Received: 16 December 2016 | Revised: 4 April 2017 | Accepted: 24 May 2017

DOI: 10.1002/cpe.4224

WILEY

SPECIAL ISSUE PAPER

Latent-ISVM classification of very high-dimensional and large-scale multi-class datasets

Thanh-Nghi Do[1] | **François Poulet**[2]

[1]UMI UMMISCO 209 (IRD/UPMC), Can Tho University, Cantho, 92100, Vietnam
[2]University of Rennes I - IRISA, Campus de Beaulieu Rennes Cedex, 35042, France

Correspondence
Thanh-Nghi Do, Can Tho University, 92100-Cantho, Vietnam.
Email: dtnghi@cit.ctu.edu.vn

Summary

We propose a new parallel learning algorithm of latent local support vector machines (SVM), called latent-ISVM for effectively classifying very high-dimensional and large-scale multi-class datasets. The common framework of texts/images classification tasks using the Bag-Of-(visual)-Words model for the data representation leads to hard classification problem with thousands of dimensions and hundreds of classes. Our latent-ISVM algorithm performs these complex tasks into two main steps. The first one is to use latent Dirichlet allocation for assigning the datapoint (text/image) to some topics (clusters) with the corresponding probabilities. This aims at reducing the number of classes and the number of datapoints in the cluster compared to the full dataset, followed by the second one: to learn in a parallel way nonlinear SVM models to classify data clusters locally. The numerical test results on nine real datasets show that the latent-ISVM algorithm achieves very high accuracy compared to state-of-the-art algorithms. An example of its effectiveness is given with an accuracy of 70.14% obtained in the classification of Book dataset having 100 000 individuals in 89 821 dimensional input space and 661 classes in 11.2 minutes using a PC Intel(R) Core i7-4790 CPU, 3.6 GHz, 4 cores.

KEYWORDS

Latent Dirichlet allocation (LDA), high-dimensional and large-scale multi-class data classification, parallel learning on multi-core computers, support vector machines (SVMs)

1 | INTRODUCTION

There are more and more multimedia data stored electronically, with increasing number of internet users and mobile internet access sharing videos, songs, or photos. There are more than 1 billion daily active users—nearly one-third of all people on the Internet (around 46% of the world population)—on Youtube and Facebook (Amazon and Yahoo! have even more), 600 000 hours (68 years) of videos are uploaded on Youtube every day, and 46 000 years are viewed at the same time. Almost all mobile phones can take photos: 2 trillions photos will be shared this year. There are 310 millions Twitter users and more than 600 millions Weibo users (the "Chinese Twitter';' Asia is the first Internet region with more than 50% of Internet users in 2016). The number of data is always increasing and their sizes too: 4K or 3D videos, sound in Dolby 5.1, higher and higher photo resolution, text messages replaced by voice messages. This leads to very huge amount of data; there is a need for high performance classification algorithms in order to help us find what we are looking for. We present a new fast and accurate parallel local support vector machine (SVM) algorithm for the classification of very large scale and high-dimensional multi-class datasets. The experimental results are performed on two different kinds of datasets: image and text classification.

The classification of texts/images is one of the important research topics in text mining, computer vision, and machine learning. The purpose is to ask a computer to assign the predefined class label to the text/image. The popular frameworks for text/image classification involve the main steps as follows: the feature extraction of texts/images, encoding features, the representation of texts/images, and learning classifiers.[1–3]

In text categorization tasks, the common approach consists of the word splitting, the representation of texts with the bag-of-words (BoW[4]) model and training supervised classifiers such as naïve Bayes, decision trees, and SVMs.[5] The dictionary has many thousands vocabulary words. Therefore, the BoW representation brings out datasets with a very large number of dimensions. And then, the SVM model is suited for classifying this kind of

data without any feature selection or reduction methods.[6–8] First benchmarks includes 20-Newsgroups,[9] Reuters-2[10] and RCV1[11] with 20 118 103 classes, respectively. More recently, our system of the Book classification at the Learning Resource Center of Can Tho University in Vietnam handles the Book dataset with 661 subjects. Partalas et his colleagues [12] proposed a benchmark LSHTC for large-scale text classification with the number of classes up to 325056. It is very difficult to train an automatic classification model for efficiently dealing with such very-high-dimensional and large-scale multi-class datasets.

In image classification tasks, the performance of a system largely depends on the image representation approach and the machine learning scheme. The popular approach (first publications, Sivic and Zisserman[3] and Li and Perona[13]) for representing images uses the Scale-Invariant Feature Transform method (SIFT [14,15]), 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, and 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 (eg, 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, eg, using probabilistic latent semantic analysis (pLSA [16]) in Bosch et al[17] and Deselaers et al,[18] using Correspondence Analysis[19] in Pham and Morin.[20] Another solution proposed in previous studies[21–25] is to use the learning algorithms such as SVM,[5] ensemble-based models that are suited for classifying very high-dimensional datasets. Furthermore, the emergence of ImageNet dataset, [21,26] Fingerprint datasets [27] 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.

The new benchmarks of text/image classification tasks yield huge classification challenges of very high-dimensional and large-scale multi-class datasets.

In this extended version of Do and Poulet,[28] we propose a new parallel learning algorithm of latent local SVM, called latent-ISVM to effectively classify very high-dimensional input spaces and large-scale multi-class 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-ISVM algorithm performs the classification task into two main steps. The latent-ISVM algorithm uses latent Dirichlet allocation (LDA [29]) for assigning the datapoint (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-ISVM algorithm constructs in parallel an ensemble of local models (a local one is to nonlinearly classify the data locally in each cluster) that are easily trained by the power mean SVM algorithm (PmSVM [23]). The numerical test results on nine real datasets [27] (with from 57 to 661 classes) showed that the latent-ISVM algorithm achieves very high accuracy compared to state-of-the-art algorithms, including AdaBoost of decision trees,[30] random forests,[31] and SVM.[5]

The paper is organized as follows. Section 2 briefly introduces the SVM algorithm. Section 3 presents our proposed latent-ISVM algorithm for the nonlinear classification of very high-dimensional and large-scale multi-class datasets in multicore computers. Section 4 shows the experimental results. Section 5 discusses about related works. We then conclude in Section 6.

2 | SUPPORT VECTOR MACHINES

2.1 | Support vector classification for two classes

Let us consider a binary classification problem (simple two-dimensional example depicted in Figure 1) with the dataset D consisting 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, there are many possible ways to separate the data into two classes. The SVM algorithm porposed by Vapnik [5] tries to find the best separating plane (denoted by the normal vector $w \in R^n$ and the scalar $b \in R$), ie, 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

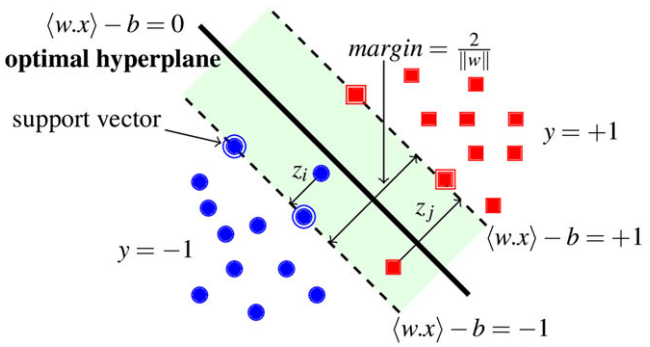

FIGURE 1 Classification of the datapoints into two classes

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 ($z_i \geqslant 0$). Therefore, SVM has to simultaneously maximize the margin and minimize the error. The standard SVM pursues these goals with the quadratic programming 1.

$$min_{\alpha}(1/2)\sum_{i=1}^{m}\sum_{j=1}^{m} y_i y_j \alpha_i \alpha_j K\langle x_i, x_j\rangle - \sum_{i=1}^{m}\alpha_i$$
$$s.t.\begin{cases}\sum_{i=1}^{m} y_i\alpha_i = 0\\ 0 \leqslant \alpha_i \leqslant C \quad \forall i = 1, 2, \ldots, m,\end{cases} \tag{1}$$

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:

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

The study in Platt[32] illustrated that the computational cost requirements of the SVM solutions in Equation 1 are at least $O(m^2)$ (where m is the number of training datapoints). Variations on SVM algorithms use different classification functions.[33] 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 $dK\langle x_i, x_j\rangle = (\langle x_i \cdot x_j\rangle + 1)^d$,
- 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,[34] and quality control of beer.[35]

2.2 | Support vector classification for multi-class

There are two strategies to extend the binary SVM solver for dealing with the multi-class problems (c classes, $c \geqslant 3$). The first one is considering the multi-class case in one optimization problem.[36,37] The second one is decomposing multi-class into a series of binary SVMs, including one-versus-all[5] and one-versus-one.[38] In practice, the most popular methods are one-versus-all (reference LIBLINEAR [39]) and one-versus-one (reference LibSVM [40]) and are due to their simplicity. The one-versus-all strategy (as illustrated in Figure 2) builds c different binary SVM models where the ith one separates the ith class from the rest. The one-versus-one strategy (as illustrated in Figure 3) 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.

This is the first algorithm we use in our approach (latent-lSVM) the second one (LDA) is decribed in the following section.

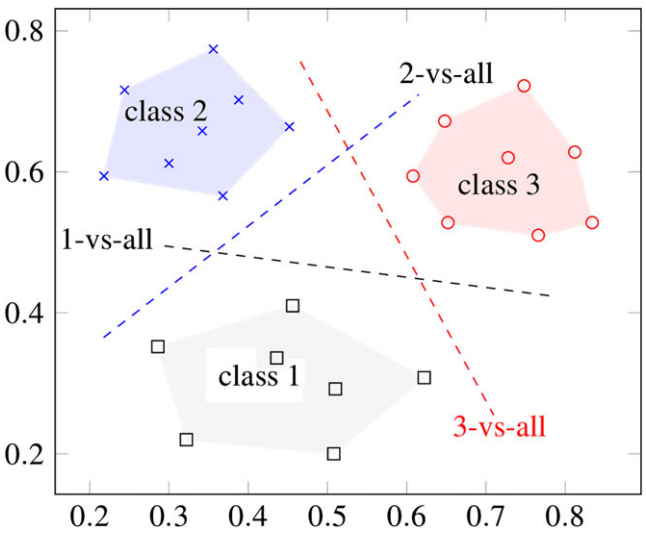

FIGURE 2 Multi-class support vector machine (one-versus-all)

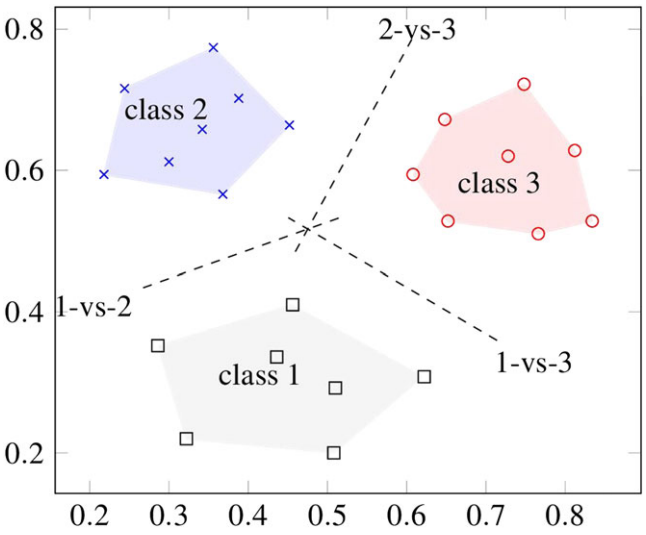

FIGURE 3 Multi-class support vector machine (one-versus-one)

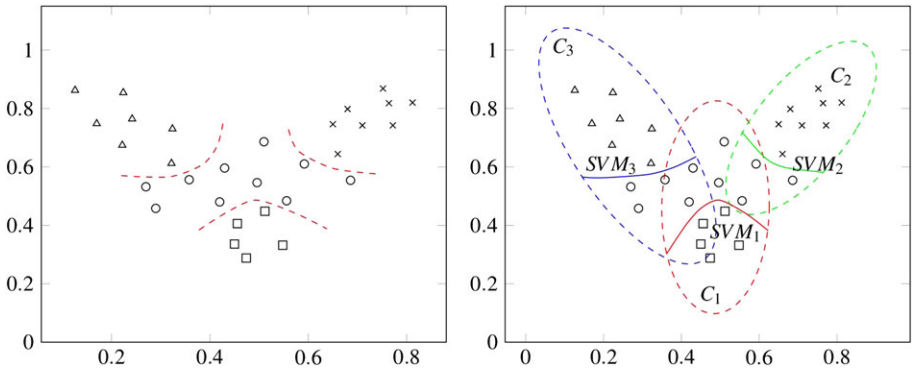

FIGURE 4 Global support vector machine (SVM) model (left part) versus local SVM models (right part)

3 | LATENT LOCAL SVM

In recent applications like the classification of texts/images, the emergence of large text benchmark LSHTC,[12] Book dataset, ImageNet datasets,[21,26] and Fingerprint datasets[27] poses more challenges in training SVM models. The popular BoW [4] used for the representation of texts/images leads to datasets with a very large number of dimensions (eg, 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 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 nonlinear SVM in each cluster to classify the data locally. Figure 4 shows the comparison between a global SVM model (left part) learnt from the full dataset and three local SVM models (right part) learnt from the subsets, using a nonlinear RBF kernel function with $\gamma = 10^2$ and a positive constant $C = 10^5$.

3.1 | Latent Dirichlet allocation

Since the standard BoW model is used to represent texts/images, the training dataset is sparse (very few non-null values) and very high-dimensional input space (many thousands of words). The LDA [29] model is well-known as an effective model for dealing with this kind of data. Therefore, we propose to use LDA to split the full training set into k joint subsets.

The graphical LDA model as illustrated in Figure 5 is a three-level hierarhical Bayesian model, in which each document (ie, text/image) d of a collection D is modeled as a mixture θ_d of k latent topics (ie, clusters), each topic t is a multinomial distribution ϕ_t over words. This means that ϕ_t states out which words are important in topic t and θ_d informs which topics appear in document d. The topic mixture θ_d of document d is a probability distribution drawn from a Dirichlet prior with parameter α. The distribution ϕ_t of words for mixture topic t is drawn from a Dirichlet prior with parameter β. For the ith word $w_{d,i}$ in document d, a topic assignment $z_{d,i} = t$ is drawn from θ_d and then $w_{d,i}$ is generated from ϕ_t.

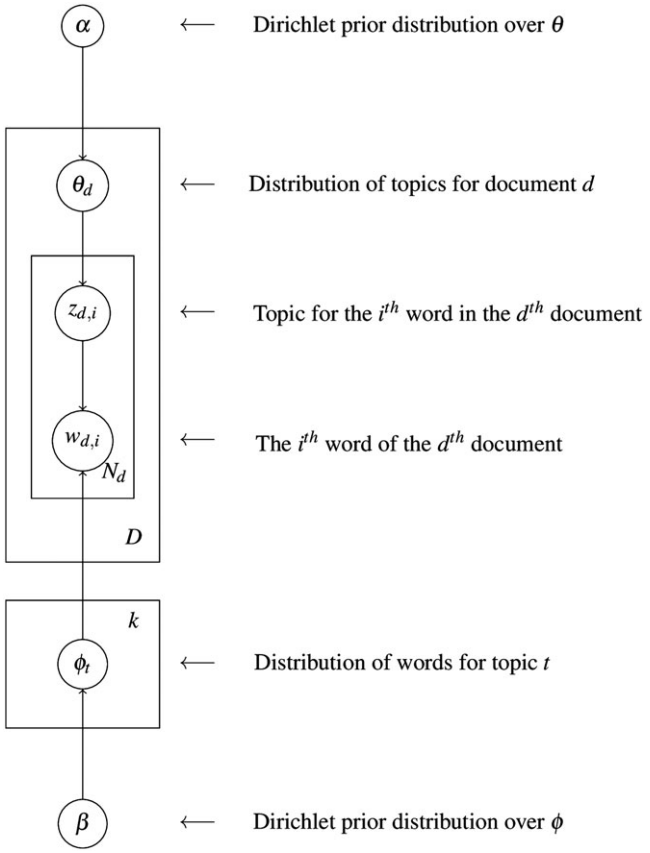

FIGURE 5 Graphical model representation of latent Dirichlet allocation

Given the N observed words w, the LDA inference is to compute the posterior distribution over the latent topic assignments z, the topic mixtures θ of documents, the topics ϕ. Unfortunately, computing this posterior distribution exactly is intractable.[29] The collapsed Gibbs sampling algorithm is one of the most popular method to perform approximate inference for LDA in practice.[41,42] In Gibbs sampling procedure, the mixtures θ and topics ϕ are marginalized out, it needs only sampling the latent topic assignments z. Given the vocabulary size W, the number of times the word w assigned to the topic t (denoted by $N_{w,t}$), the number of times a word in document d assigned to topic t (denoted by $N_{d,t}$), and the number of times topic t assigned to words in document collection D (denoted by N_t); the conditional probability of $z_{d,i}$ is

$$\begin{aligned} p\left(z_{d,i}=t|z^{\overline{d,i}},\alpha,\beta\right) &\propto p\left(w_{d,i}|z_{d,i}=t,z^{\overline{d,i}},\beta\right)p\left(z_{d,i}=t|z^{\overline{d,i}},\alpha\right) \\ &= \frac{N_{w,t}^{\overline{d,i}}+\beta}{N_{t}^{\overline{d,i}}+W\beta}\left(N_{d,t}^{\overline{d,i}}+\alpha\right), \end{aligned} \tag{3}$$

where the superscript $\overline{d,i}$ means that the count takes no account of word i in document d.

An iteration of Gibbs sampling draws a sample for $z_{d,i}$ according to Equation 3 for each word i in each document d and then the counts $N_{d,t}$, N_t, $N_{w,t}$ are updated. After enough iterations, the sampler converges to the target distribution, and then given z, the estimates for θ and ϕ are computed.

Tuning hyper-parameters α and β. While using LDA to partition the full training set into k joint subsets, it must be noted that the Dirichlet priors α and β significantly influence the behavior of the LDA model. α represents document-topic density. With a higher value of α, documents are made up of more topics, and with a lower value of α, documents contain fewer topics. β represents topic-word density. With a high value of β, topics are made up of most of the words in the document collection, and with a lower value of β, they consist of fewer words. In experimental studies,[43] Griffiths and Steyvers proposed for $\beta = 0.01$ and $\alpha = \frac{50}{k}$ with the number of topics k.

We have described the two algorithms we use in our approach (latent-lSVM), let us see now, how we combine them to learn latent local SVM model for effectively handling very-high-dimensional input spaces and large-scale multi-class image datasets.

3.2 | Learning latent local SVM models

The learning algorithm latent-lSVM is described in Figure 6 and Algorithm 1. The first step of the training process is to learn a LDA [29] 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

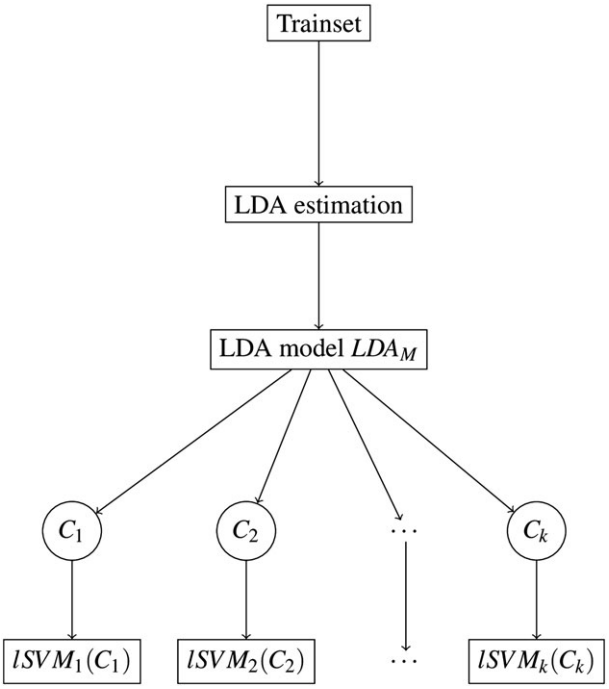

FIGURE 6 Training algorithm of latent local support vector machine (SVM) models. LDA, latent Dirichlet allocation

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 4. 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 $Latent\text{-}lSVM\text{-}model = \{LDA_M, lSVM_1, lSVM_2, \ldots, lSVM_k\}$.

Since the collapsed Gibbs sampling algorithm used for LDA training, Porteous et al[42] and Liu et al[44] proposed the parallel implementation of Gibbs sampling procedure. It means that the first step of Algorithm 1 learns in parallel LDA_M to partition full training set D into k joint subsets denoted by $C_1, C_2, \ldots, C_k$. Furthermore, the second step of Algorithm 1 needs training independently k local models from k clusters $C_1, C_2, \ldots, C_k$. It is a nice property for parallelizing this task. Therefore, the simplest development of the parallel Latent-lSVM algorithm is based on the shared memory multiprocessing programming model OpenMP [45] on multi-core computers. And then, the parallel Latent-lSVM can speed-up the training task linearly scaling with the number of threads.

Algorithm 1: Latent local support vector machines algorithm

input :
- training dataset D
- number of local models k
- Dirichlet priors α and β
- hyper-parameter of RBF kernel function γ
- positive constant for tuning margin and errors of SVMs C

output:
- k local support vector machines models

```
1  begin
2  |  /*LDA performs the data clustering on dataset D;*/
3  |  training model LDA_M in parallel to split full training set D into k joint subsets denoted by C_1, C_2, ..., C_k
4  |  #pragma omp parallel for
5  |  for i ← 1 to k do
6  |  |  /*learning local support vector machine model from C_i;*/
7  |  |  lSVM_i = svm(C_i, γ, C)
8  |  end
9  |  return Latent-lSVM-model = {lSVM_1), lSVM_2, ..., lSVM_k)}
10 end
```

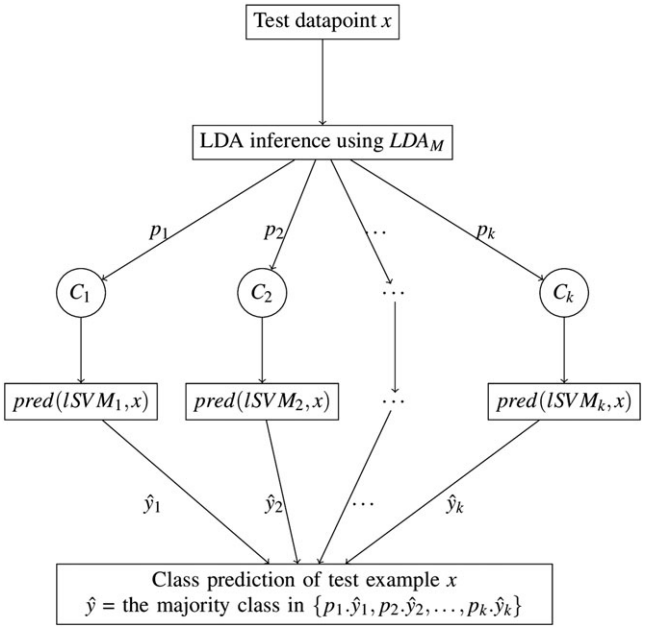

FIGURE 7 Prediction of x with latent local support vector machine (SVM) models. LDA, latent Dirichlet allocation

3.3 | Prediction in latent local SVM models

The prediction of the class for a new datapoint x is decribed in Figure 7. 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 $\hat{y}_1 = pred(lSVM_1, x), \hat{y}_2 = pred(lSVM_2, x), \ldots, \hat{y}_k = pred(lSVM_k, x)$. Finally, the datapoint x is predicted in the class $\hat{y}$ with the largest vote among the prediction classes $\{p_1.\hat{y}_1, p_2.\hat{y}_2, \ldots, p_k.\hat{y}_k\}$ (ie, the sum of the p's for the same class response over all clusters).

$$\hat{y} = \text{the majority class in} \{p_1.\hat{y}_1, p_2.\hat{y}_2, \ldots, p_k.\hat{y}_k\} \tag{4}$$

3.4 | Performance analysis

Let us examine the performance of local SVM models with the parallel Latent-lSVM algorithm.

Algorithmic complexity. We consider the two-class classification problem. The study in Platt [32] illustrated that the computational complexity of the SVM algorithm (with respect to the solution of the quadratic programming (1)) is at least the square of the number of training datapoints. It means that the algorithmic complexity of training a global SVM model is $O(m^2)$ (where m is the number of training datapoints).

Our latent-lSVM algorithm learns k local SVM models in the parallel way on multi-core computer. The full dataset with m individuals is partitioned into k clusters (the cluster size is about $\frac{m}{k}$). The computational complexity of the parallel LDA learning on P-core processor [46] is $O(\frac{m}{P})$. The training complexity of a local SVM is $O((\frac{m}{k})^2)$. Therefore, the algorithmic complexity of the parallel training k local SVM models on a P-core processor is $O(\frac{k}{P}(\frac{m}{k})^2) = O(\frac{m^2}{kP})$.

This complexity analysis illustrates that parallel learning k local SVM models in the Latent-lSVM algorithm is kP times faster than building a global SVM model (the complexity is at least $O(m^2)$).

For the multi-class problem (c classes, $c \geqslant 3$), the one-versus-all global SVM (eg, LIBLINEAR [39]) learns c binary SVM models, and the one-versus-one global SVM (eg, LibSVM [40]) performs $c(c-1)/2$ binary SVM models. Furthermore, the clustering step in the Latent-lSVM algorithm allows reducing the number of classes c_{red} in the cluster, compared to the full training set. It means that $c_{red} = \frac{c}{\alpha}$ where $1 \leqslant \alpha \leqslant c$. The local multi-class SVM training is α or α^2 times faster than the global one using one-versus-all or one-versus-one strategies, respectively.

Generalization capacity. Turn back to Theorem 1 proposed by Vapnik. [5]

Theorem 1. *(Vapnik[5p.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:*

$$EP_{error} \leqslant E\left\{min\left(\frac{sv}{m}, \frac{1}{m}\left[\frac{R^2}{\Delta}\right], \frac{n}{m}\right)\right\}. \tag{5}$$

Theorem 1 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-ISVM, the full dataset with m datapoints is partitioned into k clusters (the cluster size is about $m_k = \frac{m}{k}$*). 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*

$$EP_{error} \leqslant E\left\{min\left(\frac{sv_k}{m_k}, \frac{1}{m_k}\left[\frac{R_k^2}{\Delta_k}\right], \frac{n}{m_k}\right)\right\}. \tag{6}$$

Without loss of generality, we consider a binary classification problem because two most popular methods, one-versus-all and 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 2.

Theorem 2. *Given a training 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* $\Delta_{X_k} \geqslant \Delta_X$ *holds.*

Proof. We remark that the maximal margin Δ_X hyperplane w_{global} 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 Figure 8). 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 w_k for X_k is larger or equal than the maximal margin Δ_X one for fullset X. □

Theorem 3. *Given a training 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 model of local SVMs learnt by Latent-ISVM described in Algorithm 1 on the training dataset can guarantee the classification performance compared to the global SVM one.*

Proof. The classification performance of a local SVM model in the latent-ISVM is studied in term $\frac{1}{m_k}\left[\frac{R_k^2}{\Delta_k}\right]$ in Equation 6. 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, ie, $R_k \leqslant R$ and $m_k \leqslant m$) and the generalized capacity (the margin size, ie, consequence of Theorem $2\Delta_{X_k} \geqslant \Delta_X$). □

Role of the parameter *k*. According to the performance analysis in terms of the algorithmic complexity and the generalization capacity, it illustrates that the parameter k is used in the latent-ISVM to give a trade-off between the generalization capacity and the computational complexity. This can be understood as follows:

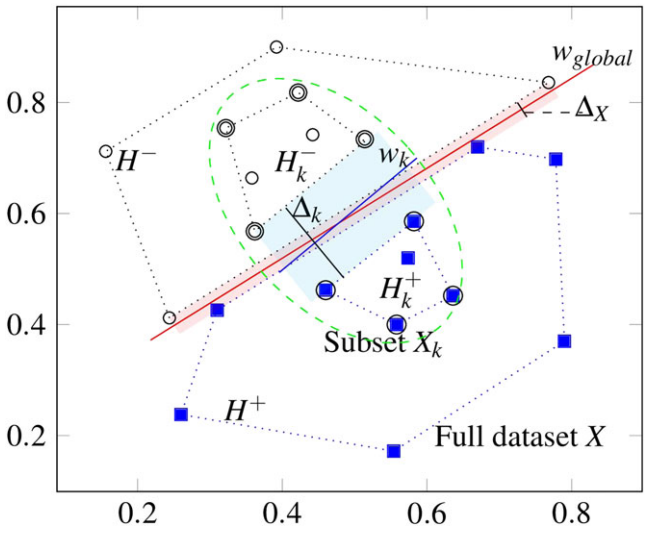

FIGURE 8 The comparison of the maximal margin hyperplane of the global support vector machine (Δ_X) learnt from the full dataset (X) and the local one (Δ_k) learnt from the subset (X_k)

- If k is large, then the latent-ISVM algorithm reduces significant training time (the speed up factor of local SVMs over the global one is k). And then, the size of a cluster is small; the locality is extreme with a very low generalization capacity (respect to $\frac{1}{m_k}\left[\frac{R_k^2}{\Delta_k}\right]$). When $k \to m$, the latent-ISVM algorithm is closed to the 1 nearest neighbor classifier.
- If k is small, then the latent-ISVM algorithm reduces insignificant training time. However, the size of a cluster is large; it improves the generalization capacity. When $k \to 1$, the Latent-ISVM algorithm is approximated the global one.

It leads to set k so that the cluster size is large enough (eg, 200 proposed by Bottou and Vapnik[47]).

4 | EVALUATION

We are interested in the classification performance (accuracy and training time) of our proposal (the latent-ISVM algorithm) for classifying very high-dimensional and large-scale multi-class datasets. Therefore, we here report the comparison of the classification performance obtained by Latent-ISVM and the best state-of-the-art algorithms, including SVM,[5] AdaBoost of J48 (AdaBoost-J48 [30]), and random forests (RF-CART [31]).

4.1 | Software programs

In order to evaluate the effectiveness in classification tasks, we have implemented latent-ISVM in C/C++, OpenMP [45], using the parallel latent Dirichl et allocation program (PLDA+ [44]) and the highly efficient PmSVM[23] with one-versus-all strategy for multi-class).

PmSVM replaces the kernel function $K\langle x_i, x_j\rangle$ in Equations 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 (eg, χ^2 kernel, histogram intersection kernel or Hellinger's kernel):

$$M_p\langle x_i, x_j\rangle = \sum_{z=1}^{n} (x_{i,z}^p + x_{j,z}^p)^{\frac{1}{p}}, \tag{7}$$

where $p \in R$ is a constant.

PmSVM also uses the coordinate descent method [48] 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-ISVM implementation pursues the interesting goals of the complexity reduction (low computational cost) and without the need of parameter tuning.

We also use the highly efficient standard SVM algorithm LibSVM [40] with one-versus-one strategy for multi-class. The rest (AdaBoost-J48, RF-CART) are implemented in Weka library. [49] All experiments are run on a PC with Linux Fedora 20, Intel(R) Core i7-4790 CPU, 3.6 GHz, 4 cores and 32 GB main memory.

4.2 | Classification of fingerprint image datasets

We do setup experiment with seven real fingerprint datasets from our previous research [27] for comparative studies. Fingerprints acquisition was done with Microsoft Fingerprint Reader (optical fingerprint scanner, resolution: 512 DPI, image size: 355×390, 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 Mikolajczyk and Schmid.[50] k-means algorithm [51] is used to group the descriptors into 5000 clusters.* The datasets are described in Table 1. The evaluation protocol is illustrated in the last column of Table 1. The datasets are already divided into training set *Trn* and testing set *Tst*(approximately, respectively, 2/3 and 1/3 of the full dataset). We used the training data to build and tune the parameters of classification models. Then, we classify the testing set using the resulting models.

Tuning parameters

We propose to use RBF kernel type in SVM models because it is general and efficient. [52] Cross-validation protocol (twofold) is used 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 cost C is chosen in $\{10^0, 10^1, 10^2, 10^3, 10^4, 10^5, 10^6\}$, and the hyper-parameter γ of RBF kernel is tried among $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^1\}$. And then, 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 nonterminal node splitting, $min\_obj = 2$ for early stopping).

*The number of clusters/visual words was optimized between 500 and over 10 000; 5000 clusters are the optimum in this experiment.[27]

TABLE 1 Description of fingerprint image datasets

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 2 Parameters of LDA used in latent-ISVM for fingerprint image datasets

ID	Dataset	#Clusters (k)	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

Abbreviations: LDA, latent Dirichlet allocation; SVM, support vector machine.

TABLE 3 Classification correctness (%) on fingerprint image datasets

		Classification Accuracy(%)			
ID	Dataset	AdaBoost-J48	RF-CART	LibSVM	Latent-ISVM
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**

Latent-ISVM requires to tune the Dirichlet hyper-parameters of LDA. According to heuristical method proposed by Griffiths and Steyvers,[43] the good model quality has been reported for $\beta = 0.01$ and $\alpha = \frac{50}{k}$ with the number of topics (clusters) k. Furthermore, Wallach et al[53] illustrates that selecting the number of topics k is one of the most important modeling choices. Latent-ISVM uses $\beta = 0.01$ and *alpha*, k, so that each cluster has about 500 individuals. The idea gives a trade-off between the generalization capacity [54] and the computational cost. Table 2 presents the hyper-parameters of Latent-ISVM used in the classification.

Classification accuracy

The classification results in terms of correctness are given in Table 3 (the best ones are in bold) and Figure 9.

Latent-ISVM 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-ISVM 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-ISVM on LibSVM is statistically significant, in so far as according to the signed rank test, the P value of the observed results (8

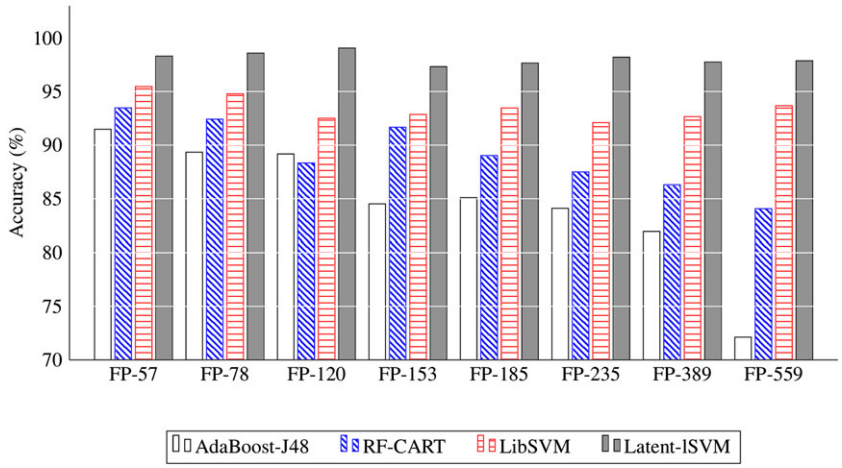

FIGURE 9 Classification correctness (%) on fingerprint image datasets

TABLE 4 Classification correctness (%) on Book dataset

Algorithm	Classification Accuracy(%)
AdaBoost-J48	50.61
RF-CART	65.36
LibSVM	69.48
Latent-lSVM	**70.14**

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.

4.3 | Classification of Book dataset

Book dataset is the real book collection at the Learning Resource Center of Can Tho University in Vietnam. It consists of 114 998 books in a quadruplet format $<$ *Title*; *Abstract*; *Keywords*; *Subject* $>$. The aim is to automatically assign the subject to the book based on the information $<$ *Title*; *Abstract*; *Keywords* $>$. Due to the book information in Vietnamese and in English, there are not only one-syllable words but also multiple syllables ones. We use JvnTextPro [55] well-known as a good Vietnamese word segmentation, to perform the word splitting. The dictionary has 89 821 vocabulary words. The representation of books in the BoW model brings out the table with 114 998 rows and 89 821 columns. Furthermore, there are 661 subjects. Therefore, it yields huge classification challenges of very high-dimensional and large-scale multi-class dataset.[†] The dataset is randomly divided into training set with 100 000 rows and testing set with 14 998 rows. The training set is used to build the classification model and tune the parameters. Then, the classification results are reported on the testing set using the resulting models.

Tuning parameters

The classification models of AdaBoost-J48 and RF-CART consist of 200 trees. The training algorithm RF-CART uses $p' = 3000$ random dimensions for nonterminal node splitting, $min\_obj = 2$ for early stopping.

The algorithm latent-lSVM partitions the training set into $k = 50$ joint clusters using the Dirichlet prior parameters $\beta = 0.01$ and $alpha = 10$.

The optimal parameters of SVM using the RBF kernel with $\gamma = 0.00001, C = 1000000$ are chosen by the grid search with cross-validation protocol (as described above in the classification of fingerprint images).

Classification accuracy

We obtained classification accuracies on Book dataset presented in Table 4 and Figure 10. Once again, the results show that Latent-lSVM achieves the highest correctness with 70.14%. LibSVM and RF-CART are ranked second and third with 69.48% and 65.36%, respectively. Adaboost-J48 is not suited for classifying Book dataset (very high-dimensional and large-scale multi-class) with the 50.61% accuracy. Latent-lSVM has a impressive superiority against Adaboost-J48 (approximately 20%).

[†]Dataset available to Researchers only upon request by email (dtnghi@cit.ctu.edu.vn).

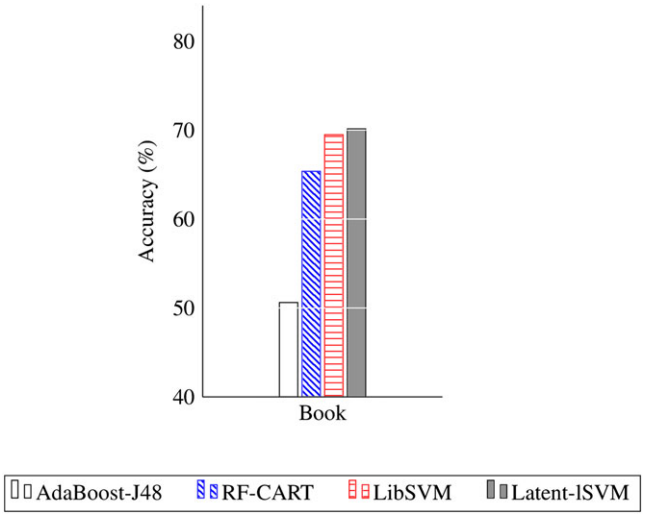

FIGURE 10 Classification correctness (%) on Book dataset

TABLE 5 Training time (min) for Book dataset

Algorithm	Training time, min
LibSVM	489.21
Latent-lSVM (1 thread)	21.50
Latent-lSVM (2 threads)	14.85
Latent-lSVM (4 threads)	11.13
Latent-lSVM (8 threads)	11.15

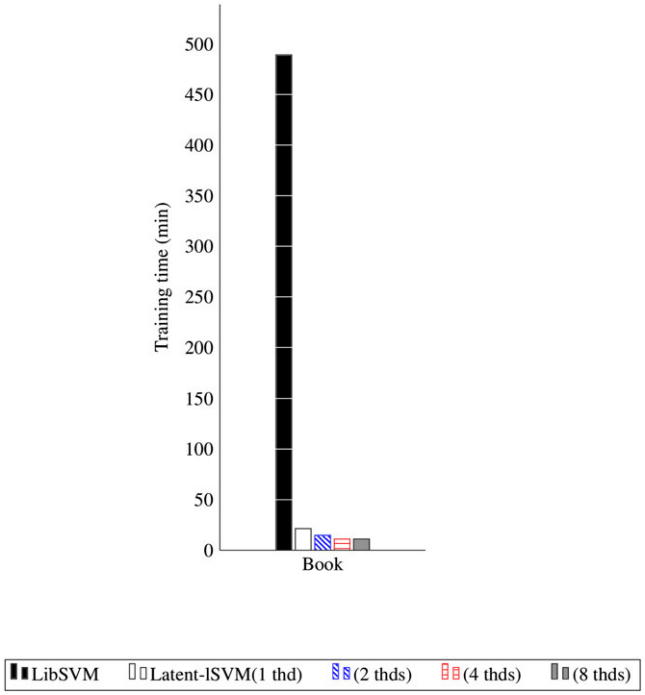

FIGURE 11 Training time (minute) for Book dataset. SVM, support vector machine

4.4 | Training time

Given the differences in implementation, including the programming language used (C/C++ versus Java), the parallelization in multi-core, a comparison of computational time is not really fair. Therefore, we only report here the training time of our latent-lSVM against LibSVM for Book dataset. Due to the PC (Intel(R) Core i7-4790 CPU, 4 cores) used in the experimental setup, we try to vary the number of OpenMP threads (1, 2, 4, and 8 threads) for all training tasks of latent-lSVM.

The training time of LibSVM and latent-ISVM for Book dataset presented in Table 5 and Figure 11 shows that the latent-ISVM using 1 thread is 22.75 times faster than LibSVM. The training time of Latent-ISVM is improved while increasing the number of threads. However, with the restriction of our computer (4 cores), there is no speedup in training process of Latent-ISVM when we set the number of threads over 4. Finally, the latent-ISVM using 4 threads is 1.93 and 43.95 times faster than the one using 1 thread, LibSVM, respectively.

5 | 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 Jacobs et al [56] proposed to use the expectation-maximization clustering algorithm [57] for partitioning the training set into *k* joint clusters (the expectation-maximization clustering algorithm makes a soft assignment based on the posterior probabilities [58]); for each cluster, an NN is learnt to classify the individuals in the cluster. The mixture of SVMs algorithms proposed by Kwok,[59] Collobert et al,[60] and Fu and Robles-Kelly [61] construct local SVM models instead of NN ones in Jacobs et al.[56] CSVM [62] uses *k*-means algorithm [51] to partition the full dataset into *k* disjoint clusters; then, the algorithm learns weighted local linear SVMs from clusters. More recent *k*SVM [63], *kr*SVM [64] (random ensemble of *k*SVM), and *t*SVM [65] propose to parallely train the local nonlinear SVMs instead of weighting linear ones of CSVM. DTSVM [66,67] uses the decision tree algorithm [68,69] 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 (*k*NN) of a new testing individual. First local learning algorithm of Bottou and Vapnik [47] find *k*NN of a test individual; train an NN 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 Vincent and Bengio.[70] More recent local SVM algorithms aim to use the different methods for *k*NN retrieval, including SVM-*k*NN [71] trying with different metrics, ALH [72] using weighted distance and features, and FaLK-SVM [73] speeding up the *k*NN retrieval with the cover tree.[74]

A theorical analysis for such local algorithms discussed in Vapnik[75] 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 parameter to control generalization capacity against locality of local learning algorithms.

Incremental SVM learning methods [76–80] improve memory performance for massive datasets by updating solutions in a growing training set without needing to load the entire dataset into memory at once.

Evolving fuzzy rule-based classifiers proposed by Angelov et al[81,82] use the various architectures (single model and multi-model) and learning mechanisms for dealing with data streams in online mode. Pratama et al[83] and Lughofer et al[84] propose on-line adaptation, evolution, and incremental active learning strategies to develop these evolving fuzzy-rule-based classifiers for stream data. Evolving fuzzy rule-based classifiers are applied in fault detection and diagnosis [85] and activity recognition.[86]

6 | CONCLUSION AND FUTURE WORKS

In order to deal with the increasing amount of data electronically stored, we have presented a novel machine learning algorithm, latent-ISVM, that achieves high performances for classifying very high-dimensional and large-scale multi-class datasets in multi-core computers. Experimental results are led with two kinds of datasets: image classification and text classification. Latent-ISVM algorithm uses LDA to group, in a first step, text/images into clusters to reduce the number of data-points and the number of training classes. Then the Latent-ISVM learns the PmSVM model for each cluster, in a parallel way, to nonlinearly classify the data locally. The experimental results on seven real datasets of fingerprint images showed that latent-ISVM algorithm is very efficient in comparison with RF-CART, AdaBoost of J48, and libSVM (the average improvement goes from 4.65% to 13.37%). Latent-ISVM achieves an accuracy of 97.87% in the classification of fingerprint dataset having 5000 dimensions into 559 classes. The accuracy is improved too on text classification datasets compared to other state of the art algorithms. This means that our algorithm performs better than others, concerning accuracy, for large scale and high-dimensional datasets with large number of classes. Furthermore, it performs the classification in a very rapidly way due to the parallelization of the learning task in multi-core computers and does not require high memory capacity. It can be adapted to the available memory of the computer used.

In the near future, we intend to develop a distributed implementation for large scale processing on an in-memory cluster-computing platform, Apache Spark [87] (running times or up to 100× faster than Hadoop MapReduce, or 10× faster on disk). A promising future research aims at automatically tuning the hyper-parameters of Latent-ISVM. We would like to provide more empirical test on large scale benchmarks like ImageNet datasets[21,26] and comparisons with other large scale linear SVM solvers.[24,39,88]

REFERENCES

1. Fabrizio S. Machine learning in automated text categorization. *ACM Comput Surv*. 2002;34:1-47.
2. Manning CD, Raghavan P, Schütze H. *Introduction to Information Retrieval*. 1st ed.: Cambridge University Press; July 2008.

3. Sivic J, Zisserman A. Video Google: A text retrieval approach to object matching in videos. In: Paper presented at: 9Th IEEE International Conference on Computer Vision (ICCV 2003); October 14/17, 2003; Nice, France:1470-1477.
4. Salton G, Wong A, Yang CS. A vector space model for automatic indexing. *Communications of the ACM*. 1975;18(18):613-620.
5. Vapnik V. *The Nature of Statistical Learning Theory*. 2nd ed.: Springer-Verlag; 2000.
6. Lewis DD, Gale WA. A sequential algorithm for training text classifiers. In: Paper presented at: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '94. New York, NY, USA: Springer-Verlag New York, Inc.; 1994;3-12.
7. Dumais S, Platt J, Heckerman D, Sahami M. Inductive learning algorithms and representations for text categorization. In: Paper presented at: Proceedings of the Seventh International Conference on Information and Knowledge Management. CIKM '98. ACM; 1998; New York, NY, USA:148-155.
8. Joachims T. Text categorization with support vector machines: Learning with many relevant features. In: Nédellec C, Rouveirol C, eds. *Machine Learning: ECML-98. Number 1398 in Lecture Notes in Computer Science*: Springer Berlin Heidelberg; January 1998:137-142.
9. Mitchell T. 20 newsgroups. https://archive.ics.uci.edu/ml/machine-learning-databases/20newsgroups-mld/; 1999.
10. Lewis D. Reuters-21578 text classification test collection. http://www.david-dlewis.com/resources/testcollections/reuters21578/; 1997.
11. Lewis D, Yang Y, Rose T, Li F. RCV1: A new benchmark collection for text categorization research. *J Mach Learn Res*. 2004;5:361-397.
12. Partalas I, Kosmopoulos A, Baskiotis N, et al. LSHTC: A benchmark for large-scale text classification. *CoRR abs/1503.08581*. 2015.
13. Li F, Perona P. A Bayesian Hierarchical Model for Learning Natural Scene Categories. In: Paper presented at: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005); June 20/26, 2005; San Diego, CA, USA:524-531.
14. Lowe D. Object recognition from local scale invariant features. In: Paper presented at: Proceedings of the 7th International Conference on Computer Vision; 1999:1150-1157.
15. Lowe D. Distinctive image features from scale invariant keypoints. *Int J Comput Vis*. 2004:91-110.
16. Hofmann T. Probabilistic Latent Semantic Indexing. In: Paper presented at: Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM; 1999; New York, NY, USA:50-57.
17. Bosch A, Zisserman A, Munoz X. Scene classification via pLSA. In: Paper presented at: Proceedings of the European Conference on Computer Vision; 2006:517-530.
18. Deselaers T, Pimenidis L, Ney H. Bag-of-visual-words models for adult image classification and filtering. In: Paper presented at: Proceeding of The 19th International Conference on Pattern Recognition; 2008:1-4.
19. Benzécri J. *L'analyse Des Correspondances*. Paris: Dunod; 1973.
20. Pham N, 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, Vol. 2; 2008:475-486.
21. Deng J, Berg AC, Li K, Li F. What Does Classifying More than 10, 000 Image Categories Tell Us? In: Paper presented at: Computer Vision - ECCV 2010 - 11Th European Conference on Computer Vision; 2010; Heraklion, Crete, Greece:71-84. Proceedings, Part V.
22. Do T. Detection of pornographic images using bag-of-visual-words and arcx4 of random multinomial naive bayes. In: Paper presented at: Proceedings of the 4th Intl Conference on Theories and Applications of Computer Science; 2011:13-24.
23. Wu J. Power mean svm for large scale visual classification. In: Paper presented at: IEEE Computer Society Conference on Computer Vision and Pattern Recognition; 2012:2344-2351.
24. Do T. Parallel multiclass stochastic gradient descent algorithms for classifying million images with very-high-dimensional signatures into thousands classes. *Vietnam J Comput Sci*. 2014;1(2):107-115.
25. Doan T, Do T, Poulet F. Large scale classifiers for visual classification tasks. *Multimedia Tools Appl*. 2015;74(4):1199-1224.
26. Deng J, Dong W, Socher R, Li L, Li K, Li F. Imagenet: A large-scale hierarchical image database. In: Paper presented at: 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009); June 20/25, 2009; Miami, Florida, USA:248-255.
27. Do T, Lenca P, Lallich S. Classifying many-class high-dimensional fingerprint datasets using random forest of oblique decision trees. *Vietnam J Comput Sci*. 2015;2(1):3-12.
28. Do TN, Poulet F. Classifying very high-dimensional and large-scale multi-class image datasets with Latent-lSVM. In: Paper presented at: CBDCom'2016, Intl IEEE Conference on Cloud and Big Data Computing; 2016:714-721.
29. Blei DM, Ng AY, Jordan MI. Latent dirichl et allocation. *J Mach Learn Res*. 2003;3:993-1022.
30. Freund Y, Schapire R. A decision-theoretic generalization of on-line learning and an application to boosting. In: Paper presented at: Computational Learning Theory: Proceedings of the Second European Conference; 1995:23-37.
31. Breiman L. Random forests. *Mach Learn*. 2001;45(1):5-32.
32. Platt J. Fast training of support vector machines using sequential minimal optimization. In: Schö lkopf B, Burges C, Smola A, eds. *Advances in Kernel Methods Support Vector Learning*; 1999:185-208.
33. Cristianini N, Shawe-Taylor J. *An Introduction to Support Vector Machines: And Other Kernel-based Learning Methods*. New York, NY USA: Cambridge University Press; 2000.
34. Guyon I. Web page on svm applications. http://www.clopinet.com/isabelle/Projects/SVM/app-list.html; 1999.
35. Cernuda C, Lughofer E, Klein H, Forster C, Pawliczek M, Brandstetter M. Improved quantification of important beer quality parameters based on nonlinear calibration methods applied to ft-mir spectra. *Analytical and Bioanalytical Chemistry*. 2017;409(3):841-857.
36. Weston J, Watkins C. Support vector machines for multi-class pattern recognition. In: Paper presented at: Proceedings of the Seventh European Symposium on Artificial Neural Networks; 1999:219-224.
37. Guermeur Y. Svm multiclasses théorie et applications; 2007.
38. Kreßel U. Pairwise classification and support vector machines. *Advances in Kernel Methods: Support Vector Learning*. 1999:255-268.
39. Fan RE, Chang KW, Hsieh CJ, Wang XR, Lin CJ. LIBLINEAR: A library for large linear classification. *J Mach Learn Res*. 2008;9(4):1871-1874.
40. Chang CC, Lin CJ. LIBSVM: A library for support vector machines. *ACM Trans Intell Syst Technol*. 2011;2(27):1-27.

41. Griffiths T. *Gibbs Sampling in the Generative Model of Latent Dirichlet Allocation*: Technical report, Stanford University; 2002.
42. Porteous I, Newman D, Ihler A, Asuncion A, Smyth P, Welling M. Fast collapsed gibbs sampling for latent dirichlet allocation. In: Paper presented at: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '08. ACM; 2008:569-577.
43. Griffiths TL, Steyvers M. Finding scientific topics. *Proc Natl Acad Sci*. 2004;101(suppl 1):5228-5235.
44. Liu Z, Zhang Y, Chang E. Y, Sun M. Plda+: Parallel latent dirichl et allocation with data placement and pipeline processing. *ACM Trans Intell Syst Technol*. 2011;2(3):26:1-26:18.
45. OpenMP Architecture Review Board: OpenMP application program interface version 3.0; 2008.
46. Ihler AT, Newman D. Understanding errors in approximate distributed latent dirichlet allocation. *IEEE Trans Knowl Data Eng*. 2012;24(5):952-960.
47. Bottou L, Vapnik V. Local learning algorithms. *Neural Comput*. 1992;4(6):888-900.
48. Yuan GX, Ho CH, Lin CJ. Recent advances of large-scale linear classification. *Proc IEEE*. 2012;100(9):2584-2603.
49. Witten I, Frank E. *Data mining: Practical Machine Learning Tools and Techniques*: Morgan Kaufmann; 2005.
50. Mikolajczyk K, Schmid C. Scale and affine invariant interest point detectors. *Int J Comput Vis*. 2004;60(1):63-86.
51. MacQueen J. Some methods for classification and analysis of multivariate observations. In: Paper presented at: Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, Vol. 1. University of California Press; January 1967; Berkeley:281-297.
52. Lin C. A practical guide to support vector classification; 2003.
53. Wallach HM, Mimno DM, McCallum A. Rethinking LDA: Why Priors Matter. In: Paper presented at: 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; 2009; Vancouver, British Columbia, Canada:1973-1981.
54. Vapnik V, Bottou L. Local algorithms for pattern recognition and dependencies estimation. *Neural Comput*. 1993;5(6):893-909.
55. Nguyen CT, Phan XH, Nguyen TT. JVnTextPro: A java-based vietnamese text processing tool. http://jvntextpro.sourceforge.net; 2010.
56. Jacobs RA, Jordan MI, Nowlan SJ, Hinton GE. Adaptive mixtures of local experts. *Neural Comput*. 1991;3(1):79-87.
57. Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the em algorithm. *J R Stat Soc*. 1977;39(1):1-38.
58. Bishop CM. *Pattern recognition and machine learning*. New York: Springer-Verlag; 2006.
59. Kwok JTY. Support vector mixture for classification and regression problems. In: Paper presented at: Proceedings of the Fourteenth International Conference on Pattern Recognition, Vol. 1; 1998:255-258.
60. Collobert R, Bengio S, Bengio Y. A parallel mixture of SVMs for very large scale problems. *Neural Comput*. 2002;14(5):1105-1114.
61. Fu Z, Robles-Kelly A. On mixtures of linear svms for nonlinear classification. In: Paper presented at: Structural, Syntactic, and Statistical Pattern Recognition, Joint IAPR International Workshop, SSPR & SPR 2008; December 4/6, 2008; Orlando, USA:489-499. Proceedings.
62. Gu Q, Han J. Clustered support vector machines. In: Paper presented at: Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2013; 2013; Scottsdale, AZ, USA, April 29 - May 1, 2013:307-315. Volume 31 of JMLR Proceedings.
63. Do T. Non-linear classification of massive datasets with a parallel algorithm of local support vector machines. In: Paper presented at: Advanced Computational Methods for Knowledge Engineering. Springer International Publishing; 2015:231-241.
64. Do T, Poulet F. Random local svms for classifying large datasets. In: Paper presented at: Future Data and Security Engineering - Second International Conference, FDSE 2015. Springer; 2015; Ho Chi Minh City, Vietnam, November 23-25, 2015:3-15. Proceedings. Volume 9446 of Lecture Notes in Computer Science.
65. Do T, Poulet F. Parallel learning of local SVM algorithms for classifying large datasets. *T. Large-Scale Data- and Knowledge-Centered Systems*. 2016;31:67-93.
66. Chang F, Guo CY, Lin XR, Lu CJ. Tree decomposition for large-scale SVM problems. *J Mach Learn Res*. 2010;11:2935-2972.
67. Chang F, Liu C. C. Decision Tree as an Accelerator for Support Vector Machines. In: Ding X, ed. *Advances in Character Recognition*: InTech; 2012.
68. Breiman L, Friedman JH, Olshen RA, Stone C. *Classification and Regression Trees*: Wadsworth International; 1984.
69. Quinlan JR. *C4.5: Programs for machine learning*. Morgan Kaufmann, San Mateo, CA; 1993.
70. Vincent P, Bengio Y. K-local hyperplane and convex distance nearest neighbor algorithms. In: Paper presented at: Advances in Neural Information Processing Systems. The MIT Press; 2001:985-992.
71. Zhang H, Berg A, Maire M, Malik J. SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In: Paper presented at: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2; 2006:2126-2136.
72. Yang T, Kecman V. Adaptive local hyperplane classification. *Neurocomputing*. 2008;71(1315):3001-3004.
73. Segata N, Blanzieri E. Fast and scalable local kernel machines. *J Mach Lear Res*. 2010;11:1883-1926.
74. Beygelzimer A, Kakade S, Langford J. Cover trees for nearest neighbor. In: Paper presented at: Proceedings of the 23rd international conference on Machine learning. ACM; 2006:97-104.
75. Vapnik V. Principles of risk minimization for learning theory. In: Paper presented at: Advances in Neural Information Processing Systems 4; 1991; [NIPS Conference, Denver, Colorado, USA, December 2-5, 1991]:831-838.
76. Syed N, Liu H, Sung K. Incremental learning with support vector machines. In: Paper presented at: Proc. of the ACM SIGKDD Intl. Conf. on KDD. ACM; 1999.
77. Cauwenberghs G, Poggio T. Incremental and decremental support vector machine learning. In: Leen TK, Dietterich TG, Tresp V, eds. *Advances in Neural Information Processing Systems 13*: MIT Press; 2001:409-415.
78. Do T, Nguyen VH. A novel speed-up SVM algorithm for massive classification tasks. In: Paper presented at: 2008 IEEE International Conference on Research, Innovation and Vision for the Future in Computing & Communication Technologies, RIVF 2008; 2008; Ho Chi Minh City, Vietnam, 13-17 July 2008:215-220.
79. Doan T, Do T, Poulet F. Parallel incremental power mean SVM for the classification of large-scale image datasets. *IJMIR*. 2014;3(2):89-96.

80. Do T, Tran-Nguyen M. Incremental Parallel Support Vector Machines for Classifying Large-Scale Multi-Class Image Datasets. In: Paper presented at: Future Data and Security Engineering - Third International Conference, FDSE 2016. Proceedings, Springer; 2016; Can Tho City, Vietnam, November 23-25, 2016:20-39.
81. Angelov PP, Lughofer E, Zhou X. Evolving fuzzy classifiers using different model architectures. *Fuzzy Set Syst*. 2008;159(23):3160-3182.
82. Angelov PP, Zhou X. Evolving fuzzy-rule-based classifiers from data streams. *IEEE Trans Fuzzy Syst*. 2008;16(6):1462-1475.
83. Pratama M, Anavatti SG, Er MJ, Lughofer E. An effective classifier for streaming examples. *IEEE Trans Fuzzy Syst*. 2015;23(2):369-386.
84. Lughofer E, Weigl E, Heidl W, Eitzinger C, Radauer T. Integrating new classes on the fly in evolving fuzzy classifier designs and their application in visual inspection. *Appl Soft Comput*. 2015;35:558-582.
85. Lemos AP, Caminhas WM, Gomide F. Adaptive fault detection and diagnosis using an evolving fuzzy classifier. *Inf Sci*. 2013;220:64-85.
86. Ordóñez FJ, de Toledo P, Sanchis A. Activity recognition using hybrid generative/discriminative models on home environments using binary sensors. *Sensors*. 2013;13(5):5460-5477.
87. Zaharia M, Chowdhury M, Franklin MJ, Shenker S, Stoica I. Spark: Cluster computing with working sets. In: Paper presented at: Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing. HotCloud'10. USENIX Association; 2010:10-10.
88. Poulet F, Pham NK. High dimensional image categorization. In: Cao L, Feng Y, Zhong J, eds. *Advanced Data Mining and Applications: 6th International Conference, ADMA 2010*. Chongqing, China, November 19-21, 2010; 2010:465-476. Proceedings, Part I, Berlin, Heidelberg, Springer Berlin Heidelberg.

How to cite this article: Do T-N, Poulet F. Latent-lSVM classification of very high-dimensional and large-scale multi-class datasets. *Concurrency Computat: Pract Exper*. 2017;e4224. https://doi.org/10.1002/cpe.4224