mathbf{x}_{B'})$ and $q(\mathbf{x}_{\mathbf{B}'})$, where $\mathbf{x}_{B'} = \{x_{b_1}, ..., x_{b_k}\}$, are the joint probabilities of occurrence of a specific instance of the binary vector $\mathbf{x}$ over the k previously selected features with indices $B' = \{b_1, b_2, ..., b_k\}$, for classes P and Q respectively. Similarly, the terms $p(x_f | \mathbf{x}_{B'})$ and $q(x_f | \mathbf{x}_{B'})$ are the normalised binary histograms for feature x_f conditional on the specific binary vector $\mathbf{x}$ over the selected features B'.

At step $k+1$ and for each feature $f \notin B'$, the algorithm computes equation 2, the expectation of the histogram distance between the datasets P and Q over the joint density for feature set B'. This involves at most 2^k values for $\mathbf{x}$, although only values present in both datasets need be included in the computation. Datasets containing features with strongly separating statistics will generally have far fewer shared values between them, requiring less computation to find. The algorithm stops when K features have been selected or no value of $\mathbf{x}$ has a positive probability for both models simultaneously, meaning their corresponding joint binary histograms have zero intersection. Intuitively, this algorithm finds features with maximally divergent two-bin binary densities when previously selected features fail to discriminate. It should be noted that histogram distance may be replaced with another measure, such as Kullback-Leibler distance.

2.2 MSLBF classification and computational cost

An MSLBF classifier is simply a list of pairs of histograms, each pair uniquely corresponding to a specific two of N classes. Consequently there are $\frac{1}{2}N(N-1)$ binary clas-

sifiers required for an N-class classification task. Each classifier $c_{P,Q}$ comprises a set of K selected feature indices $B_{P,Q}$ corresponding to their positions in the feature pool along with two 2^K-bin histograms corresponding to the joint densities over $B_{P,Q}$, one for each of the two classes P and Q. Given a set of training classes T_1 to T_N, the trainer cycles through all possible combinations of pairs of classes T_P and T_Q, $P \neq Q$ and calls the feature selection algorithm with the samples for those classes to generate $B_{P,Q}$. Adding classes is straightforward and requires N extra binary classifiers to be generated, one for each of the N classes against the new class indexed $N+1$. Each class n ($1 \leq n \leq N$) in an N-class problem has $N-1$ binary classifiers for comparing against each of the other $N-1$ classes.

Classification of an input involves keeping a score for each of the N classes. Since each combination of pairs of classes has a separate set of discriminative features, histograms are assembled for each of the $\frac{1}{2}N(N-1)$ binary classifiers $c_{P,Q}$ given their corresponding features. Each pair-specific input histogram is intersected with the two pair-specific model histograms. At this point, a scoring procedure is considered which involves updating the score of the highest match, adding the intersection value itself. After all binary classifications are performed, the class with the highest score is assigned to the input. Algorithm 1 provides pseudocode for the classification procedure.

With a Matlab implementation, classification of an input image for a 24-class texture experiment shown in Section 3.1 (involving 276 binary classifiers) required only a fraction of a second. Classification of an input for a forty-class face recognition task shown in Section 3.2 involving 780 classifiers required just over a second on average. The low number of bits for each classifier helped to keep computation time down both in terms of histogram assembly and comparison. The computation time for classifying an input is linear with the number of classifiers although the number of classifiers increases quadratically with the number of classes.

```
for P=1 to N-1 do
    for Q=P+1 to N do
        r=gen_hist(I,B(P,Q))
        s_P=hist_int(r,h(P,Q,P))
        s_Q=hist_int(r,h(P,Q,Q))
        if s_P greater than s_Q then
            score(P)=score(P)+s_P
        end
        if s_Q greater than s_P then
            score(Q)=score(Q)+s_Q
        end
    end
end
result=argmaxj (score(j)), j=1..N
```

Algorithm 1: The MSLBF classification procedure. $gen\_hist(I, B(P,Q))$ is a function returning the histogram for input data I given the features $B(P,Q)$ specific to the pair of classes P and Q. The $hist\_int(a,b)$ function computes the histogram intersection between two histograms a and b. The $h(P,Q,j)$ function returns the histogram corresponding to class j, $j = P$ or $j = Q$, from the binary classifier pairing classes P and Q.

3 Experiments

The BHIM algorithm was applied to generate MSLBF models for two different domains; texture recognition and facial identity recognition. The aim of the experiments was to investigate the improvement that can be gained from modelling jointly across scales (predicates) and the benefits of a larger feature pool for selecting more relevant features. Furthermore, the BHIM, CMIM and AdaBoost algorithms were applied for comparison in feature selection. A traditional LBP classifier was trained on both the texture data and the face data for comparison with MSLBF in classification.

3.1 Texture recognition

The MSLBF approach was applied to a suite of the Outex [7] database, specifically, Outex_TC_00000. This data set comprises 24 texture classes across 480 128×128 pixel images with 20 images per class. Samples are shown in Figure 1. An $LBP^{riu2}_{8,1} + ^{riu2}_{16,3} + ^{riu2}_{24,5}$ classifier was trained on part of the data (with samples defined by problem no.25 in the Outex_00000 suite) along with three MSLBF classifiers, each with 8 features selected per class-pair by a different selection algorithm. These MSLBF classifiers were provided with more training data by including predicates constructed from circular neighbourhoods at 1, 2.5, 4, 5.5, 7 and 8.5 pixel radii with 8, 16, 24, 32, 40 and 48 samples respectively. The samples were extracted using bilinear sampling. The MSLBF classifiers for this task comprised 276 binary classifiers. The results here correspond to the application of the four classifiers to a separate testing set comprising the images not used in training.

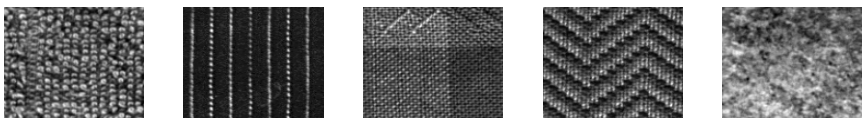

Figure 1: Examples from the Outex_00000 texture suite.

The minimum number of features required (between 1 and 8) for the best score obtained for each class was recorded. Table 1 provides the overall best scores across all classes along with the number of features required to achieve those scores, averaged over all classes. It can be seen that all three MSLBF classifiers outperformed the vanilla combined-predicate LBP. Only the MSLBF+BHIM combination achieved a perfect score and with a lower number of average features required per class. The highest number of features required for BHIM was 4 (class “carpet009”), 5 for CMIM and 6 for AdaBoost. Consequently the MSLBF+BHIM combination constituted a more compact and effective discriminative model than the other combinations. To compare the relative strength of the four classifiers, average histogram distances for each class was computed. Figure 2 plots discriminative strengths of BHIM, CMIM and AdaBoost which shows consistently superior model separation for BHIM generated models.

3.2 Face recognition

A more challenging problem of face recognition given face images captured under large variations in lighting and 3D pose was also considered. The ORL face database [10] was

Classifier	Success (%)	Mean no. features
$LBP^{riu2}_{8,1}+^{riu2}_{16,3}+^{riu2}_{24,5}$	95.4	-
MSLBF+BHIM	100	1.625
MSLBF+CMIM	99.6	1.958
MSLBF+AdaBoost	99.6	2

Table 1: Overall success rate of the four classifiers with Outex_00000 along with the average number of features needed to gain the best scores. The LBP classifier is constructed from smaller sample areas than the MSLBF.

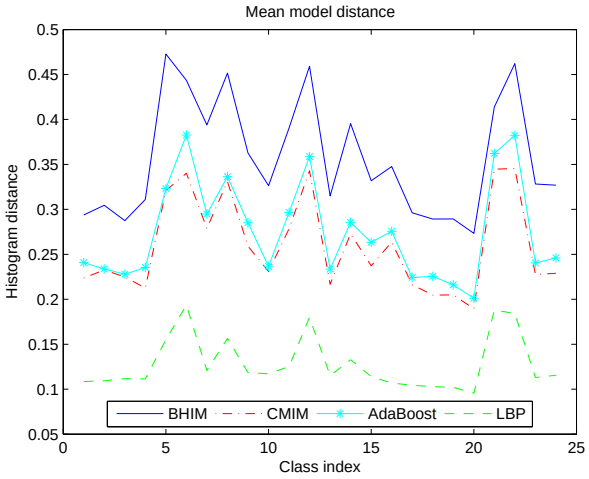

Figure 2: Average histogram separation per class for LBP and MSLBF models generated by BHIM, CMIM and AdaBoost. The mean histogram distance for each class for Outex_00000 is plotted. MSLBF+BHIM has significantly larger between-class distances.

employed for comparing the MSLBF approach to standard multipredicate $LBP^{u2}_{8,1}+^{u2}_{16,3}+^{u2}_{24,5}$. This is a relatively small database comprising 400 unregistered images of 40 people with 10 samples for each person. The samples are greyscale and sized at 92×112 pixels. They contain large within-class variance in lighting, pose and appearance due to the presence/absence of glasses/facial hair and different times of capture (see Figure 3).

Figure 3: Examples from the ORL face database demonstrating within-class variations of appearance, lighting and/or pose.

The data was split up with five samples per person used for training (even indices) and the other five used for testing (odd indices). LBP has been previously applied using a windowed approach [1] to model different facial regions separately with good results. Although modelling different facial regions separately and weighting them according to importance was shown to demonstrate better classification [11], here the faces were mod-

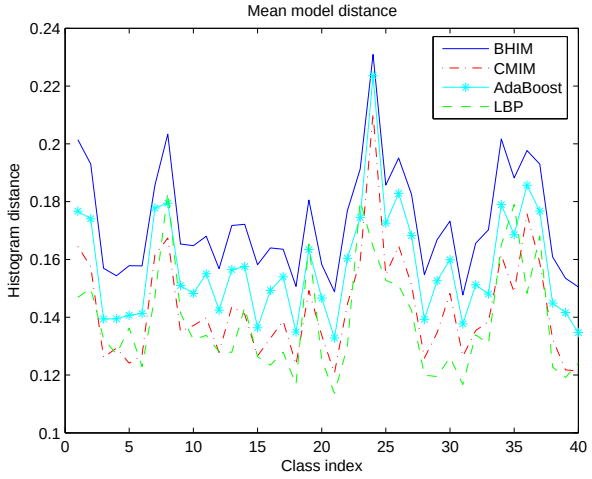

Figure 4: Average histogram separation for LBP and MSLBF models for the ORL face database. BHIM, CMIM and AdaBoost were compared for selecting features for MSLBF models. As with textures, the BHIM features show larger histogram distances.

elled globally to make the problem more generic (independent of ad-hoc region segmentation) and to gauge the benefit of the pairwise-coupled MSLBF approach. The same sample sizes were used for both an LBP and three MSLBF classifiers, again trained with BHIM, CMIM and AdaBoost for up to 8 features, with predicates in the training data being formed at 1, 3 and 5 pixel radii with 8, 16 and 24 samples respectively. Table 2 shows the overall results averaged over classes. The MSLBF combinations outperformed the vanilla LBP classifier with the MSLBF+BHIM combination proving the best. Figure 4 demonstrates the mean histogram distances for each face class for LBP and the three MSLBF classifiers as combined with BHIM, CMIM and AdaBoost. As with textures, MSLBF+BHIM shows better histogram distances.

Classifier	Success (%)	Average bits
$LBP^{u2}_{8,1}+^{u2}_{16,3}+^{u2}_{24,5}$	87	-
MSLBF+BHIM	93	3.9
MSLBF+CMIM	87.5	4.2
MSLBF+AdaBoost	91	3.9

Table 2: Overall success rate of the four classifiers with the ORL database along with the average number of bits needed to gain the best scores. The MSLBF classifiers were constructed from the same sample regions as the LBP classifier.

3.3 Comparison of feature selection methods

Figure 4 showed the MSLBF+AdaBoost combination to be close to MSLBF+BHIM. In order to further examine the effectiveness of feature selection with these combinations, additional experiments (with ground truth) were designed to compare an efficient implementation of BHIM with CMIM and AdaBoost on binary feature selection tasks. CMIM (Conditional Mutual Information Maximisation) [3] is a filter that employs information

theory in a rigorous manner to select features correlated with class labels with minimal redundancy amongst themselves. Ideally, the best set of K features $\{b_1, b_2, ..., b_K\}$ given training data are those that minimise the conditional entropy $H(Y|X_{b_1}, X_{b_2}, ..., X_{b_K})$ where Y is the random variable corresponding to class labels. The experiments here made use of a Matlab implementation of the fast CMIM algorithm. Also implemented for testing was a very efficient binary AdaBoost algorithm based on [4].

For the objective of this experiment, synthetic datasets were created with each binary feature drawn from a flat density. 12 features were randomly selected and a 2^{12}-bin histogram randomly generated for the joint density for each class. The densities for each class were overlapped to random degrees so that a set of joint values had a positive probability for both classes. Samples were drawn from these densities and the corresponding 12-bit binary strings placed into the data at the selected positions as samples. These embedded structures provided a target set of strong features among random ones for algorithms to find. Three factors were examined; (1) the "quality" of features selected, measured by the conditional entropy of the class variable given the selected features, (2) the percentage of features selected that matched the features randomly embedded and (3) computation time for selection. Each of these three factors were plotted against: (a) feature pool size varying between 100 and 800, (b) number of training samples ranging between 10000 and 80000 and (c) number of features an algorithm was required to select from 2 to 12. Each parameter configuration was applied to 50 randomly generated pairs of densities with random structure and the results averaged.

Figure 5 shows the results. Columns correspond to different measures and rows to different parameter changes. From the first row, it can be seen that BHIM outperformed both CMIM and AdaBoost across varying feature pool sizes. The entropies for BHIM correlated closely with the embedded entropies and the features selected were strongly (and often perfectly) correlated with the features that were randomly embedded. CMIM and AdaBoost are comparable but selected significantly lower quality features compared to BHIM. In terms of computation time, all three algorithms are linear in the size of the feature pool. The middle row of Figure 5 plots performance against varying training-set sizes. The figure demonstrates the same trends as with varying feature pools. The bottom row plots performance against increasing numbers of features to select. While entropies and embedded feature selection showed similar trends for all three algorithms as for varying feature pools and training-set sizes, BHIM showed a weakness in its exponential increase in computation time with the number of features to select. This is because the main loop in BHIM involves comparing features against all values shared between two data sets given the previously selected features. The maximum possible number of shared values is equal to 2^k where k is the number of previously selected features. However, the limited size of training sets restricts the number of features that can be considered reliably when calculating expected histogram distances, resulting eventually in a linearisation of computation time. The "window" of the expectation calculation w, involving only at most the w previously selected features, may be estimated as $w = log_2 \frac{T}{v}$, where v is the desired minimum average number of samples per shared value and T is the number of samples in the training set for a class. For a binary histogram, assuming ten samples per bin for a representative sample, v may be set to 20.

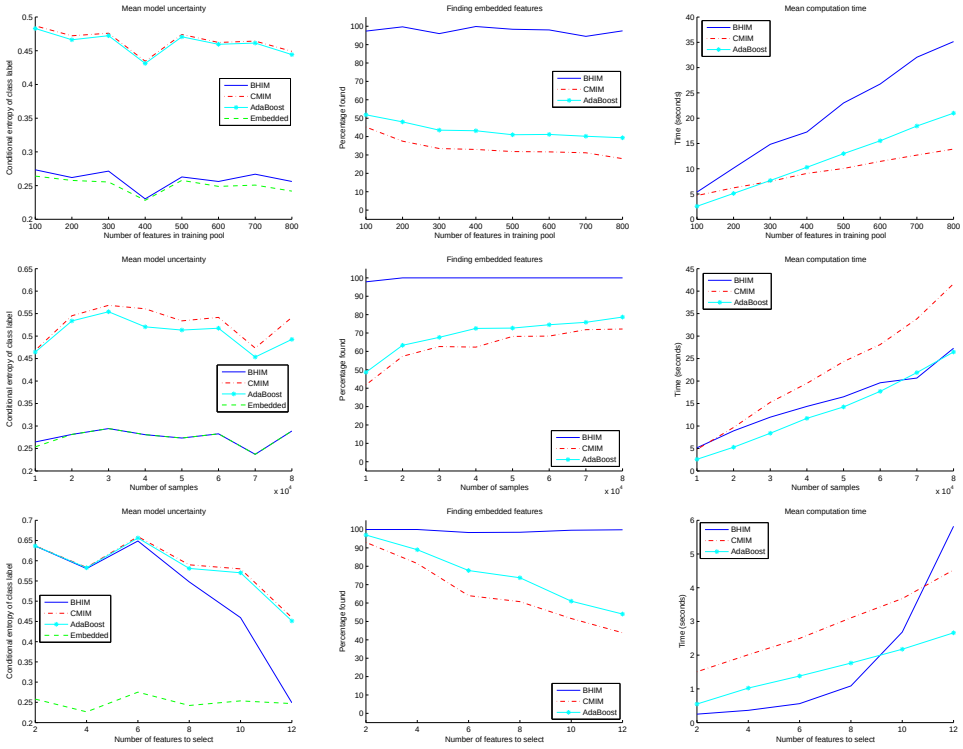

Figure 5: Compare BHIM, CMIM and AdaBoost. The columns (left to right) correspond to average remaining entropy, average correlation of selected features with embedded features and time to compute features. The rows (top to bottom) correspond to varying feature pool size, number of training samples and number of features to select.

4 Conclusions

The contributions of this paper were twofold. Firstly, a new LBP-type model was introduced known as Multiscale Selected Local Binary Features (MSLBF). These are compact predicates that model jointly across scales and are generated through the use of feature selection to select strong single-point features from multiple circular feature pools. A pairwise-coupling classification approach is taken to enable greater specificity in selected features and simplify the feature selection process. Selecting individual pixel features rather than taking combined spatially contiguous groups of features with possible redundancy enables more compact and descriptive models. Importantly, it permits circular feature pools at any scale and angular resolution to be incorporated into the training data. The experiments illustrated that MSLBF combined with a feature selection algorithm enabled models with greater discriminative power to be constructed. Secondly, a novel feature selection algorithm was described known as Binary Histogram Intersection Minimisation. This algorithm selects a typically small number of features with strong discriminative power and minimal redundancy. It is relatively computationally inexpensive and in these experiments consistently demonstrated its ability to select stronger features

than either CMIM or AdaBoost in terms of a concrete information theoretical measure. The algorithm expends variable computational resources depending on the strength of the data available (stronger features require less computation to find) and has limits on the exponential nature of reliable expectation estimates at each step, enabling linearity of computation time in the long term. A major reason for the overall efficiency of the algorithm is that the computationally cheap two-bin histogram intersection is all that is required and entropy calculations are unecessary.

There are two main drawbacks to the pairwise-coupled approach. Firstly, stable results required the same number of features to be used for all classes despite the varying numbers of features required for a given error per class. Secondly, the complete separation between the training of individual binary classifiers does not preclude the possibility of histograms for two classes being similar despite being constructed from completely different features. This can lead to a degradation in performance and effectively dilute the potency of the features originally selected.

References

[1] T. Ahonen, M. Pietikäinen, A. Hadid, and T. Mäenpää. Face recognition based on the appearance of local regions. In *ICPR*, pages 153–156, 2004.

[2] X. Feng, M. Pietikäinen, and A. Hadid. Facial expression recognition with local binary patterns and linear programming. *PRIA*, 15:546–548, 2005.

[3] F. Fleuret. Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research*, 5:1531–1555, 2004.

[4] Y. Freund and R.E. Schapire. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, pages 148–156, 1996.

[5] Trevor Hastie and Robert Tibshirani. Classification by pairwise coupling. In M. Jordan, M. Kearns, and S. Solla, editors, *NIPS*, volume 10. The MIT Press, 1996.

[6] T. Mäenpää, M. Pietikäinen, and T. Ojala. Texture classification by multipredicate local binary pattern operators. In *ICPR*, pages 3951–3954, Barcelona, 2000.

[7] T. Ojala, T. Mäenpää, M. Pietikäinen, J. Viertola, J. Kyllönen, and S. Huovinen. Outex - new framework for empirical evaluation of texture analysis algorithms. In *ICPR*, volume 1, pages 701–706, Quebec, Canada, 2002.

[8] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray scale and rotation invariant texture analysis with local binary patterns. *PAMI*, 24:971–987, 2002.

[9] X. Qing, Y. Jie, and D. Siyi. Texture segmentation using lbp embedded region competition. *Electronic Letters on CVIA*, 5:41–47, 2005.

[10] F. Samaria and A. Harter. Parameterisation of a stochastic model for human face identification. In *IEEE Workshop on Applications of Computer Vision*, 1994.

[11] C. Shan, S. Gong, and P. McOwan. Conditional mutual information based boosting for facial expression recognition. In *BMVC*, Oxford, UK, 2005.