Vietnam J Comput Sci (2015) 2:3–12
DOI 10.1007/s40595-014-0024-7

REGULAR PAPER

Classifying many-class high-dimensional fingerprint datasets using random forest of oblique decision trees

Thanh-Nghi Do · Philippe Lenca · Stéphane Lallich

Received: 16 January 2014 / Accepted: 17 May 2014 / Published online: 10 June 2014

Abstract Classifying fingerprint images may require an important features extraction step. The scale-invariant feature transform which extracts local descriptors from images is robust to image scale, rotation and also to changes in illumination, noise, etc. It allows to represent an image in term of the comfortable bag-of-visual-words. This representation leads to a very large number of dimensions. In this case, random forest of oblique decision trees is very efficient for a small number of classes. However, in fingerprint classification, there are as many classes as individuals. A multi-class version of random forest of oblique decision trees is thus proposed. The numerical tests on seven real datasets (up to 5,000 dimensions and 389 classes) show that our proposal has very high accuracy and outperforms state-of-the-art algorithms.

Keywords Fingerprint classification · Scale-invariant feature transform · Bag-of-visual-words · Random forest of oblique decision trees

T.-N. Do · P. Lenca
Institut Mines-Telecom; Telecom Bretagne, CNRS UMR 6285
Lab-STICC, Université européenne de Bretagne, Brest, France

P. Lenca
e-mail: philippe.lenca@telecom-bretagne.eu

T.-N. Do (✉)
Can Tho University, Can Tho , Vietnam
e-mail: dtnghi@cit.ctu.edu.vn

S. Lallich
Laboratoire ERIC, Université de Lyon 2, Lyon, France
e-mail: stephane.lallich@univ-lyon2.fr

1 Introduction

Due to their uniqueness and consistency over time [1], fingerprint identification is one of the most well-known technique for person identification. It is successfully used in both government and civilian applications such as suspect and victim identifications, border control, employment background checks, and secure facility access [2]. Fingerprint recognition systems commonly use minutiae (i.e. ridge ending, ridge bifurcation, etc.) as features since a long time. Recently a method based on feature-level fusion of fingerprint and finger-vein has been proposed [3]. Recent advances in technology make each day easier the acquisition of fingerprints features. In addition, there is a growing need for reliable person identification and thus fingerprint technology is more and more popular.

Fingerprint systems mainly focus on two applications: fingerprint matching which computes a match score between two fingerprints, and fingerprint classification which assigns fingerprints into one of the (pre)defined classes. The state-of-the-art methods are based on minutiae points and ridge patterns, including a crossover, core, bifurcation, ridge ending, island, delta and pore [2,4,5]. Useful features and classification algorithms are found in [6,7]. Most of these techniques have no difficulty in matching or classifying good quality fingerprint images. However, dealing with low-quality or partial fingerprint images still remains a challenging pattern recognition problem. Indeed a biometric fingerprint acquisition process is inherently affected by many factors [8]. Fingerprint images are concerned by displacement (the same finger may be captured at different locations or rotated at different angles on the fingerprint reader), partial overlap (part of the fingerprint area fall outside the fingerprint reader), distortion, noise, etc.

An efficient feature extraction technique, called the scale-invariant feature transform (SIFT) was proposed by [9] for detecting and describing local features in images. The local features obtained by the SIFT method are robust to image scale, rotation, changes in illumination, noise and occlusion. Therefore, the SIFT is used for image classification and retrieval. The bag-of-visual-words (BoVW) model based on SIFT extraction is proposed in [10]. Some recent fingerprint techniques [11–13] showed that the SIFT local feature can improve matching tasks.

Unfortunately, when using SIFT and the BoVW models, the number of features could be very large (e.g. thousands dimensions or visual words). We here propose to classify fingerprint images with random forest of oblique decision trees [15,16] which have shown to have very high accuracy when dealing with very-high-dimensional datasets for few class problems. However, for individual identification each person is considered as a single class. We thus extend this approach to deal with a very large number of classes. Experiments with real datasets and comparison with state-of-the-art algorithms show the efficiency of our proposal.

The paper is organized as follows. Section 2 presents the image representation using the SIFT and the BoVW model. Section 3 briefly introduces random forests of oblique decision trees and then extend this algorithm to multi-classes classification of very-high-dimensional datasets. The experimental results are presented in Sect. 4. We then conclude in Sect. 5.

2 SIFT and bag-of-visual-words model

When dealing with images like fingerprint one has to extract first local descriptors. The SIFT method [9] detects and describes local features in images. SIFT is based on the appearance of the object at particular interest points. It is invariant to image scale, rotation and also robust to changes in illumination, noise, occlusion. It is thus adapted for fingerprint images as pointed out by [11–13].

Step 1 (Fig. 2) detects the interest points in the image. These points are either maximums of Laplace of Gaussian, 3D local extremes of Difference of Gaussian [17], or points detected by a Hessian-affine detector [18]. Figure 1 shows some interest points detected by a Hessian-affine detector for fingerprint images. The local descriptors of interest point are computed on a grey level gradient of the region around the interest point (step 2 in Fig. 2). Each SIFT descriptor is a 128-dimensional vector.

A main stage consists of forming visual words from the local descriptors. Most of approaches perform a *k*-means [19] on descriptors. Each cluster is considered as a visual word represented by the cluster centre [10] (step 3 in Fig. 2). The set of clusters constitutes a visual vocabulary (step 4 in Fig. 2). Each descriptor is then assigned to the nearest cluster (step 5 in Fig. 2). The frequency of a visual word is the number of descriptors attached to the corresponding cluster (step 6 in Fig. 2). An image is then represented by the frequencies of the visual words, i.e. a BoVW.

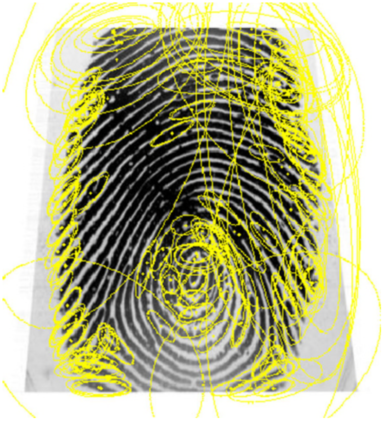


Fig. 1 Interest points detected by Hessian-affine detector on a fingerprint

SIFT method has demonstrated very good qualities to represent images. However, it leads to a very large number of dimensions. Indeed, a large number of descriptors may require a very large number of visual words to be efficient. In addition when dealing with fingerprint classification, the number of classes corresponds to the number of individuals in the dataset. In the next section, we investigate machine learning algorithms for this kind of data.

3 Multi-classes random forest of oblique decision trees

Random forests are one of the most accurate learning algorithms, but their outputs are difficult for humans to interpret [20]. We are here only interested in classification performance.

Reference [21] pointed out that the difficulty of high-dimensional classification is intrinsically caused by the existence of many redundant or noisy features. The comparative studies in [20,22–25] showed that support vector machines [26], boosting [27] and random forests [28] are appropriate for very high dimensions. However, there are few studies [29,30] for extremely large number of classes (typically hundreds of classes).

3.1 From random forests to random forests of oblique decision trees

A random forest is an ensemble classifier that consists of (potentially) large collection of decision trees. The algorithm for inducing a random forest was proposed in [28].

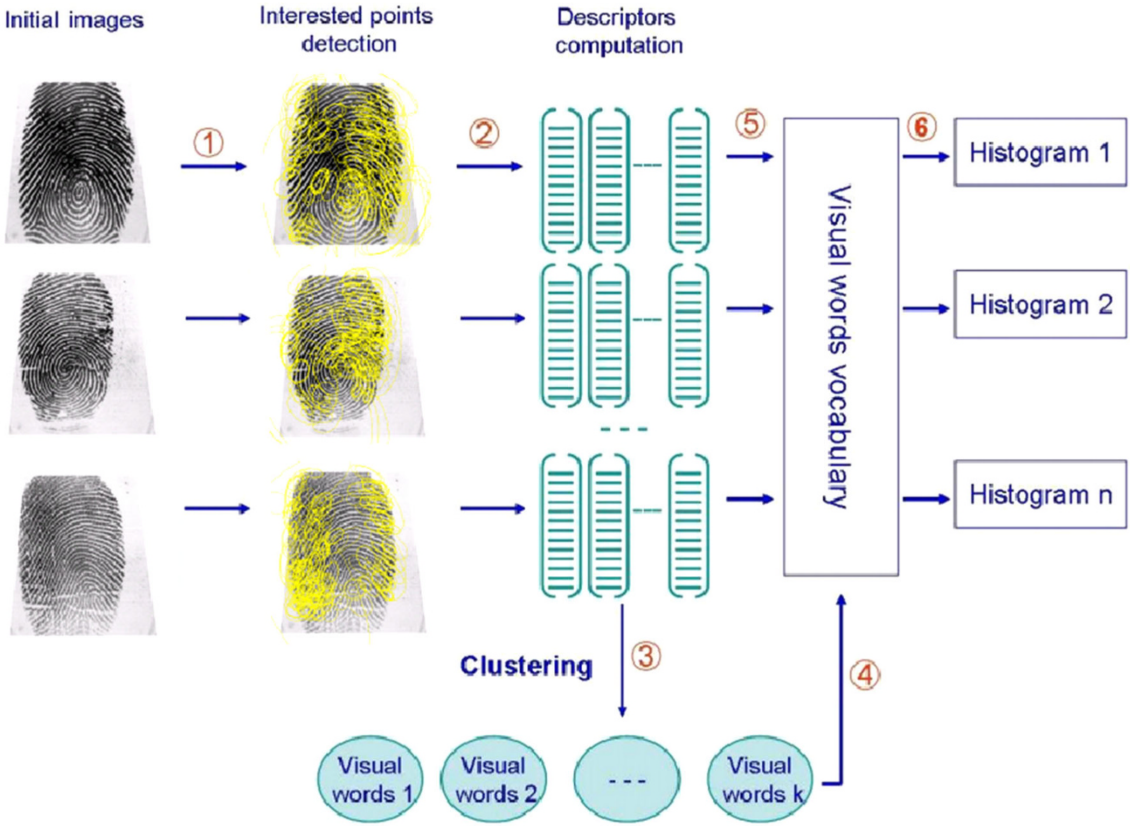


Fig. 2 Bag-of-visual-words model for representing images

The algorithm combines bagging idea [31] and the selection of a random subset of attributes introduced in [32,33] and [34].

Let us consider a training set D of n examples x_i described by p attributes. The bagging approach generates t new training sets B_j, $j = 1 \dots t$, known as bootstrap sample, each of size n, by sampling n examples from D uniformly and with replacement. The t decision trees in the forest are then fitted using the t bootstrap samples and combined by voting for a classification task or averaging the output for regression task. Each decision tree DT_j $(j = 1, \dots, t)$ in the forest is thus constructed using the bootstrap sample B_j as follows: for each node of the tree, randomly choose p' attributes $(p' << p)$ and calculate the best split based on one of these p' attributes; the tree is fully grown and not pruned.

A random forest is thus composed of trees having sufficient diversity (thanks to bagging and random subset of attributes) each of them having low bias (thanks to unpruning). Random forests are known to produce highly accurate classifier and are thus very popular [20].

However, for each tree only a single attribute is used to split each node. Such univariate strategy does not take into account dependencies between attributes. The strength of individual trees could thus be reduced typically when dealing with very-high-dimensional datasets which are likely to contain dependencies among attributes.

One can thus use oblique decision trees (e.g. OC1 [35]) or hybridization in a post-growing phase that uses other classifiers in tree's node (e.g. genetic algorithm [36], neural network [37,38], linear discriminant analysis [39, 40], support vector machines [41]). Recently, ensemble of oblique decision trees has attracted much research interests. For example proximal linear support vector machines [42] (PSVM) are used in [14–16] and ridge regression is proposed in [43] for random forests. The embedded support vector machines (SVM) in a forest of trees have shown very high performance especially for very-high-dimensional datasets [16] and a reasonable number of classes [15].

We thus here extend these approaches to deal with very large number of classes. Indeed, the fingerprint application needs to manage very-high-dimensional points with hundreds of individuals to classify. Furthermore, we provide also the performance analysis of multi-class random oblique decision trees in terms of the error bound and the algorithmic complexity. This theoretical analysis illustrates how our proposed algorithm is efficient in the fingerprint classification with many classes.

3.2 Multi-class random forests of oblique decision trees

We propose to induce a forest of binary oblique decision trees. Our approach will thus build a set of trees that will separate the c classes at each non-terminal node into two subsets of classes of size c_1 and c_2 ($c_1 + c_2 = c$). In such a way, the algorithm will reach terminal nodes (leaves). As proposed in the Random Forest of Oblique Decision Trees algorithm (RF-ODT) [16] these binary splits are done by proximal SVM [42].

The state-of-the-art multi-class SVMs are categorized into two types of approaches. The first one solves an optimization problem for multi-class separation [44,45]. This approach can thus require expensive calculations and parameter tuning.

The second one uses a series of binary SVMs to decompose the multi-class problem (e.g. "One-Versus-All" (OVA [26]), "One-Versus-One" (OVO [46]) and Decision-Directed Acyclic Graph (DDAG [47]). Decision-Directed Acyclic Graph is rather complex and OVO needs to train $c(c-1)/2$ classifiers, while OVA needs only to build c classifiers.

Hierarchical methods divide the data into two subsets until every subset consists of only one class. The Divide-by-2 (DB2) [48] proposes three strategies (class centroid-based division using k-means [19], class mean distances and balanced subsets) to construct the two subsets of classes. The Dendrogram-based SVM [49] uses ascendant hierarchical clustering method. The Dendrogram clustering algorithms have a complexity that is at least cubic in the number of datapoints compared to the linear complexity of k-means.

Furthermore, the oblique tree construction aims at partitioning the data of the non–terminal node into two subsets. In practice, k-means is the most widely used partitional clustering algorithm because it is simple, easily understandable, and reasonably scalable. The partitional clustering algorithm k-means (setting $k = 2$ due to the data partition at the non-terminal node into two subsets) is the appropriate method.

Our proposal consists of an efficient hybrid method using the previous methods. Multi-Class Oblique Decision Trees (MC-ODT) are build using OVA (for small number of classes, i.e. $c \leq 3$) and a DB2-like method (for, i.e. $c > 3$) to perform the binary multivariate splits with a proximal SVM (denoted OVA-DB2-like approach in the later). These MC-ODT are then used to form a Random Forest of MC-ODT (MCRF-ODT) as illustrated in Fig. 5. Theoretical considerations supporting our approach are presented in Sect. 3.3.

Figure 3 illustrates the OVA-DB2-like approach for $c \leq 3$. On the left-hand side, $c = 3$, the algorithm creates two super classes (a positive part and a negative part). One super class groups together 2 classes and the other one matches the third class. This corresponds to the classical OVA strategy. Therefore, the algorithm only uses the OVA and the biggest margin criteria for performing an oblique binary split PSVM while dealing with $c \leq 3$ (i.e. the plane P_1) as illustrated on the right-hand-side of Fig. 3.

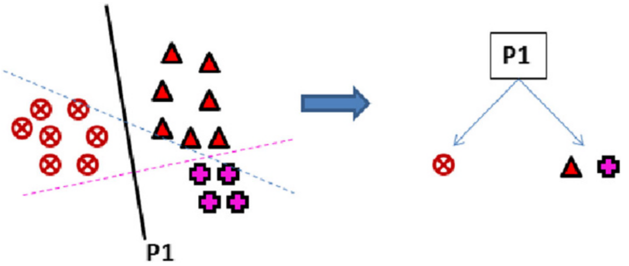


Fig. 3 Oblique splitting for c classes ($c \leq 3$)

When the number of classes c is greater than 3, a k-means [19] is used on all the datapoints. This improves the quality of the two super classes in comparison with Divide-by-2 which only uses the class centroid (obviously Divide-by-2 is faster, but the quality of the classes is lower).

The most impure cluster is considered as the positive super class, the second cluster as the negative super one. The classes of this cluster (positive part) are then sorted in descending order of class size so that around 15 % of the datapoints of the minority classes are moved to the other cluster (negative part). This is done to reduce the noise in the positive cluster and also to balance the two clusters[1] (in terms of size and number of classes). Finally, the proximal SVM performs the oblique split to separate the two super classes.

These processes (OVA, k-means clustering and PSVM) are repeated to split the datapoints into terminal nodes (w.r.t. two criteria: the first one concerns the minimum size of nodes and the second one concerns the error rate in the node) as illustrated in Fig. 4. The majority class rule is applied to each terminal node.

The pseudocode of the random oblique decision tree algorithm for multi-class (MC-ODT) is presented in Algorithm 1 and the MCRF-ODT algorithm is illustrated in Fig. 5.

3.3 Performance analysis

For a non-terminal node D with c classes, the OVO and DDAG strategies require $\frac{c(c-1)}{2}$ tests (each test corresponds to a binary SVM) to perform a binary oblique split. These strategies thus become intractable when c is large.

The OVA-DB2-like approach used in MC-ODT is designated for separating the data into two balanced super classes. The OVA-DB2-like approach can be considered as the Twoing rule [50].

[1] Our empirical tests, from 2 % up to 30 %, showed that 15 % gives good super classes.

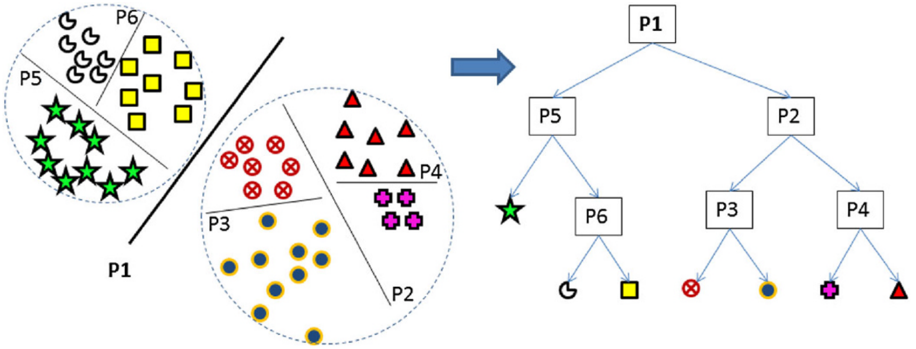
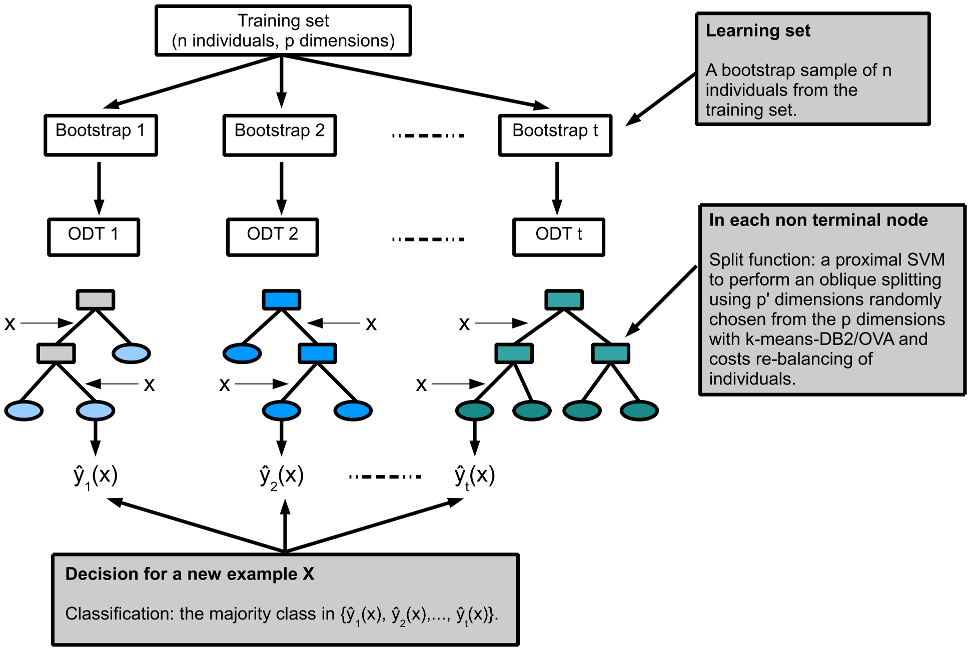


Fig. 4 Oblique splitting for c classes ($c > 3$)



Learning set

A bootstrap sample of n individuals from the training set.

In each non terminal node

Split function: a proximal SVM to perform an oblique splitting using p' dimensions randomly chosen from the p dimensions with k-means-DB2/OVA and costs re-balancing of individuals.

Decision for a new example X

Classification: the majority class in {ŷ1(x), ŷ2(x),..., ŷt(x)}.

Fig. 5 Multi-class random forest of oblique decision trees

Therefore, the OVA-DB2-like approach tends to produce MC-ODTs with less non-terminal nodes than the OVA method. Thus, it requires less tests and lower computational time.

Error bound

Furthermore, according to [47], if one can classify a random n sample of labelled examples using a perceptron (e.g. a linear SVM) DDAG G (i.e. a DDAG with a perceptron at every node) on c classes containing K decision nodes (e.g. non-terminal nodes) with margins γ_i at node i, then the generalization error bound $\epsilon_j(G)$, with probability greater than $1-\delta$, is given by:

$$\epsilon_j(G) \leq \frac{130R^2}{n}\left(M' \log(4en)\log(4n) + \log\left(\frac{2(2n)^K}{\delta}\right)\right) \quad (1)$$

where $M' = \sum_{i=1}^{K} \frac{1}{\gamma_i^2}$ and R is the radius of a hypersphere enclosing all the datapoints.

The error bound thus depends on M' (the margin γ_i's) and K decision nodes (non-terminal nodes). Let, now examine why our proposal has two interesting properties in comparison with the OVA approach:

- as mentioned above a MC-ODT based on the OVA-DB2-like approach has smaller K than the ones using the OVA method.
- the separating boundary (margin size) at a non-terminal node obtained by the OVA-DB2-like approach is larger than the one by the OVA method. As a consequence M' is smaller.

Therefore, the error bound of MC-ODT based on the OVA-DB2-like approach is smaller than the one made by the OVA strategy.

Algorithm 1: Random oblique decision tree for multi-class

```
input :
        $D$ bootstrap sample of $n$ datapoints in $p$ dimensions with $c$ classes
        $p'$ number of random dimensions used to perform oblique splits ($p' < p$)
        $\epsilon$ error_tolerant acceptance at terminal nodes
        $min\_obj$ minimum size of terminal nodes
output:
        ODT model
Learn:
PROC RECURSIVE-ODT($D$, $p'$, $\epsilon$, $min\_obj$)
begin
    1- counting the number of datapoints for each class
    2- computing the error rate of $D$ upon the majority class as the label of $D$
    3- if ($n < min\_obj$) or ($error\_rate < \epsilon$) then
      {
            - assigning the majority class as the label of $D$
            - return terminal node $D$
      }
      else
      {
            - randomly sampling without replacement $p'$ dimensions
            - if ($c \leq 3$) then
              {
                  - PSVM uses One-Versus-All method with the biggest margin strategy to perform a binary oblique split $D$ in $p'$ dimensions into $D_{pos}$ (positive part) and $D_{neg}$ (negative part)
              }
              else
              {
                  - using k-means algorithm to group datapoints in $D$ into two clusters, denoted by $cluster\_1$ and $cluster\_2$
                  - the most impure cluster is considered as the candidate of the positive class $candidate\_class$
                  - around 15% of the datapoints of the minority classes in $candidate\_class$ are moved to the other cluster
                  - PSVM performs a binary oblique split $D$ in $p'$ dimensions (with $candidate\_class$ being the positive class and the rest being the negative class) into $D_{pos}$ (positive part) and $D_{neg}$ (negative part)
              }
            - RECURSIVE-ODT($D_{pos}$, $p'$, $\epsilon$, $min\_obj$)
            - RECURSIVE-ODT($D_{neg}$, $p'$, $\epsilon$, $min\_obj$)
      }
end
```

In comparison with the OVO and DDAG approaches, our proposal can reduce the error bound in terms of K. But the margin size at each decision node (two classes separation) obtained by OVO and DDAG is larger than the one obtained by the OVA-DB2-like (two super classes separation). Therefore, it is not easy to compare the error bound in terms M' in this context. However, an optimal split of two classes in D obtained by a binary SVM under OVO or DDAG constraints can not assure efficient separation of the c classes into two super classes.

Computational costs

According to [47], a binary SVM classification with n training datapoints has an empirical complexity as follows:

$$\Omega(n, 2) \approx \alpha n^{\beta} \tag{2}$$

where $\beta \approx 2$ for binary SVM algorithms using the decomposition method and some positive constant α.

Let us consider a multi-class classification problem at a non-terminal node D with n training datapoints and c balanced classes (i.e. the number of training datapoints of each class is about n/c). The standard OVA approach needs c tests (binary SVM learning tasks on n training datapoints) to perform a binary oblique split. The algorithmic complexity is:

$$\Omega_{\text{OVA}}(n, c) \approx c\alpha n^{\beta} \tag{3}$$

The OVO or DDAG approaches need $c(c-1)/2$ tests (binary SVM learning tasks on $2n/c$ training datapoints) to perform a binary oblique split at a non-terminal node D. The algorithmic complexity is:

$$\begin{aligned}\Omega_{\text{OVO,DAG}}(n, c) &\approx \frac{c(c-1)}{2}\alpha\left(\frac{2n}{c}\right)^{\beta} \\ &\approx 2^{(\beta-1)}c^{(2-\beta)}\alpha n^{\beta}\end{aligned} \tag{4}$$

The OVA-DB2-like approach requires only one test (binary SVM learning tasks on n training datapoints) to perform a binary oblique split in a non-terminal node D to separate the two super classes (positive and negative parts). The algorithmic complexity is the same as for the binary case (formula 2) which is the smallest complexity. It must be noted that the complexity of the OVA-DB2-like approach in formula (2) does not include the k-means clustering used to create two super classes. But this step requires insignificant time compared with the quadratic programming time.

Let now examine the complexity of building an oblique multi-class classification tree with the OVA-DB2-like approach that tends to maintain, at each node, balanced classes. This strategy can thus build a balanced oblique decision tree (i.e. the tree height is $\lceil \log_2 c \rceil$) and any ith tree level has 2^i nodes having $n/2^i$ training datapoints. Therefore, the complexity of the multi-class oblique tree algorithm based on OVA-DB2-like approaches is:

$$\begin{aligned}\Omega_{\text{OVA}-\text{DB2}-\text{like}}(n, c) &\approx \sum_{i=0}^{\lceil \log_2 c \rceil} \alpha 2^i \left(\frac{n}{2^i}\right)^{\beta} \\ &= \sum_{i=0}^{\lceil \log_2 c \rceil} \alpha n^{\beta} \left(2^{(1-\beta)}\right)^i\end{aligned} \tag{5}$$

Due to $2^{(1-\beta)} < 1$ ($\beta \approx 2$), we have:

$$\sum_{i=0}^{\lceil \log_2 c \rceil} \left(2^{(1-\beta)}\right)^i \approx \frac{1}{1-2^{(1-\beta)}} \tag{6}$$

Thus, applying formula (6) to the right side of (5) yields the new algorithmic complexity of the multi-class oblique tree based on the OVA-DB2-like approach as follows.

$$\Omega_{\text{OVA}-\text{DB2}-\text{like}}(n, c) \approx \frac{\alpha n^{\beta}}{1-2^{(1-\beta)}} = \frac{\alpha n^{\beta} 2^{(\beta-1)}}{2^{(\beta-1)}-1} \tag{7}$$

Formula (7) shows that the training task of a MC-ODT scales $O(n^2)$. Therefore, the complexity of a MCRF-ODT forest is $O(t.n^2)$ for training t models of MC-ODT.

4 Numerical test results

Experiments are conducted with seven real fingerprint datasets (respectively, FPI-57, FPI-78, …, and FPI-389, with 57, 78, …, and 389 colleagues; between 15 and 20 fingerprints were captured for each individual). Fingerprints acquisition was done with Microsoft Fingerprint Reader (optical fingerprint scanner, resolution: 512 DPI, image size: 355 × 390, colours: 256 levels greyscale). Local descriptors were extracted with the Hessian-affine SIFT detector proposed in [18]. These descriptors were then grouped into 5,000 clusters with k-means algorithm [19] (the number of clusters/visual words was optimized between 500 and over 5,000, 5,000 clusters was the optimum). The BoVW model was thus calculated from these 5,000 visual words. Last, the datasets were splitted into training set and testing set. The datasets are described in Table 1.

The training set was used to tune the parameters of the competitive algorithms including MCRF-ODT (MCRF-ODT is implemented in C++, using the Automatically Tuned Linear Algebra Software [51]), SVM [26] (using the highly efficient standard SVM algorithm LibSVM [52] with OVO for multi-class), kNN [53], C4.5 [54], AdaBoost [27] of C4.5, RF-CART [28]. The Weka library [55] was used for the four last algorithms.

Table 1 Description of seven fingerprint image datasets

Datasets	# Classes	# Dimensions	# Images (train–test)
FPI-57	57	5,000	700–352
FPI-78	78	5,000	950–422
FPI-120	120	5,000	1,438–480
FPI-153	153	5,000	1,700–672
FPI-185	185	5,000	2,000–765
FPI-235	235	5,000	2,485–1,000
FPI-389	389	5,000	4,306–2,000

We tried to use different kernel functions of the SVM algorithm, including a polynomial function of degree d, a RBF (RBF kernel of two datapoints x_i, x_j, $K[i, j] = exp(-\gamma \|x_i - x_j\|^2)$). The optimal parameters for accuracy are the following: RBF kernel (with $\gamma = 0.0001$, $c = 10{,}000$) for SVM, one neighbour for kNN, at least two example in a leave for C4.5, 200 trees and 1,000 random dimensions for MCRF-ODT, RF-CART, 200 trees for AdaBoost-C4.5. We remark that MCRF-ODT and RF-CART used the out-of-bag samples (the out of the bootstrap samples) during the forest construction for finding the parameters (with $p' = 1,000$, $\epsilon = 0$, min_obj $= 2$ and $t = 200$), corresponding to the best experimental results.

Given the differences in implementation, including the programming language used (C++ versus Java), a comparison of computational time is not really fair. Table 2, Fig. 6 report average computational times for the faster algorithms to illustrate that MCRF-ODT is very competitive. Obviously, the univariate algorithm RF-CART is faster.

The accuracies of the seven algorithms on the seven datasets are given in Table 3 and Fig. 7.

The experimental results showed that our proposal using SIFT/BoVW and MCRF-ODT has achieved more than 93 % accuracy for fingerprint images classification.

As it was expected, firstly 1-NN, C4.5 and NB methods which are based on an unique classifier are overmatched by LibSVM and ensemble methods, secondly the performance of these methods dramatically decreases with the number of classes. 1-NN, C4.5 and NB are always bottom of the ranking for each of the seven datasets (7th, 6th and 5th position) and they lose a lot of accuracy when the number of classes increases (from 57 to 389), especially 1-NN and C4.5 which, respectively, decrease from 59.9 to 28.75 % and from 75.0 to 45.8 %, while NB decreases only from 85.2 to 74.6 %.

RF-CART and Adaboost-C4.5, which are among the most common ensemble-based methods, occupy an intermediate position, with a slight superiority of RF-CART on Adaboost-C4.5 (mean rank score of, respectively, 3.1 and 3.9). The accuracies of these methods are already somewhat less affected by the increase in the number of classes, decreasing from 93.5 to 86.3 % for RF-CART and from 91.5 to 82 % for Adaboost C4.5.

The best results are always obtained by LibSVM and above all by our multi-class MCRF-ODT, the new proposed method. LibSVM holds the rank 2 on each experimented dataset, with a mean accuracy of 93.4 %, while MCRF-ODT gets the best result on each of the seven datasets with an average accuracy of 95.89 %, which corresponds to an improvement of 2.49 percentage points compared with LibSVM. This superiority of MCRF-ODT on LibSVM is statistically significant, in so far as according to the sign test, the p value of the observed results (7 wins of MCRF-ODT on LibSVM with 7 datasets) is equal to 0.0156. In addition, these two meth-

Table 2 Average time calculation (s/tree, PC-3.4 GHz)

Algorithm	FPI-57	FPI-78	FPI-120	FPI-153	FPI-185	FPI-235	FPI-389
MCRF-ODT	2.5	3.5	6.0	8.5	12.0	15.0	26
AdaBoost-C4.5	4.0	6.0	10.5	14.0	18.5	26.5	90.0
RF-CART	0.7	1.0	2.0	3.0	4.3	6.3	17.0
LibSVM	3.5	6.2	14.3	21.7	30.0	46.0	120.0

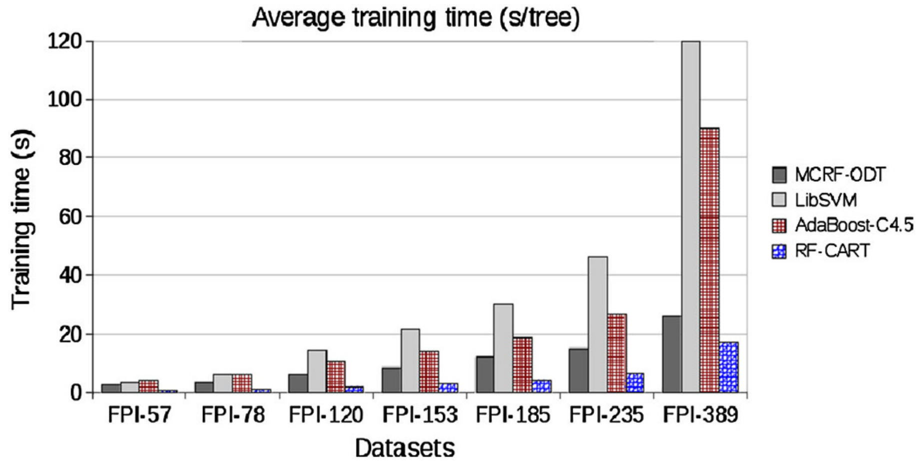


Fig. 6 Training time (s/tree)

Table 3 Classification results in terms of accuracy (%)

Algorithms	FPI-57	FPI-78	FPI-120	FPI-153	FPI-185	FPI-235	FPI-389
MCRF-ODT	**97.44**	**97.60**	**96.04**	**95.24**	**95.29**	**94.60**	**95.00**
LibSVM	*95.46*	*94.79*	*92.50*	*92.86*	*93.46*	*92.10*	*92.65*
AdaBoost-C4.5	91.48	89.34	89.17	84.52	85.10	84.10	81.95
RF-CART	93.47	92.42	88.33	91.67	89.02	87.50	86.30
C4.5	75.00	69.19	61.67	59.23	55.16	53.60	45.80
NB	85.23	81.52	78.33	75.15	78.30	74.90	74.60
1NN	59.94	51.19	46.88	41.52	36.99	33.20	28.75

Bold values indicate the best algorithm and italic values indicate the second best

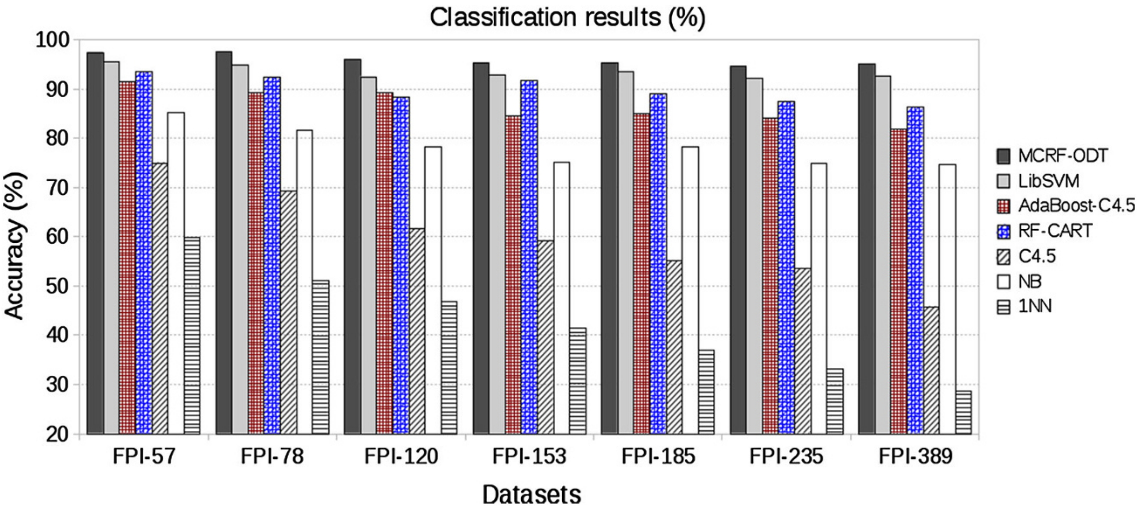


Fig. 7 Classification results in terms of accuracy (%)

ods lose only little efficiency when the number of classes increases, since the corresponding accuracies decrease from 97.60 to 94.60 % for MCRF-ODT and from 95.5 to 92.1 % for LibSVM.

5 Conclusion and future works

We presented a novel approach that achieves high performances for classification tasks of fingerprint images. It associates the BoVW model (induced from the SIFT method which detects and describes local features in images) and an extension of random forest of decision trees to deal with hundreds of classes and thousands of dimensions. The experimental results showed that the Multi-class RF-ODT algorithm is very efficient in comparison with C4.5, random forest RF-CART, AdaBoost of C4.5, support vector machine and *k* nearest neighbours.

A forthcoming improvement will be to extend this algorithm to deal with extremely large number of classes (e.g. up to thousands of classes). A parallel implementation can greatly speed up learning and classifying tasks of the multi-class RF-ODT algorithm.

References

1. Galton, F.: Finger Prints. Macmillan and Co, London (1892)
2. Maltoni, D., Maio, D., Jain, A., Prabhakar, S.: Handbook of Fingerprint Recognition. Springer, New York (2009)
3. Yang, J., Zhang, X.: Feature-level fusion of fingerprint and finger-vein for personal identification. Pattern Recognit. Lett. **33**(5), 623–628 (2012)
4. Jain, A., Feng, J., Nandakumar, K.: Fingerprint matching. IEEE Comput. **43**(2), 36–44 (2010)
5. Yager, N., Amin, A.: Fingerprint verification based on minutiae features: a review. Pattern Anal. Appl. **7**, 94–113 (2004)
6. Yager, N., Amin, A.: Fingerprint classification: a review. Pattern Anal. Appl. **7**, 77–93 (2004)
7. Cappelli, R., Maio, D., Maltoni, D.: A multi-classifier approach to fingerprint classification. Pattern Anal. Appl. **5**, 136–144 (2002)
8. Poh, N., Kittler, J.: A unified framework for biometric expert fusion incorporating quality measures. IEEE Trans. Pattern Anal. Mach. Intell. **34**(1), 3–18 (2012)
9. Lowe, D.: Object recognition from local scale invariant features. In: Proceedings of the 7th International Conference on Computer Vision, pp 1150–1157 (1999)
10. Bosch, A., Zisserman, A., Muñoz, X.: Scene classification via pLSA. In: Proceedings of the European Conference on Computer Vision, pp. 517–530 (2006)
11. Park, U., Pankanti, S., Jain, A.: Fingerprint verification using SIFT features. In: SPIE Defense and Security Symposium (2008)
12. Malathi, S., Meena, C.: Partial fingerprint matching based on SIFT features. Int. J. Comput. Sci. Eng. **2**(4), 1411–1414 (2010)
13. Zhou, R., Sin, S., Li, D., Isshiki, T., Kunieda, H.: Adaptive SIFT-based algorithm for specific fingerprint verification. In: 2011 International Conference on Hand-Based Biometrics (ICHB), pp. 1–6 (2011)
14. Do, T.N., Lallich, S., Pham, N.K., Lenca, P.: Un nouvel algorithme de forêts aléatoires d'arbres obliques particulièrement adapté à la classification de données en grandes dimensions. In: Ganascia, J.G., Gançarski, P. (eds.) Extraction et Gestion des Connaissances 2009, pp. 79–90. Strasbourg, France (2009)
15. Simon, C., Meessen, J., De Vleeschouwer, C.: Embedding proximal support vectors into randomized trees. In: European Symposium on Artificial Neural Networks. Advances in Computational Intelligence and Learning, pp. 373–378 (2009)
16. Do, T.N., Lenca, P., Lallich, S., Pham, N.K.: Classifying very-high-dimensional data with random forests of oblique decision trees. In: Advances in Knowledge Discovery and Management. Studies in Computational Intelligence, vol. 292, pp. 39–55. Springer-Verlag, Berlin (2010)
17. Lowe, D.: Distinctive image features from scale invariant keypoints. Int. J. Comput. Vis., 91–110 (2004)
18. Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. Int. J. Comput. Vis. **60**(1), 63–86 (2004)
19. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press, vol. 1, pp. 281–297 (January 1967)
20. Caruana, R., Karampatziakis, N., Yessenalina, A.: An empirical evaluation of supervised learning in high dimensions. In: Proceedings of the 25th International Conference on Machine Learning, pp. 96–103 (2008)
21. Donoho, D.: A high-dimensional data analysis: the curses and blessings of dimensionality (2000). http://www-stat.stanford.edu/donoho/Lectures/AMS2000/Curses. Accessed 15 Sept 2012
22. Statnikov, A., Aliferis, C., Tsamardinos, I., Hardin, D., Levy, S.: A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. Bioinformatics **21**, 631–643 (2005)
23. Statnikov, A., Wang, L., Aliferis, C.: A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. BMC Bioinform. **9:319**(1), 10 (2008)
24. Yang, P., Hwa, Y., Zhou, B., Zomaya, A.: A review of ensemble methods in bioinformatics. Curr. Bioinform. **5**(4), 296–308 (2010)
25. Ogutu, J., Piepho, H., Schulz-Streeck, T.: A comparison of random forests, boosting and support vector machines for genomic selection. BMC Proc. **5**, 1–5 (2011)
26. Vapnik, V.: The Nature of Statistical Learning Theory. Springer-Verlag, New York (1995)
27. Freund, Y., Schapire, R.: A decision-theoretic generalization of on-line learning and an application to boosting. In: Computational Learning Theory. Proceedings of the Second European Conference, pp. 23–37 (1995)
28. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
29. Liu, T., Yang, Y., Wan, H., Zeng, H., Chen, Z., Ma, W.: Support vector machines classification with a very large-scale taxonomy. SIGKDD Explor. **7**(1), 36–43 (2005)
30. Madani, O., Connor, M.: Large-scale many-class learning. In: SIAM Data Mining, pp. 846–857 (2008)
31. Breiman, L.: Bagging predictors. Mach. Learn. **24**(2), 123–140 (1996)
32. Ho, T.K.: Random decision forest. In: Proceedings of the Third International Conference on Document Analysis and Recognition, pp. 278–282 (1995)
33. Amit, Y., Geman, D.: Shape quantization and recognition with randomized trees. Neural Comput. **9**(7), 1545–1588 (1997)

34. Ho, T.K.: The random subspace method for constructing decision forests. IEEE Trans. Pattern Anal. Mach. Intell. **20**(8), 832–844 (1998)
35. Murthy, S., Kasif, S., Salzberg, S., Beigel, R.: OC1: randomized induction of oblique decision trees. In: Proceedings of the Eleventh National Conference on Artificial Intelligence, pp. 322–327 (1993)
36. Carvalho, D., Freitas, A.: A hybrid decision tree/genetic algorithm method for data mining. Inf. Sci. **163**(1–3), 13–35 (2004)
37. Zhou, Z.H., Chen, Z.Q.: Hybrid decision tree. Knowl. Based Syst. **15**(8), 515–528 (2002)
38. Maji, P.: Efficient design of neural network tree using a new splitting criterion. Neurocomputing **71**(4–6), 787–800 (2008)
39. Loh, W.Y., Vanichsetakul, N.: Tree-structured classification via generalized discriminant analysis (with discussion). J. Am. Stat. Assoc. **83**, 715–728 (1988)
40. Yildiz, O., Alpaydin, E.: Linear discriminant trees. Int. J. Pattern Recognit. Artif. Intell. **19**(3), 323–353 (2005)
41. Wu, W., Bennett, K., Cristianini, N., Shawe-Taylor, J.: Large margin trees for induction and transduction. In: Proceedings of the Sixth International Conference on Machine Learning, pp. 474–483 (1999)
42. Fung, G., Mangasarian, O.: Proximal support vector classifiers. In: Proceedings KDD-2001: Knowledge Discovery and Data Mining, pp. 77–86 (2001)
43. Menze, B.H., Kelm, B.M., Splitthoff, D.N., Koethe, U., Hamprecht, F.A.: On oblique random forests. In: Proceedings of the 2011 European Conference on Machine Learning and Knowledge Discovery in Databases , vol. Part II, ECML PKDD'11, pp. 453–469. Springer-Verlag, New York (2011)
44. Weston, J., Watkins, C.: Support vector machines for multi-class pattern recognition. In: Proceedings of the Seventh European Symposium on Artificial Neural Networks, pp. 219–224 (1999)
45. Guermeur, Y.: SVM multiclasses, théorie et applications. Thèse HDR, Université Nancy I (2007)
46. Kreßel, U.: Pairwise classification and support vector machines. In: Advances in Kernel Methods: Support Vector Learning, pp. 255–268 (1999)
47. Platt, J., Cristianini, N., Shawe-Taylor, J.: Large margin DAGs for multiclass classification. Adv. Neural Inf. Process. Syst. **12**, 547–553 (2000)
48. Vural, V., Dy, J.: A hierarchical method for multi-class support vector machines. In: Proceedings of the Twenty-first International Conference on Machine Learning, pp. 831–838 (2004)
49. Benabdeslem, K., Bennani, Y.: Dendogram-based SVM for multi-class classification. J. Comput. Inf. Technol. **14**(4), 283–289 (2006)
50. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.: Classification and Regression Trees. Wadsworth International, Boston (1984)
51. Whaley, R., Dongarra, J.: Automatically tuned linear algebra software. In: CD-ROM Proceedings of the Ninth SIAM Conference on Parallel Processing for Scientific Computing (1999)
52. Chang, C.C., Lin, C.J.: LIBSVM—a library for support vector machines (2001). http://www.csie.ntu.edu.tw/cjlin/libsvm. Accessed 10 Jan 2012
53. Fix, E., Hodges, J.: Discriminatory analysis: small sample performance. In: Technical Report 21–49-004, USAF School of Aviation Medicine, Randolph Field (1952)
54. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo (1993)
55. Witten, I., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, San Mateo (2005)