2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing
and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress

Classifying very high-dimensional and large-scale multi-class image datasets with Latent-lSVM

Thanh-Nghi Do
College of Information Technology
Can Tho University, Vietnam
UMI UMMISCO 209 (IRD/UPMC)
Email: dtnghi@cit.ctu.edu.vn

François Poulet
University of Rennes I - IRISA
Campus de Beaulieu
35042 Rennes Cedex, France
Email: francois.poulet@irisa.fr

***Abstract*—We propose a new learning algorithm of latent local support vector machines (SVM), called Latent-lSVM for effectively classifying very-high-dimensional and large-scale multi-class image datasets. The common framework of image classification tasks using the Scale-Invariant Feature Transform method (SIFT) and the Bag-of-visual-Words (BoW), leads to hard classification problem with thousands of dimensions and hundreds of classes. Our Latent-lSVM algorithm performs these complex tasks into two main steps. The first one is to use latent Dirichlet allocation (LDA) for assigning the image to some topics (clusters) with the corresponding probabilities. This aim is to reduce the number of classes and the number of datapoints in the cluster compared to the full dataset, followed by the second one: to learn a SVM model for each cluster to non-linearly classify the data locally. The numerical test results on eight real datasets show that the Latent-lSVM algorithm achieves very high accuracy compared to state-of-the-art algorithms. An example of its effectiveness is given with an accuracy of 97.87% obtained in the classification of fingerprint dataset having 5000 dimensions into 559 classes.**

***Index Terms*—Support Vector Machines (SVM), Latent Dirichlet Allocation (LDA), high-dimensional and large-scale multi-class image classification.**

I. Introduction

The classification of images is one of the important research topics in computer vision and machine learning. The purpose is to ask a computer to assign the pre-defined class label to an image. The popular framework for image classification involves the main steps as follows: the feature extraction of images, encoding features, the representation of images and learning classifiers. Therefore, the performance of an image classification system largely depends on the image representation approach and the machine learning scheme. The popular approach (first publications [1], [2]) for representing images uses the Scale-Invariant Feature Transform method (SIFT [3], [4]), the Bag-of-visual-Words representation model (BoW). The SIFT features are locally based on the appearance of the object at particular interest points, invariant to image scale, rotation and also robust to changes in illumination, noise, occlusion. And then, the representation of the image in the BoW model, is constructed from the local descriptors and the counting of the occurrences of visual words in a histogram like fashion.

The image representation in this approach leads to datasets with a very large number of dimensions (e.g. many thousands of visual words with each one containing only a small amount of information). For dealing with these datasets, one solution is to reduce the number of dimensional input spaces, e.g. using probabilistic Latent Semantic Analysis (pLSA [7]) in [5], [6], using Correspondence Analysis (CA [9]) in [8]. Another solution proposed in [10], [11], [12], [13], [14] is to use the learning algorithms such as Support Vector Machines (SVM [15]), ensemble-based models that are suited for classifying very-high-dimensional datasets. Furthermore, the emergence of ImageNet dataset [16], [10], Fingerprint datasets [17] poses more challenges in training classifiers. Fingerprint datasets contain from 57 to 559 individual classes of fingerprint images. ImageNet is much larger in scale and diversity than other benchmark datasets with more than 14 million images and $21,841$ classes. This yields huge classification challenges of very-high-dimensional and large-scale multi-class image datasets.

Our investigation is to propose a new learning algorithm of latent local SVM, called Latent-lSVM to effectively classify very-high-dimensional input spaces and large-scale multi-class image datasets. Instead of building a global SVM model, as done by the classical algorithm which is very difficult to deal with large-scale multi-class datasets, the Latent-lSVM algorithm performs the classification task into two main steps. The Latent-lSVM algorithm uses Latent Dirichlet Allocation (LDA [18]) for assigning the image (the representation in the BoW model) to some topics (clusters). This aim is to reduce the number of classes and the number of datapoints in the cluster compared to the full dataset. Then, the Latent-lSVM algorithm constructs an ensemble of local models (a local one is to non-linearly classify the data locally in each cluster) that are easily trained by the Power mean SVM algorithm (PmSVM [12]). The numerical test results on eight real datasets [17] (with from 57 to 559 classes) showed that the Latent-lSVM algorithm achieves very high accuracy compared to state-of-the-art algorithms, including AdaBoost of decision trees [19], random forests [20] and SVM [15].

The paper is organized as follows. Section II briefly introduces the SVM algorithm. Section III presents our proposed Latent-lSVM algorithm for the non-linear classification of very

978-1-5090-2771-2/16 $31.00 © 2016 IEEE
DOI 10.1109/UIC-ATC-ScalCom-CBDCom-IoP-SmartWorld.2016.98

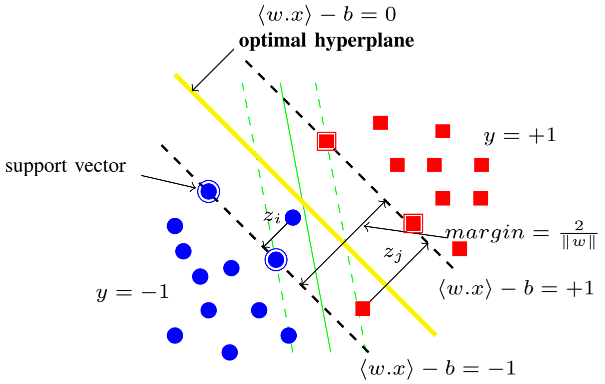
$\langle w.x \rangle - b = 0$
optimal hyperplane
support vector
$y = +1$
z_i
$margin = \frac{2}{\|w\|}$
z_j
$y = -1$
$\langle w.x \rangle - b = +1$
$\langle w.x \rangle - b = -1$

Fig. 1: Classification the datapoints into two classes

high-dimensional and large-scale multi-class image datasets. Section IV shows the experimental results. Section V discusses about related works. We then conclude in section VI.

II. Support vector machines

Let us consider a binary classification problem (simple example in figure 1). The dataset D consists of m datapoints $\{x_1, x_2, \ldots, x_m\}$ in the n-dimensional input space R^n, having corresponding labels $\{y_1, y_2, \ldots, y_m\}$ being ± 1.

For this classification problem, the SVM algorithm [15] tries to find the best separating plane (denoted by the normal vector $w \in R^n$ and the scalar $b \in R$), i.e. furthest from both class $+1$ and class -1. It is accomplished through the maximization of the margin (or the distance) between the supporting planes for each class ($x.w - b = +1$ for class $+1$, $x.w - b = -1$ for class -1). The margin between these supporting planes is $2/\|w\|$ (where $\|w\|$ is the 2-norm of the vector w). Any point x_i falling on the wrong side of its supporting plane is considered to be an error, its error distance denoted by z_i ($z_i \geq 0$). Therefore, SVM has to simultaneously maximize the margin and minimize the error. The standard SVM pursues these goals with the quadratic programming (1).

$$min_\alpha (1/2) \sum_{i=1}^{m} \sum_{j=1}^{m} y_i y_j \alpha_i \alpha_j K\langle x_i, x_j \rangle - \sum_{i=1}^{m} \alpha_i$$

$$s.t. \begin{cases} \sum_{i=1}^{m} y_i \alpha_i = 0 \\ 0 \leq \alpha_i \leq C \quad \forall i = 1, 2, ..., m \end{cases} \tag{1}$$

where C is a positive constant used to tune the margin and the error and a linear kernel function $K\langle x_i, x_j \rangle = \langle x_i \cdot x_j \rangle$.

The support vectors (for which $\alpha_i > 0$) are given by the solution of the quadratic programming (1), and then, the separating surface and the scalar b are determined by the support vectors. The classification of a new data point x based on the SVM model is as follows:

$$predict(x, SVMmodel) = sign(\sum_{i=1}^{\#SV} y_i \alpha_i K\langle x, x_i \rangle - b) \tag{2}$$

Variations on SVM algorithms use different classification functions [21]. No algorithmic changes are required from the usual kernel function $K\langle x_i, x_j \rangle$ as a linear inner product, $K\langle x_i, x_j \rangle = \langle x_i \cdot x_j \rangle$ other than the modification of the kernel function evaluation, including:

- a polynomial function of degree d
  $K\langle x_i, x_j \rangle = (\langle x_i \cdot x_j \rangle + 1)^d$,
- a Radial Basis Function (RBF)
  $K\langle x_i, x_j \rangle = e^{-\gamma \|x_i - x_j\|^2}$.

The SVMs are accurate models for dealing with classification, regression and novelty detection in the very high dimensional datasets. Successful applications of SVMs have been reported for such varied fields including facial recognition, text categorization and bioinformatics [22].

There are two strategies to extend the binary SVM solver for dealing with the multi-class problems (c classes, $c \geq 3$). The first one is considering the multi-class case in one optimization problem [23] [24]. The second one is decomposing multi-class into a series of binary SVMs, including One-Versus-All [15], One-Versus-One [25]. In practice, the most popular methods are One-Versus-All (ref. LIBLINEAR [26]), One-Versus-One (ref. LibSVM [27]) and are due to their simplicity. The One-Versus-All strategy builds c different binary SVM models where the i^{th} one separates the i^{th} class from the rest. The One-Versus-One strategy constructs $c(c-1)/2$ binary SVM models for all the binary pairwise combinations of the c classes. The class is then predicted with the largest distance vote.

III. Latent local support vector machines

In recent applications like the classification of images, the emergence of ImageNet dataset [16], [10], Fingerprint datasets [17] pose more challenges in training SVM models. The popular model for representing images is the bag-of-words (BoW) constructed from the Scale-Invariant Feature Transform (SIFT [3], [4]) extracted in the images. It leads to datasets with a very large number of dimensions (e.g. many thousands of visual words with each one containing only a small amount of information). In addition, these datasets contain large number of classes (hundreds, even thousands of classes). This yields huge classification challenges of the very-high-dimensional and large-scale multi-class image datasets.

We propose a new learning algorithm of SVM, called Latent-lSVM to effectively classify very-high-dimensional input spaces and large-scale multi-class image datasets. Instead of building a global SVM model, as done by the classical algorithm which is very difficult to deal with large-scale multi-class datasets, the Latent-lSVM creates a partition of the full dataset into k joint clusters and then it is easier to learn a non-linear SVM in each cluster to classify the data locally. Figure 2 shows the comparison between a global SVM model (left part) and 3 local SVM models (right part), using a non-linear RBF kernel function with $\gamma = 10^2$ and a positive constant $C = 10^5$.

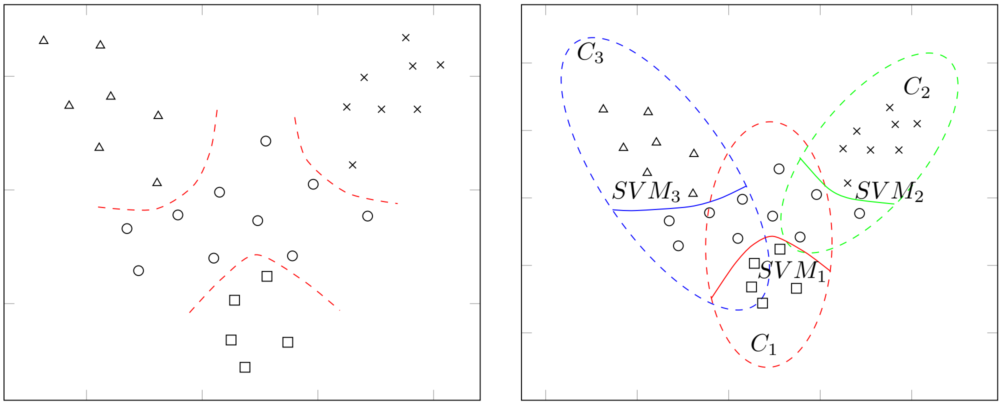

Fig. 2: Global SVM model (left part) versus local SVM models (right part)

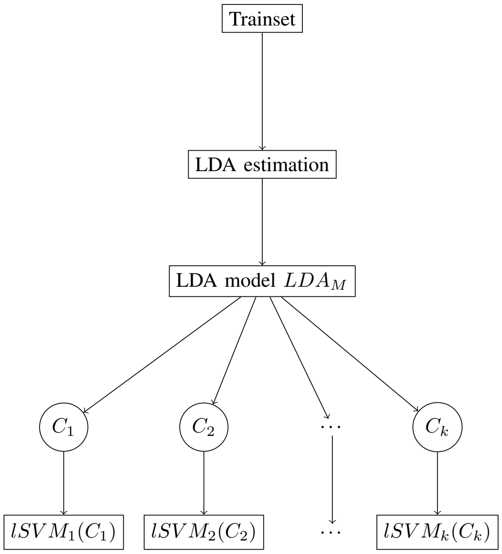

Fig. 3: Training algorithm of latent local SVM models

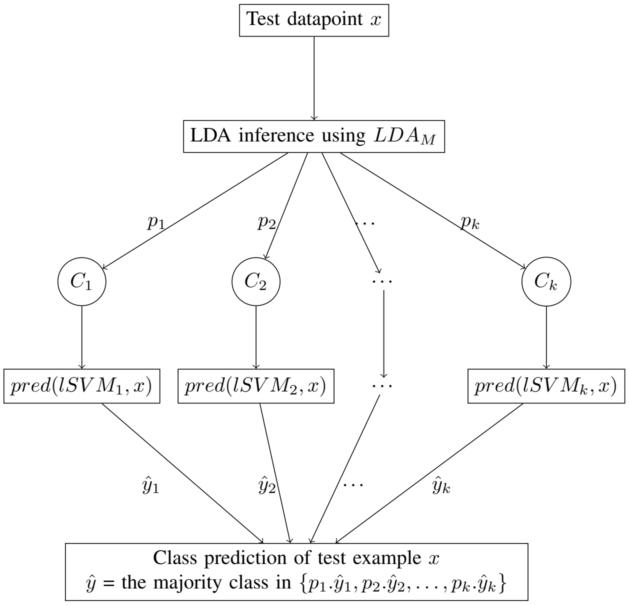

Fig. 4: Prediction of x with latent local SVM models

A. Learning latent local SVM models

The learning algorithm Latent-lSVM is described in figure 3. The first step of the training process is to learn a Latent Dirichlet Allocation (LDA [18] model, being well-known as an effective model type for the very-high-dimensional data represented in the format of the BoW model), denoted by LDA_M for partitioning the full dataset $Trainset$ into k joint clusters $C_1, C_2, \ldots, C_k$. It assumes that the datapoints belonging to the nearest classes have the same distribution (the homogeneous group). And then, the number of classes and the number of datapoints in each data cluster $C_i (i = 1, k)$ is less than the number of classes and the number of datapoints in the full dataset, as shown in figure 2. Therefore, the construction of an ensemble of local SVM models $lSVM_1(C_1), lSVM_2(C_2), \ldots, lSVM_k(C_k)$ makes the second step easier than the global SVM model. We obtain the Latent-lSVM model, denoted by $Latent\text{-}lSVM\text{-}model$ = $\{LDA_M, lSVM_1, lSVM_2, \ldots, lSVM_k\}$.

B. Prediction in latent local SVM models

The prediction of the class for a new datapoint x is decribed in figure 4. The LDA inference step based on the LDA model LDA_M is to assign the datapoint x to the clusters $C_1, C_2, \ldots, C_k$ with the corresponding probabilities $p_1, p_2, \ldots, p_k$. Then, the local models $lSVM_1, lSVM_2, \ldots, lSVM_k$ are used to predict the class of x and the results are: $\hat{y}_1 = pred(lSVM_1, x)$, $\hat{y}_2 = pred(lSVM_2, x)$, ..., $\hat{y}_k = pred(lSVM_k, x)$. Finally, the datapoint x is predicted in the class $\hat{y}$ with the largest vote among the prediction classes $\{p_1.\hat{y}_1, p_2.\hat{y}_2, \ldots, p_k.\hat{y}_k\}$.

$$\hat{y} = \text{the majority class in } \{p_1.\hat{y}_1, p_2.\hat{y}_2, \ldots, p_k.\hat{y}_k\} \tag{3}$$

C. Performance analysis

Let us now examine the classification performance of k local SVM models with the Latent-lSVM algorithm. Turn back to theorem 5.2 proposed by Vapnik in [15].

Theorem 5.2 ([15] p.139). If training sets containing m examples are separated by the maximal margin hyperplanes, the expectation (over training sets) of the probability of test error is bounded by the expectation of the minimum of three values: the ratio $\frac{sv}{m}$, where sv is the number of support vectors, the ratio $\frac{1}{m}\frac{R^2}{\Delta}$, where R is the radius of the sphere containing the data and Δ is the value of the margin, and the ratio $\frac{n}{m}$, where n is the dimensionality of the input space:

$$EP_{error} \leq E\left\{min\left(\frac{sv}{m}, \frac{1}{m}\left[\frac{R^2}{\Delta}\right], \frac{n}{m}\right)\right\} \quad (4)$$

Theorem 5.2 illustrates that the maximal margin hyperplane found by the minimization of $\left[\frac{R^2}{\Delta}\right]$ can generalize well. It means that the generalization ability of the large margin hyperplane is high.

In the Latent-lSVM, the full dataset with m datapoints is partitioned into k clusters (the cluster size is about $m_k = \alpha_k \frac{m}{k}$ with $0 < \alpha_k < k$). Here, the index notation k is used to present m in the context of the cluster (subset). And then the expectation of the probability of test error for a local SVM model (learnt from a cluster) is bounded by:

$$EP_{error} \leq E\left\{min\left(\frac{sv_k}{m_k}, \frac{1}{m_k}\left[\frac{R_k^2}{\Delta_k}\right], \frac{n}{m_k}\right)\right\} \quad (5)$$

Without loss of generality, we consider a binary classification problem because two most popular methods One-Versus-All, One-Versus-One decompose a multi-class problem into a series of binary ones.

The performance analysis starts with the comparison between the margin size of the global SVM model for the full dataset and the local SVM model learnt from a cluster illustrated in theorem 1.

Theorem 1 Given a dataset with m datapoints $X = \{x_1, x_2, \ldots, x_m\}$ in the n-dimensional input space R^n, having corresponding labels $Y = \{y_1, y_2, \ldots, y_m\}$ being ± 1, a maximal margin Δ_X hyperplane is to separate furthest from both class $+1$ and class -1, there exists a maximal margin Δ_{X_k} hyperplane for separating a subset of m_k datapoints $X_k \subset X$ into two classes so that the inequality $\Delta_{X_k} \geq \Delta_X$ holds.

Proof We remark that the maximal margin Δ_X hyperplane can be seen as the minimum distance between two convex hulls, $H+$ of the positive class P and $H-$ of the negative class N (the farthest distance between the two classes, illustrated in figure5). For subset $X_k \subset X$ containing the subset of the positive class $P_k \subset P$ and the subset of the negative class $N_k \subset N$, it leads to the reduced convex hull H_k+ of $H+$ for the positive class and the reduced convex hull H_k- of $H-$ for the negative class. And then the minimum distance between H_k+ and H_k- can not be smaller than between $H+$ and $H-$. It means that the maximal margin Δ_{X_k} hyperplane for X_k is larger than the maximal margin Δ_X one for fullset X.

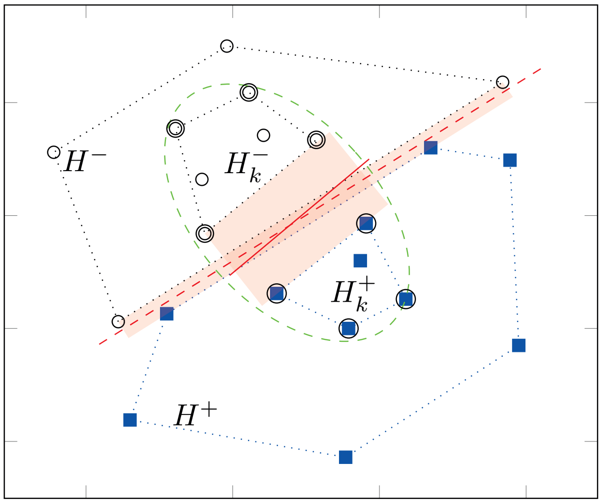

Fig. 5: The comparison of the maximal margin hyperplane of the global SVM and the local SVM

The classification performance of a local SVM model in the Latent-lSVM is studied in term $\frac{1}{m_k}\left[\frac{R_k^2}{\Delta_k}\right]$ in equation 5. In the comparison with the global SVM constructed for the full dataset X, a local SVM model using a subset $X_k \subset X$ of m_k datapoints can guarantee the classification performance because there exists a compromise between the locality (the subset size, i.e. $R_k \leq R$ and $m_k \leq m$) and the generalized capacity (the margin size, i.e. consequence of theorem 1 $\Delta_{X_k} \geq \Delta_X$).

IV. Evaluation

We are interested in terms of classification correctness of our proposal (the Latent-lSVM algorithm) for classifying the very-high-dimensional and large-scale multi-class image datasets. Therefore, we here report the comparison of the classification performance obtained by Latent-lSVM and the best state-of-the-art algorithms, including SVM [15], AdaBoost of J48 (AdaBoost-J48 [19]) and random forests (RF-CART [20]).

A. Software programs

In order to evaluate the effectiveness in classification tasks, we have implemented Latent-lSVM in C/C++, OpenMP [29], using the parallel latent Dirichlet allocation program (PLDA+ [30]) and the highly efficient Power mean SVM (PmSVM [12] with One-Versus-All strategy for multi-class).

PmSVM replaces the kernel function $K\langle x_i, x_j\rangle$ in (1) and (2) of the standard SVM with the power mean kernel

$M_p\langle x_i, x_j\rangle$ (x_i and $x_j \in R^n_+$), which is well-known as a general form of many additive kernels (e.g. χ^2 kernel, histogram intersection kernel or Hellinger's kernel):

$$M_p\langle x_i, x_j\rangle = \sum_{z=1}^{n}(x^p_{i,z} + x^p_{j,z})^{\frac{1}{p}} \tag{6}$$

where $p \in R$ is a constant.

PmSVM also uses the coordinate descent method [31] for dealing with training tasks. Furthermore, the gradient computation step of the coordinate descent algorithm and the parameter p can be estimated approximately by using polynomial regression with very low cost in both training and testing tasks. Therefore, the use of PmSVM in our Latent-lSVM implementation pursues the interesting goals of the complexity reduction (low computational cost) and without parameter.

We also use the highly efficient standard SVM algorithm LibSVM [27] with One-Versus-One strategy for multi-class. The rest (AdaBoost-J48, RF-CART) are implemented in Weka library [32]. Due to the different programming languages (C/C++, Java) used for the implementation of the algorithms, therefore we do not report the comparison of computational time.

All experiments are run on PC with Linux Fedora 20, Intel(R) Core i7-4790 CPU, 3.6 GHz, 4 cores and 32 GB main memory.

B. Fingerprint image datasets

We do setup experiment with seven real fingerprint datasets from our previous research [17] for comparative studies. Fingerprints acquisition was done with Microsoft Fingerprint Reader (optical fingerprint scanner, resolution: 512 DPI, image size: 355x390, colors: 256 levels grayscale). Datasets FP-57, FP-78, ..., and FP-559 are the fingerprint images of 57, 78, ..., and 559 colleagues respectively (between 15 and 20 fingerprints were captured for each individual - class label). And then, local descriptors are extracted with the Hessian-Affine SIFT detector proposed in [33]. k-means algorithm [34] is used to group the descriptors into 5000 clusters [1]. The datasets are described in table I.

The evaluation protocol is illustrated in the last column of table I. The datasets are already divided into training set (Trn) and testing set (Tst). We used the training data to build and tune the parameters of classification models. Then, we classify the testing set using the resulting models.

C. Tuning parameters

We propose to use RBF kernel type in SVM models because it is general and efficient [35]. We also tried to tune the hyper-parameter γ of RBF kernel (RBF kernel of two individuals x_i, x_j, $K[i, j] = exp(-\gamma\|x_i - x_j\|^2)$) and the cost C (a trade-off between the margin size and the errors) to obtain the best correctness. The optimal parameters with $\gamma = 0.0001$, $C = 100000$ give the highest accuracy for all datasets.

AdaBoost-J48 and RF-CART build 200 trees. The out-of-bag samples (the out of the bootstrap samples) are used during the forest construction to find the parameters of RF-CART (with $p' = 1000$ random dimensions for non-terminal node splitting, $min\_obj = 2$ for early stopping).

Latent-lSVM requires to tune the Dirichlet hyper-parameters of LDA. According to heuristical method proposed by [36], the good model quality has been reported for $\beta = 0.01$ and $\alpha = \frac{50}{T}$ with the number of topics (clusters) T. Furthermore, [37] illustrates that selecting the number of topics T is one of the most important modeling choices. Latent-lSVM uses $\beta = 0.01$ and $alpha$, T so that each cluster has about 500 individuals. The idea gives a trade-off between the generalization capacity [38] and the computational cost. Table II presents the hyper-parameters of Latent-lSVM used in the classification.

D. Classification results

The classification results are given in table III and figure 6.

Latent-lSVM and LibSVM outperform RF-CART and Adaboost-J48 in the classification of all datasets. The results show that RF-CART has a slight superiority against Adaboost-J48 (mean rank score of respectively 3.1 and 3.9). The accuracies of RF-CART and AdaBoost-J48 are already somewhat less affected by the increase in the number of classes, decreasing from 93.5% to 84.07% for RF-CART and from 91.5% to 72.13% for Adaboost-J48.

LibSVM holds the rank 2 on each experimented dataset, with a mean accuracy of 93.44%, while Latent-lSVM gets the best result on each of the eight datasets with an average accuracy of 98.09%, which corresponds to an improvement of 4.65 percentage points compared with LibSVM. This superiority of Latent-lSVM on LibSVM is statistically significant, in so far as according to the signed rank test, the p-value of the observed results (8 wins of Latent-lSVM on LibSVM with 8 datasets) is equal to 0.007813. In addition, these two methods lose only little efficiency when the number of classes increases, since the corresponding accuracies decrease from 98.30% to 97.87% for Latent-lSVM and 95.5% to 93.67% for LibSVM.

V. Discussion on related works

Our proposal is in some aspects related to local SVM learning algorithms. The first approach is to classify data in hierarchical strategy. This kind of training algorithm performs the classification task with two main steps. The first one is to cluster the full dataset into homogeneous groups (clusters) and then the second one is to learn the local supervised classification models from clusters. The paper of [39] proposed to use the expectation-maximization (EM) clustering algorithm [40] for partitioning the training set into k joint clusters (the EM clustering algorithm makes a soft assignment based on the posterior probabilities [41]); for each cluster, a neural network (NN) is learnt to classify the individuals in the cluster. The parallel mixture of SVMs algorithm proposed by [42] constructs local SVM models instead of NN ones in [39].

[1] The number of clusters/visual words was optimized between 500 and over 10000, 5000 clusters are the optimum in this experiment [17]

ID	**Dataset**	**#Datapoints**	**#Dimensions**	**#Classes**	**Evaluation protocol**
1	FP-57	1052	5000	57	700 Trn - 352 Tst
2	FP-78	1372	5000	78	950 Trn - 422 Tst
3	FP-120	1918	5000	120	1438 Trn - 480 Tst
4	FP-153	2372	5000	153	1700 Trn - 672 Tst
5	FP-185	2765	5000	185	2000 Trn - 765 Tst
6	FP-235	3485	5000	235	2485 Trn - 1000 Tst
7	FP-389	6306	5000	389	4306 Trn - 2000 Tst
8	FP-559	10270	5000	559	7270 Trn - 3000 Tst

TABLE I: Description of datasets

ID	**Dataset**	**#Clusters** (T)	**Alpha** (α)	**Beta** (β)
1	FP-57	10	10	0.01
2	FP-78	15	10	0.01
3	FP-120	15	10	0.01
4	FP-153	15	10	0.01
5	FP-185	15	10	0.01
6	FP-235	30	10	0.01
7	FP-389	30	10	0.01
8	FP-559	30	10	0.01

TABLE II: Parameters of LDA used in Latent-lSVM

		Classification accuracy(%)			
ID	**Dataset**	**AdaBoost-J48**	**RF-CART**	**LibSVM**	**Latent-lSVM**
1	FP-57	91.48	93.47	95.46	98.30
2	FP-78	89.34	92.42	94.79	98.58
3	FP-120	89.17	88.33	92.50	99.05
4	FP-153	84.52	91.67	92.86	97.32
5	FP-185	85.10	89.02	93.46	97.65
6	FP-235	84.10	87.50	92.10	98.20
7	FP-389	81.95	86.30	92.65	97.75
8	FP-559	72.13	84.07	93.67	97.87

TABLE III: Classification results in terms of accuracy (%)

CSVM [43] uses k-means algorithm [34] to partition the full dataset into k disjoint clusters; then, the algorithm learns weighted local linear SVMs from clusters. More recent kSVM [44] and krSVM [45] (random ensemble of kSVM) propose to parallely train the local non-linear SVMs instead of weighting linear ones of CSVM. DTSVM [46], [47] uses the decision tree algorithm [48], [49] to split the full dataset into disjoint regions (tree leaves) and then the algorithm builds the local SVMs for classifying the individuals in tree leaves. These algorithms aim at speeding up the learning time.

The second approach is to learn local supervised classification models from k nearest neighbors (kNN) of a new testing individual. First local learning algorithm of Bottou & Vapnik [50] find kNN of a test individual; train a neural network with only these k neighborhoods and apply the resulting network to the test individual. k-local hyperplane and convex distance nearest neighbor algorithms are also proposed in [51]. More recent local SVM algorithms aim to use the different methods for kNN retrieval; including SVM-kNN [52] trying with different metrics, ALH [53] using weighted distance and features, FaLK-SVM [54] speeding up the kNN retrieval with the cover tree [55].

A theorical analysis for such local algorithms discussed in [56] introduces the trade-off between the capacity of learning system and the number of available individuals. The size of the neighborhoods is used as an additional free parameters to control generalisation capacity against locality of local learning algorithms.

VI. Conclusion and future works

We have presented a novel learning algorithm Latent-lSVM that achieves high performances for classifying very high-

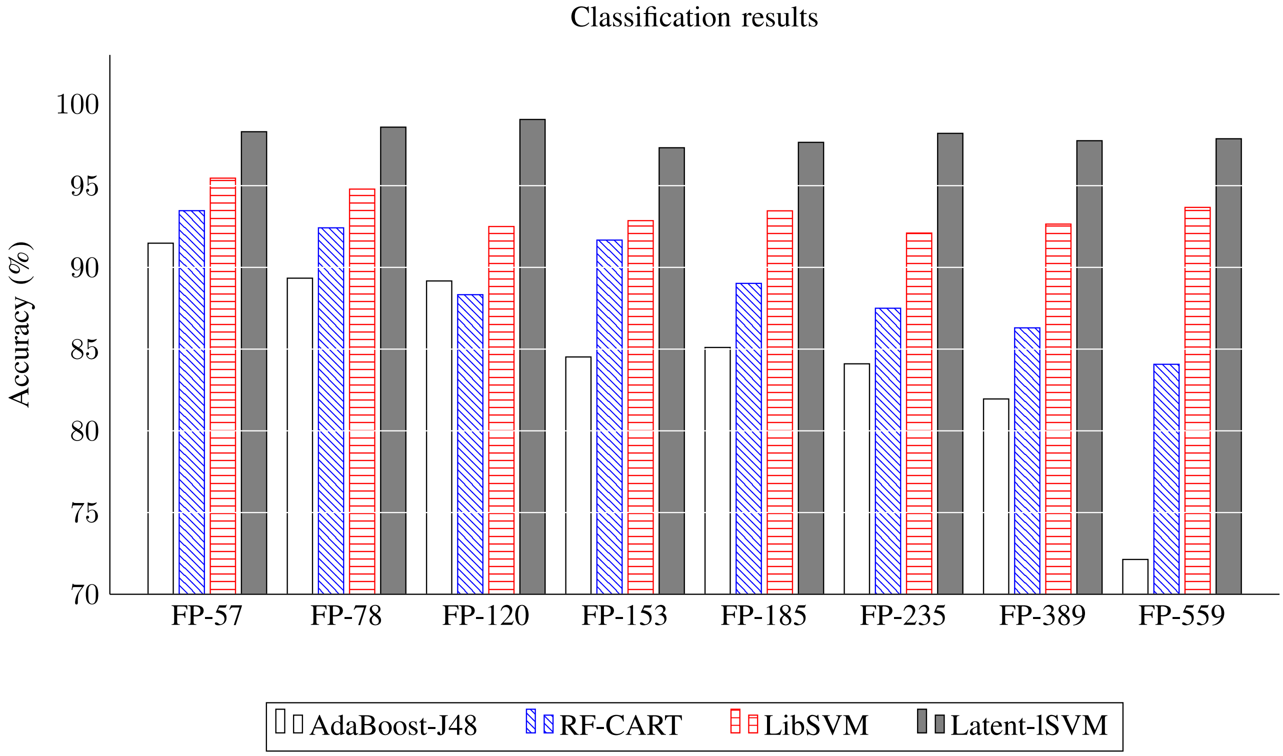

Fig. 6: Comparison of classification results

dimensional and large-scale multi-class image datasets. Latent-lSVM algorithm uses LDA to group images into clusters to reduce the number of datapoints and the number of training classes. Then the Latent-lSVM learns the PmSVM model for each cluster to non-linearly classify the data locally. The experimental results on seven real datasets of fingerprint images showed that Latent-lSVM algorithm is very efficient in comparison with RF-CART, AdaBoost of J48 and SVM (the average improvement goes from 4.65% to 13.37%). Latent-lSVM achieves an accuracy of 97.87% in the classification of fingerprint dataset having 5000 dimensions into 559 classes.

In the near future, we intend to develop a distributed implementation for large scale processing on an in-memory cluster-computing platform, Apache Spark [57] (running times or up to 100x faster than Hadoop MapReduce, or 10x faster on disk). A promising future research aims at automatically tuning the hyper-parameters of Latent-lSVM. We would like to provide more empirical test on large scale benchmarks like ImageNet datasets [16], [10] and comparisons with other large scale linear SVM solvers [26], [58], [13].

References

[1] Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In: 9th IEEE International Conference on Computer Vision (ICCV 2003), 14-17 October 2003, Nice, France. (2003) 1470–1477

[2] Li, F., Perona, P.: A Bayesian hierarchical model for learning natural scene categories. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA. (2005) 524–531

[3] Lowe, D.: Object recognition from local scale invariant features. In: Proceedings of the 7th International Conference on Computer Vision. (1999) 1150–1157

[4] Lowe, D.: Distinctive image features from scale invariant keypoints. International Journal of Computer Vision (2004) 91–110

[5] Bosch, A., Zisserman, A., Muñoz, X.: Scene classification via pLSA. In: Proceedings of the European Conference on Computer Vision. (2006) 517–530

[6] Deselaers, T., Pimenidis, L., Ney, H.: Bag-of-visual-words models for adult image classification and filtering. In: Proceeding of The 19th International Conference on Pattern Recognition. (2008) 1–4

[7] Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, USA, ACM (1999) 50–57

[8] Pham, N-K., Morin, A.: Une nouvelle approche pour la recherche d'images par le contenu. In: Extraction et gestion des connaissances (EGC'2008), Actes des 8èmes journées Extraction et Gestion des Connaissances, Sophia-Antipolis, France, 29 janvier au 1er février 2008, 2 Volumes. (2008) 475–486

[9] Benzécri, J.: L'analyse des correspondances. Paris: Dunod (1973)

[10] Deng, J., Berg, A.C., Li, K., Li, F.: What does classifying more than 10, 000 image categories tell us? In: Computer Vision - ECCV 2010 - 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part V. (2010) 71–84

[11] Do, T-N.: Detection of pornographic images using bag-of-visual-words and arcx4 of random multinomial naive bayes. In: Proceedings of the 4th Intl Conf. on Theories and Applications of Computer Science. (2011) 13–24

[12] Wu, J.: Power mean svm for large scale visual classification. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition. (2012) 2344–2351

[13] Do, T-N.: Parallel multiclass stochastic gradient descent algorithms

for classifying million images with very-high-dimensional signatures into thousands classes. Vietnam J. Computer Science **1**(2) (2014) 107–115

[14] Doan, T-N., Do, T-N., Poulet, F.: Large scale classifiers for visual classification tasks. Multimedia Tools Appl. **74**(4) (2015) 1199–1224

[15] Vapnik, V.: The Nature of Statistical Learning Theory. Springer-Verlag; 2nd edition (2000)

[16] Deng, J., Dong, W., Socher, R., Li, L., Li, K., Li, F.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA. (2009) 248–255

[17] Do, T-N., Lenca, P., Lallich, S.: Classifying many-class high-dimensional fingerprint datasets using random forest of oblique decision trees. Vietnam J. Computer Science **2**(1) (2015) 3–12

[18] Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of Machine Learning Research **3** (March 2003) 993–1022

[19] 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. (1995) 23–37

[20] Breiman, L.: Random forests. Machine Learning **45**(1) (2001) 5–32

[21] Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines: And Other Kernel-based Learning Methods. Cambridge University Press, New York, NY, USA (2000)

[22] Guyon, I.: Web page on svm applications (1999) http://www.clopinet.com/isabelle/Projects/SVM/applist.html.

[23] Weston, J., Watkins, C.: Support vector machines for multi-class pattern recognition. In: Proceedings of the Seventh European Symposium on Artificial Neural Networks. (1999) 219–224

[24] Guermeur, Y.: Svm multiclasses, théorie et applications (2007)

[25] Kreßel, U.: Pairwise classification and support vector machines. Advances in Kernel Methods: Support Vector Learning (1999) 255–268

[26] Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: LIBLINEAR: A library for large linear classification. Journal of Machine Learning Research **9**(4) (2008) 1871–1874

[27] Chang, C.C., Lin, C.J.: LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology **2**(27) (2011) 1–27

[28] Platt, J.: Fast training of support vector machines using sequential minimal optimization. Advances in Kernel Methods Support Vector Learning, B. Schölkopf, C. Burges, and A. Smola Eds (1999) 185–208

[29] OpenMP Architecture Review Board: OpenMP application program interface version 3.0 (2008)

[30] Liu, Z., Zhang, Y., Chang, E.Y., Sun, M.: Plda+: Parallel latent dirichlet allocation with data placement and pipeline processing. ACM Trans. Intell. Syst. Technol. **2**(3) (May 2011) 26:1–26:18

[31] Yuan, G.X., Ho, C.H., Lin, C.J.: Recent advances of large-scale linear classification. Proceedings of the IEEE **100**(9) (2012) 2584–2603

[32] Witten, I., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann (2005)

[33] Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. International Journal of Computer Vision **60**(1) (2004) 63–86

[34] MacQueen, J.: Some methods for classification and analysis of multivariate observations. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press **1** (January 1967) 281–297

[35] Lin, C.: A practical guide to support vector classification (2003)

[36] Heinrich, G.: Parameter estimation for text analysis. Technical report, University of Leipzig (2004)

[37] Wallach, H.M., Mimno, D.M., McCallum, A.: Rethinking LDA: why priors matter. In: Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada. (2009) 1973–1981

[38] Vapnik, V., Bottou, L.: Local algorithms for pattern recognition and dependencies estimation. Neural Computation **5**(6) (1993) 893–909

[39] Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural Computation **3**(1) (1991) 79–87

[40] Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. Journal of the royal statistical society, Series B **39**(1) (1977) 1–38

[41] Bishop, C.M.: Pattern Recognition and Machine Learning. Springer-Verlag New York (2006)

[42] Collobert, R., Bengio, S., Bengio, Y.: A parallel mixture of SVMs for very large scale problems. Neural Computation **14**(5) (2002) 1105–1114

[43] Gu, Q., Han, J.: Clustered support vector machines. In: Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2013, Scottsdale, AZ, USA, April 29 - May 1, 2013. Volume 31 of JMLR Proceedings. (2013) 307–315

[44] Do, T-N.: Non-linear classification of massive datasets with a parallel algorithm of local support vector machines. In: Advanced Computational Methods for Knowledge Engineering. Springer International Publishing (2015) 231–241

[45] Do, T-N., Poulet, F.: Random local svms for classifying large datasets. In: Future Data and Security Engineering - Second International Conference, FDSE 2015, Ho Chi Minh City, Vietnam, November 23-25, 2015, Proceedings. Volume 9446 of Lecture Notes in Computer Science., Springer (2015) 3–15

[46] Chang, F., Guo, C.Y., Lin, X.R., Lu, C.J.: Tree decomposition for large-scale SVM problems. Journal of Machine Learning Research **11** (2010) 2935–2972

[47] Chang, F., Liu, C.C.: Decision tree as an accelerator for support vector machines. In Ding, X., ed.: Advances in Character Recognition. InTech (2012)

[48] Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.: Classification and Regression Trees. Wadsworth International, (1984)

[49] Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo, CA (1993)

[50] Bottou, L., Vapnik, V.: Local learning algorithms. Neural Computation **4**(6) (1992) 888–900

[51] Vincent, P., Bengio, Y.: K-local hyperplane and convex distance nearest neighbor algorithms. In: Advances in Neural Information Processing Systems, The MIT Press (2001) 985–992

[52] Zhang, H., Berg, A., Maire, M., Malik, J.: SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Volume 2. (2006) 2126–2136

[53] Yang, T., Kecman, V.: Adaptive local hyperplane classification. Neurocomputing **71**(1315) (2008) 3001–3004

[54] Segata, N., Blanzieri, E.: Fast and scalable local kernel machines. Journal Machine Learning Research **11** (2010) 1883–1926

[55] Beygelzimer, A., Kakade, S., Langford, J.: Cover trees for nearest neighbor. In: Proceedings of the 23rd international conference on Machine learning, ACM (2006) 97–104

[56] Vapnik, V.: Principles of risk minimization for learning theory. In: Advances in Neural Information Processing Systems 4, [NIPS Conference, Denver, Colorado, USA, December 2-5, 1991]. (1991) 831–838

[57] Zaharia, M., Chowdhury, M., Franklin, M.J., Shenker, S., Stoica, I.: Spark: Cluster computing with working sets. In: Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing. HotCloud'10, Berkeley, CA, USA, USENIX Association (2010) 10–10

[58] Poulet, F., Pham, N.: High dimensional image categorization. In: Advanced Data Mining and Applications - 6th International Conference, ADMA 2010. (2010) 465–476

Acknowledgement

This work is supported by EUH2020 project-AniAge (691215).