

Fast Polynomial Kernel Classification for Massive Data

Jinshan Zeng, Minrun Wu, Shao-Bo Lin, and Ding-Xuan Zhou

Abstract—In the era of big data, it is highly desired to develop efficient machine learning algorithms to tackle massive data challenges such as storage bottleneck, algorithmic scalability, and interpretability. In this paper, we develop a novel efficient classification algorithm, called fast polynomial kernel classification (FPC), to conquer the scalability and storage challenges. Our main tools are a suitable selected feature mapping based on polynomial kernels and an alternating direction method of multipliers (ADMM) algorithm for a related non-smooth convex optimization problem. Fast learning rates as well as feasibility verifications including the convergence of ADMM and the selection of center points are established to justify theoretical behaviors of FPC. Our theoretical assertions are verified by a series of simulations and real data applications. The numerical results demonstrate that FPC significantly reduces the computational burden and storage memory of the existing learning schemes such as support vector machines and boosting, without sacrificing their generalization abilities much.

Index Terms—Learning theory, classification, support vector machine, polynomial kernel, ADMM.

I. INTRODUCTION

With the rapid development of data mining, massive data abound around our lives in terms of medical records, high-frequency financial data, internet data, network data, longitudinal data, image data and so on. For examples, millions of Internet URLs are inspected for detection of pop-up junk messages; hundreds of millions of financial records are exploited for predicting financial trends; and billions of customer activities are gathered for making marketing decisions. These massive data make the prediction much more precise and bring opportunities to discover subtle information which cannot be captured by data of small size. However, they simultaneously produce a series of scientific challenges such as storage bottleneck, algorithmic scalability, and interpretability [56]. In the machine learning community, developing scalable learning algorithms with theoretical verifications to conquer the massive data challenges is a recent focus and has triggered enormous research activities [4], [56].

For large-scale regression tasks, some scalable learning schemes including localized kernel ridge regression [27], distributed learning [53], learning with sub-sampling [14], have

J. Zeng and M. Wu are with the School of Computer and Information Engineering, Jiangxi Normal University, Nanchang, China (email: jsh.zeng@gmail.com, wuminrun36@gmail.com).

S.B. Lin is with the School of Management, Xi'an Jiaotong University, Xi'an, China (email: sblin1983@gmail.com)

D.X. Zhou is with the School of Data Science and Department of Mathematics, City University of Hong Kong, Kowloon, Hongkong (email: mazhou@cityu.edu.hk)

The corresponding author is Shao-Bo Lin

been proposed to tackle massive data. All these schemes are rigorously justified to significantly reduce the computational burden of classical learning schemes such as kernel methods [42] and neural networks [17] without sacrificing their generalization performances very much. In fact, it has been proved in [27], [21], [23], [33] that these scalable schemes can achieve the optimal learning rates for kernel approaches in the framework of statistical learning theory [6].

From regression to classification, the least-square fitting schemes are frequently replaced by the margin theory [42]. As a consequence, loss functions are changed from least-squares to margin-based functions such as the hinge loss for support vector machine (SVM) [42], logistic loss for logistic regression [28], and exponential loss for boosting [12]. As a result, the proposed approaches for regression are no more efficient for massive data classification. Based on the maximal margin principle, there have been several scalable algorithms for classification, including distributed SVM [11], localized SVM [8], sequential minimal optimization (SMO) [31] and classification with sub-sampling [5]. Although these classification schemes can reduce the computational complexity of SVM and perform well in some special classification tasks, their performance is sensitive to involved parameters and thus, they generally require delicate parameter-selection strategies, which usually brings huge computations in the training process. Furthermore, most of them lack theoretical verifications on the generalization ability, which hinders practitioners' spirits to use them in massive data classifications.

The aim of the present paper is to propose a novel scalable learning algorithm with theoretical verifications for massive data classification. Different from the maximal margin principle, our basic idea is to select a suitable feature space in which all linear classifiers perform similarly. For this purpose, we use polynomial kernels to build up the feature mapping and control the capacity of feature space via tuning the kernel parameter. Furthermore, since the margin is not so important in our approach and can be removed, our method then turns to solving a non-smooth convex optimization problem. We adopt an alternating direction method of multipliers (ADMM) to solve this problem and then propose a novel learning algorithm, named as fast polynomial kernel classification (FPC) to tackle massive data. Two important advantages of FPC are: (a) since the margin constraint (or regularization parameter) in SVM is not required in FPC and capacity of feature space is determined by the kernel parameter, there is only one parameter to be tuned in FPC; (b) For each fixed kernel parameter, the computational complexity of FPC is much smaller than that of SVM. Both advantages show

that FPC can significantly reduce the computational burden of SVM and thus succeeds in tackling massive data.

The prominent performance of FPC is verified by both theoretical analysis and numerical experiments. Theoretically, we present some feasibility guarantees for FPC including the expressivity of the feature mapping and convergence of the ADMM algorithm. Furthermore, we rigorously prove that under the Tsybakov noise [46] and geometric noise [38] assumptions, FPC achieves the existing optimal learning rates for SVM in the framework of statistical learning theory [6], [39]. Experimentally, numerical studies including toy simulations, UCI standard data experiments, massive data trials and a real world cat-and-dot image classification experiment are conducted to illustrate the outperformance of FPC. Our numerical results show that FPC is more efficient than some state-of-the-art methods in the sense that it can significantly reduce the computational burden and storage requirements without degrading the generalization capability much.

The rest of this paper is organized as follows. In Section II, we present the motivation of our study. Section III proposes the FPC algorithm as well as some feasibility verifications. Section IV derives the generalization error estimates for FPC in the framework of learning theory. Section V provides a series of toy simulations to verify the feasibility of FPC. Section VI conducts a series of UCI data sets and real applications to show the effectiveness of FPC in massive data classification. We conclude this paper in Section VII.

II. MOTIVATION AND ROADMAP

In this section, we aim at presenting motivations of our study and providing a roadmap to conquer the massive data challenge for binary classification.

A. Motivations

The maximal margin principle is an important tool to design learning algorithms for binary classification. A margin-based algorithm explicitly utilizes the margin of each data point to produce an efficient classifier, where the margin of a single data point is defined to be the distance from the data point to a decision boundary. In this way, both SVM [42] and boosting [32] can be regarded as margin-based algorithms.

SVM is a classical and popular method to tackle binary classification problems. The magic behind SVM is a feature mapping and a maximal margin principle. As shown in Figure 1, the former focuses on leveraging a low dimensional input space into a high dimensional feature space by some feature mapping such that the features associated with data are linearly separable. The latter one, as exhibited in Figure 2 (a), produces a unique classification decision by maximizing the margin and then transforms the classification problem into a quadratic programming problem.

Numerous practical applications and theoretical studies [42], [39] verified the power of SVM in classification, provided the size of data is not so large. However, when the size of data continuously increases, SVM is confronted with two crucial design flaws. The first one, as shown in Figure 2 (b), is that the number of features near the separation plane increases

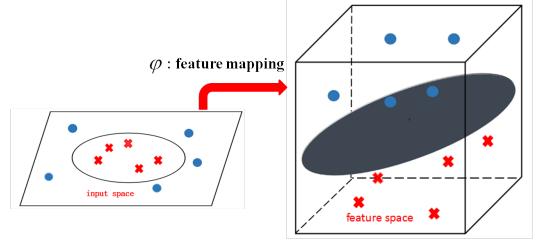
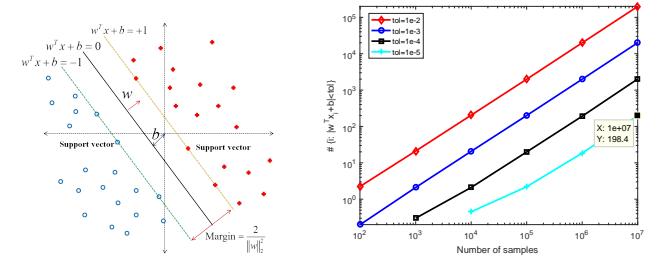


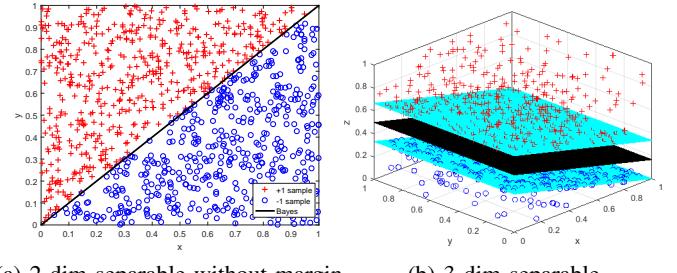
Fig. 1: Feature mapping in SVM



(a) Margin principle for SVM

(b) Margin v.s. data size

Fig. 2: Maximal margin principle and its difficulty in massive data classification



(a) 2-dim separable without margin

(b) 3-dim separable

Fig. 3: Leverage the dimension to guarantee the feasibility of the margin theory in the feature space

almost linearly with respect to the size of data, which makes the margin principle infeasible when the data size is huge. The other one is that it requires at least $\mathcal{O}(m^2)$ memory requirements and usually $\mathcal{O}(m^3)$ computational complexity to solve the quadratic programming problem, where m is the size of data. Both design flaws hinder the use of SVM in tackling massive data classification problems.

Intuitively, there are two approaches to modify SVM to guarantee the availability of the maximal margin principle. The first one is to leverage the dimension of the feature space further with additional feature mapping, just like Figure 3 purports to show. The other one is the data reduction approach that reduces the size of data via selecting a small number of representative samples. However, the former expands the feature space and thus needs more delicate algorithm to find a suitable classifier, which brings additional computational burden, while the performance of the latter depends heavily on the representative samples and usually requires clustering algorithms to find them out, which is also time-consuming for tackling massive data.

In a nutshell, for massive data classification, it is time-

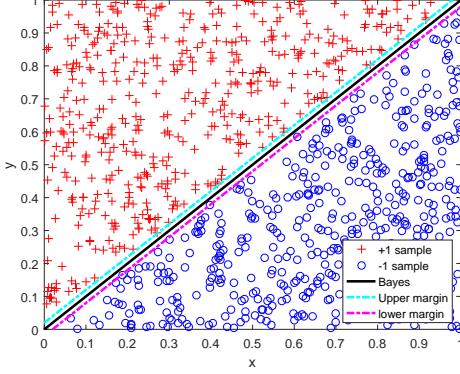


Fig. 4: Margins for massive data

consuming to produce classifiers via the maximal margin principle. In this paper, we drive a totally different direction from the maximal margin principle to design learning algorithms for massive data classification. Our basic idea is to select appropriate feature mappings such that all linear classifiers in the feature space perform similarly. With these feature mappings, the massiveness of data makes the margin of support vectors be extremely small and thus it is not necessary to distinguish the linear classifier from the margin, just as Figure 4 exhibits. With the maximal margin principle, SVM is equivalent to solving a regularized problem [42]

$$f_{D,\lambda} = \arg \min_{f \in \mathcal{H}_K} \left\{ \frac{1}{m} \sum_{i=1}^m (1 - y_i f(x_i))_+ + \lambda \|f\|_K^2 \right\}. \quad (1)$$

where $D = \{(x_i, y_i)\}_{i=1}^m$ is the sample set, $t_+ = \max\{t, 0\}$ for $t \in \mathbb{R}$ and λ is a regularization parameter which is proportional to the margin in Figure 2 (a). However, in our approach, since the maximal margin principle is not considered, we are faced with an un-regularized convex optimization problem

$$f_D = \arg \min_{f \in \mathcal{H}} \left\{ \frac{1}{m} \sum_{i=1}^m (1 - y_i f(x_i))_+ \right\}, \quad (2)$$

where \mathcal{H} is a linear space whose dimension is much smaller than m . Two important ingredients in our approach are to find suitable \mathcal{H} and algorithms to solve (2) such that (2) performs at least as well as (1).

B. Feature mapping with polynomial kernels

In this subsection, we focus on selecting the feature mapping for massive data classification. Our purpose is to equip (2) with a suitable feature space \mathcal{H} such that (2) performs similarly as (1). Like SVM, we constitute the feature mapping with kernels and the problem then boils down to choose kernels and centers for the kernel. As discussed in Subsection II-A, we are highly desired for kernels which determine the dimension of the corresponding feature space directly. The most popular kernel for this purpose is the polynomial kernel $K_s(x, x') = (1 + x \cdot x')^s$, where $s \in \mathbb{N}$ is the tunable parameter referring to the degree of kernel polynomial and $x, x' \in \mathbb{R}^d$. Denote by \mathcal{H}_s the RKHS associated with K_s endowed with the inner product $\langle \cdot, \cdot \rangle_s$ and norm $\|\cdot\|_s$. It is well known [54]

that \mathcal{H}_s is the set of d -variable polynomials of degree at most s , and the dimension of \mathcal{H}_s is $n = \binom{s+d}{d} = \frac{(s+d)!}{d!s!}$.

Denote by \mathcal{P}_s^d the set of algebraic polynomials defined on the input space $X \subset \mathbb{R}^d$ of degree at most s . Setting $\mathcal{H} = \mathcal{P}_s^d$, we are concerned with the empirical risk minimization:

$$f_D = \arg \min_{f \in \mathcal{P}_s^d} \left\{ \frac{1}{m} \sum_{i=1}^m (1 - y_i f(x_i))_+ \right\}. \quad (3)$$

Since the dimension of \mathcal{P}_s^d is $n = \binom{s+d}{s}$, if we select $\{\eta_j\}_{j=1}^n \subset X$ such that $\{(1 + \eta_j \cdot x)^s\}_{j=1}^n$ is a linear independent system, then

$$\mathcal{P}_s^d = \left\{ \sum_{j=1}^n c_j (1 + \eta_j \cdot x)^s : c_j \in \mathbb{R} \right\} =: \mathcal{H}_{\eta,n}. \quad (4)$$

In this way, (3) can be converted to

$$f_{D,n} = \arg \min_{f \in \mathcal{H}_{\eta,n}} \left\{ \frac{1}{m} \sum_{i=1}^m (1 - y_i f(x_i))_+ \right\}. \quad (5)$$

To determine $\{\eta_j\}_{j=1}^n$, we introduce the fundamental system with respect to the polynomial kernel K_s [24] as follows. Let $\zeta := \{\zeta_i\}_{i=1}^n \subset X$. It is called a K_s -fundamental system if

$$\dim \mathcal{H}_{\zeta,n} = \binom{s+d}{s},$$

where $\dim \mathcal{H}$ denote the dimension of the linear space \mathcal{H} . It is easy to see that an arbitrary K_s -fundamental system implies (4). The following proposition [24] reveals that almost every set with $n = \binom{d+s}{s}$ points is a K_s -fundamental system.

Proposition 1. *Let $s, n \in \mathbb{N}$ and $n = \binom{d+s}{s}$. Then the set*

$$\{\zeta = (\zeta_i)_{i=1}^n : \dim \mathcal{H}_{\zeta,n} < n\}$$

has Lebesgue measure 0.

Based on Proposition 1, we can design simple strategies to choose the centers $\{\eta_j\}_{j=1}^n$. In particular, $\{\eta_j\}_{j=1}^n$ can be selected either deterministically on X or randomly independently and identically (i.i.d.) according to the uniform distribution, since the uniform distribution is continuous with respect to the Lebesgue measure. In summary, there is only a discrete parameter in (5) which reduces the difficulty of model selection. Furthermore, since $n = \binom{s+d}{d}$ and s is frequently not larger than 10, n is usually much smaller than m , especially when d is not so large. Thus, the capacity of the feature space is usually small, reducing the difficulty for algorithm selection.

C. ADMM for non-smooth convex optimization

In this subsection, we try to exploit the alternating direction method of multipliers (ADMM) [13] to solve the un-regularized optimization problem (5). Let $A \in \mathbb{R}^{m \times n}$ with $A_{ij} = (1 + x_i \cdot \eta_j)^s$. To solve (5), it suffices to find a solution to the following nonsmooth convex optimization problem:

$$\min_{u \in \mathbb{R}^n} \frac{1}{m} \sum_{i=1}^m \left(1 - y_i \sum_{j=1}^n A_{ij} u_j \right)_+. \quad (6)$$

Due to the nonsmoothness, the well-known gradient descent method is not available to the optimization problem (6). Concerning the sub-gradient methods, [30] showed that $\mathcal{O}(1/\varepsilon)$ iterations are required and $\mathcal{O}(mn)$ float computations are needed in each iteration, where ε is the approximation accuracy between the estimator generated by a sub-gradient method and the global minimum of (6). Faced with massive data, ε should be extremely small and thus sub-gradient methods involve extremely high computational burden. From the optimization viewpoint, we can also develop some algorithms to (6) based on its dual form. In the following proposition whose proof will be given in *Appendix E*, we present the dual form of (6).

Proposition 2. *The dual problem of (6) is a linear programming shown as follows:*

$$\begin{aligned} & \max_{\mathbf{a}, \mathbf{c} \in \mathbb{R}^m} \mathbf{1}_m^T \mathbf{a} \\ & \text{s.t. } \mathbf{a} + \mathbf{c} = \frac{1}{m} \mathbf{1}_m, \quad A^T \text{Diag}(y) \mathbf{a} = 0, \end{aligned} \quad (7)$$

where $\mathbf{1}_m$ is the all one vector of dimension m , $\text{Diag}(y)$ is a diagonal matrix with $y = (y_1, \dots, y_m)^T$ being its diagonal vector.

From Proposition 2, the dual problem of the suggested learning scheme (6) is a linear programming, which is different from that of the classical SVM, i.e., the quadratic programming. The interior point algorithm [50] is a well known algorithm which converges to the global minimum of liner programming problems. The problem is, however, the interior point algorithm requires the computation of some Hessian matrices, which also needs huge computational and storage complexities if m is large.

In a word, both sub-gradient methods and dual methods lack scalability in tackling massive data. Alternatively, we turn to ADMM, another type of powerful optimization methods that can handle the nonsmooth convex problem (6). We firstly reformulate the unconstrained problem (6) as the following equivalent constrained optimization problem via introducing another variable v ,

$$\min_{u \in \mathbb{R}^n, v \in \mathbb{R}^m} f(v) \quad \text{s.t.} \quad Au - v = 0, \quad (8)$$

where $f(v) := \frac{1}{m} \sum_{i=1}^m (1 - y_i v_i)_+$. The augmented Lagrangian function of (8) is defined by

$$\mathcal{L}_\beta(u, v, w) = f(v) + \langle w, Au - v \rangle + \frac{\beta}{2} \|Au - v\|_2^2, \quad (9)$$

where $w \in \mathbb{R}^m$ is a multiplier variable, $\beta > 0$ is the augmented Lagrangian parameter. Based on \mathcal{L}_β , the ADMM algorithm for problem (6) can be described as follows: given an initialization u^0, v^0, w^0 , parameters $\alpha > 0, \beta > 0$, for $k = 0, 1, \dots$,

$$u^{k+1} = \arg \min_{u \in \mathbb{R}^n} \left\{ \mathcal{L}_\beta(u, v^k, w^k) + \frac{\alpha}{2} \|u - u^k\|_2^2 \right\}, \quad (10)$$

$$v^{k+1} = \arg \min_{v \in \mathbb{R}^m} \mathcal{L}_\beta(u^{k+1}, v, w^k), \quad (11)$$

$$w^{k+1} = w^k + \beta(Au^{k+1} - v^{k+1}). \quad (12)$$

From (10), we adopt the *proximal* update strategy for u^{k+1} instead of the original minimization strategy, mainly due to

the following two reasons: the first one is to overcome the possible ill-conditionedness of matrix $A^T A$ via introducing a proximal term, as shown by (13) in Lemma 1 below, and the second one is to stabilize the optimization procedure such that successive two iterations change relatively smoothly.

III. FAST POLYNOMIAL KERNEL CLASSIFICATION

In this section, we present the feasibility and parameter-selection of the ADMM algorithm for solving the non-smooth convex optimization problem (6). With these helps, we propose a novel algorithm called fast polynomial kernel classification (FPC) for massive data classification.

A. On feasibility and convergence of ADMM

To make ADMM user-friendly, we present closed-form solutions to (10) and (11) such that the sequence u^k, v^k can be updated analytically. The following lemma shows the feasibility to solve the corresponding minimization problems.

Lemma 1. *Let (u^k, v^k, w^k) be the k -th iterate of ADMM. Then the updates (10) and (11) can be expressed analytically as follows*

$$u^{k+1} = (\beta A^T A + \alpha \mathbf{I}_n)^{-1}(\alpha u^k + \beta A^T v^k - A^T w^k) \quad (13)$$

and

$$v^{k+1} = \text{Hinge}_{m\beta}(y, A u^{k+1} + \beta^{-1} w^k), \quad (14)$$

where \mathbf{I}_n is the identity matrix of size n ,

$$\begin{aligned} & \text{Hinge}_\gamma(\xi, \zeta) \\ &= (\text{hinge}_\gamma(\xi(1), \zeta(1)), \dots, \text{hinge}_\gamma(\xi(m), \zeta(m)))^T, \\ & \xi = (\xi(1), \dots, \xi(m))^T \text{ for } \xi \in \mathbb{R}^m, \gamma > 0 \text{ and} \\ & \text{hinge}_\gamma(a, b) = \\ & \begin{cases} b, & \text{if } a = 0, \\ b + \gamma^{-1}a, & \text{if } a \neq 0 \text{ and } ab \leq 1 - \gamma^{-1}a^2, \\ a^{-1}, & \text{if } a \neq 0 \text{ and } 1 - \gamma^{-1}a^2 < ab < 1, \\ b, & \text{if } a \neq 0 \text{ and } ab \geq 1. \end{cases} \end{aligned} \quad (15)$$

The proof of this lemma will be presented in *Appendix D*. Lemma 1 presents the feasibility of ADMM via showing the closed-form update sequences (10), (11) and (12). Besides the feasibility verification of (10), (11) and (12), the update rules bring four types of parameters: α in (10), β in (10), (11), (12), an initial point (u^0, v^0, w^0) and a stopping rule. In the following theorem, we show that the proposed ADMM algorithm can get a global minimum of the non-smooth convex optimization problem (6) and the convergence is independent of the selection of α, β and the initial point.

Theorem 1. *Let $\{p^k := (u^k, v^k, w^k)\}$ be the sequence generated by (10), (11) and (12) for any $\alpha > 0, \beta > 0$ and finite initial point (u^0, v^0, w^0) . Suppose that there exists a solution to problem (6). Then p^k converges to some $p^* = (u^*, v^*, w^*)$ and u^* is a global minimizer of (6).*

The proof of this theorem is motivated by [18, Theorem 6.1] and will be given in *Appendix A*. Theorem 1 shows the global convergence of the ADMM algorithm and presents theoretical

guidance on parameter-selection. In particular, since Theorem 1 holds for any $\alpha, \beta > 0$ and finite initial point (u^0, v^0, w^0) , we can set $\alpha = \beta = 1$ and $(u^0, v^0, w^0) = (0, y, 0)$. In the following theorem, we present some guidance on setting the stopping rule.

Theorem 2. *Under the assumptions of Theorem 1, we have*

$$\|p^{k+1} - p^k\|_H^2 \leq \|p^k - p^{k-1}\|_H^2, \quad \forall k \geq 1, \quad (16)$$

and

$$\|p^{k+1} - p^k\|_H^2 = o(1/k), \quad (17)$$

where

$$H = \begin{pmatrix} \alpha \mathbf{I}_n & 0 & 0 \\ 0 & \beta \mathbf{I}_m & 0 \\ 0 & 0 & \beta^{-1} \mathbf{I}_m \end{pmatrix}, \quad (18)$$

and $\|\xi\|_H^2 = \xi^T H \xi$ for any $\xi \in \mathbb{R}^{2m+n}$.

The proof of Theorem 2 will be presented in Appendix B. Theorem 2 shows that the discrepancy between two successive iterations is monotonically decreasing at the rate of $o(1/k)$ under the metric of matrix norm. This yields an efficient stopping criterion for ADMM, i.e., $\|p^{k+1} - p^k\|_H^2 < tol$ for some small positive constant tol .

B. Fast polynomial kernel classification

Based on the previous analysis, we aim at the non-smooth convex optimization problem (5) and adopt the ADMM update rules (10), (11) and (12) to generate a global minimum $\sum_{j=1}^n u_j^* K_s(\eta_j, \cdot)$ of (5), where $\{\eta_j\}_{j=1}^n$ is a K_s fundamental system. With these, we propose a novel classification algorithm, called fast polynomial kernel classification (FPC), for massive data classification.

Algorithm 1 Fast Polynomial kernel Classification (FPC)

Input: training samples $D := \{(x_i, y_i)\}_{i=1}^m$, the degree $s \in \mathbb{N}$ of polynomial kernel $K_s(x, x') = (1 + x \cdot x')^s$, a K_s fundamental system $\{\eta_j\}_{j=1}^n$ with $n = \binom{s+d}{s}$, $\alpha = \beta = 1$, initialization $(u^0, v^0, w^0) = (0, y, 0)$, and stopping parameter $tol > 0$. Let $A \in \mathbb{R}^{m \times n}$ with $A_{ij} = (1 + x_i \cdot \eta_j)^s$.

Update: For $k = 0, 1, \dots$, update

$$\begin{aligned} u^{k+1} &= (\beta A^T A + \alpha \mathbf{I}_n)^{-1} (\alpha u^k + \beta A^T v^k - A^T w^k), \\ v^{k+1} &= \text{Hinge}_{m\beta}(y, Au^{k+1} + \beta^{-1} w^k), \\ w^{k+1} &= w^k + \beta(Au^{k+1} - v^{k+1}). \end{aligned}$$

End: the first k satisfying $\|p^{k+1} - p^k\|_H^2 < tol$ with H defined by (18).

Output: $f_{D,s}(\cdot) = \sum_{j=1}^n u_j^* K_s(\eta_j, \cdot)$.

As shown in Theorem 1, we can use any α, β and finite initialization (u^0, v^0, w^0) . From the definition of the K_s fundamental system and Proposition 1, $\{\eta_j\}$ can be generated i.i.d. according to the uniform distribution from X directly or from $\{x_i\}_{i=1}^m$, or drawn directly to be the first n points from $\{x_i\}_{i=1}^m$. Our numerical results in Section V exhibit that the selection of these parameters does not affect the performance of FPC very much. For an iterative algorithm, an efficient

stopping criterion is very important, especially when the size of data is huge. On one hand, Theorem 2 shows that the distance between two successive iterations, i.e., $\|p^{k+1} - p^k\|_H^2$, monotonically decreases to 0 at the speed of $o(1/k)$, implying tol the smaller the better. On the other hand, too small tol will result in a large number of iterations and thus requires huge computations. Due to numerous practical trials, we find that $tol = 5 \times 10^{-4}$ is a good choice for massive data classification task. Of course, if the data size is not so large, we can set tol to be extremely small to guarantee the convergence. Therefore, there is only one parameter, the degree of kernel polynomial s , to be tuned in FPC. As shown in Theorem 3 below, the optimal s is less than $(m/\log m)^{\frac{1}{d}}$. If the input space has a relatively large dimension, then the optimal s is generally less than 10, which makes n be very small and thus reduce the computational and storage complexities of SVM. Since s is a discrete value, we use the well known cross-validation [16, Chapter 8] to choose the best s from the set $\{1, 2, \dots, 10\}$ in practice. The FPC algorithm is summarized in Algorithm 1.

Next, we analyze the computational complexity of the proposed FPC. Since $(\beta A^T A + \alpha \mathbf{I}_n)^{-1}$ is applied for each iteration, we calculate and store it in advance, which requires $\mathcal{O}(mn)$ storage complexity and $\mathcal{O}(mn^2 + n^3)$ computational complexity, respectively. Once the inverse matrix is calculated in advance, for each iteration, the computational cost to update u^{k+1} is $\mathcal{O}(mn + n^2)$. By Lemma 1, $\text{Hinge}_{m\beta}$ is an element-wise operator. Thus, as shown by (15), the computational cost of the update of v^{k+1} is $\mathcal{O}(mn)$. It is obvious that the computational cost of the update of w^{k+1} is also $\mathcal{O}(mn)$. Let T be the maximal number of iterations achieving the given stopping criterion. Then the total computational cost of the proposed FPC is $\mathcal{O}(mn^2 + Tmn)$. As shown by our simulations to be presented later, it usually suffices to use very few iterations (about 5 iterations) to achieve the default stopping criterion with $tol = 5 \times 10^{-4}$. Since n and T are generally much less than m , especially for a huge m , the total computational complexity of FPC $\mathcal{O}(mn(T+n))$ is far lower than $\mathcal{O}(m^3)$ required for the classical SVM. This shows that FPC is an efficient algorithm with an approximately linear computational complexity, and a good candidate for handling massive data classification tasks.

C. Related works

The existing variants of SVM for massive data classification can be mainly divided into two categories. The first class is decomposition-based methods that divide the original large scale quadratic programming (QP) into smaller QP subproblems [31], [20]. Among these, the Sequential Minimal Optimization (SMO) algorithm proposed in [31] is a representative one. SMO transforms the large QP problem into a series of small QP problems, each involving only two dual variables according to the violation of the Karush-Kuhn-Tucker (KKT) conditions. The major advantage of SMO is that each QP subproblem can be solved analytically in an efficient way, without a numerical QP solver. Another advantage of SMO is that extra matrix storage is not required for keeping the kernel matrix, since no matrix operations are involved. However,

SMO converges slowly for massive data classification problem [20], mainly due to each iteration only involves two dual variables in the optimization. For example, when the size of training samples is in a million level, then the iteration number of SMO to run ergodically for all training samples is also in a million level, which limits its application in massive data classification. The second class is the data reduction-based methods which reduce the number of training data points via selecting a small number of representative training samples from the large data set [5], [45], [36]. Such a method firstly discards many training samples according to either clustering or sub-sampling schemes, and then implements SVM on the rest training samples. Thus, its effectiveness depends heavily on the quality of selected training samples. As a consequence, the effectiveness of sampling scheme and clustering scheme plays a central role for these methods, while the sampling or clustering scheme generally is associated to the a-prior information of the data distribution. Moreover, discarding amount of training samples may result in waste of data. Although the practical effectiveness of these variants for massive data classification was verified, most of them lack theoretical guarantees on the generalization ability.

In the design flow of FPC, we propose a different direction to avoid the maximal margin theory in classification. Using the special features for polynomial kernel, we propose an ADMM rather than QP to find the classifier. The advantages of FPC are the reduction of the tunable parameters, user-friendly design flow, low computational burden and optimal generalization error guarantee given in the following section.

IV. GENERALIZATION ERROR ANALYSIS

Our analysis is carried out in a standard binary classification framework [39], where the sample $D = \{(x_i, y_i)\}_{i=1}^m$ with $x \in X$ and $y \in Y = \{-1, 1\}$ is assumed to be drawn independently and identically according to some unknown distribution ρ , which admits a decomposition into the marginal distribution ρ_X on X and the conditional distribution $\rho(\cdot|x)$ at each $x \in X$. Binary classification algorithms aim to generate a classifier $\mathcal{C} : X \rightarrow Y$, based on D , to minimize the misclassification error

$$\mathcal{R}(\mathcal{C}) = \mathbf{P}[\mathcal{C}(x) \neq y] = \int_X \mathbf{P}[y \neq \mathcal{C}(x)|x] d\rho_X,$$

where $\mathbf{P}[y|x]$ is the conditional probability at $x \in X$. Theoretically, the Bayes rule

$$f_c(x) = \begin{cases} 1, & \text{if } \eta(x) \geq 1/2, \\ -1, & \text{otherwise} \end{cases}$$

minimizes the misclassification error, where $\eta(x) = \mathbf{P}[y = 1|x]$ is the Bayes decision function. Since f_c is independent of the classifier \mathcal{C} , the performance of the classifier \mathcal{C} can be measured by the excess misclassification error $\mathcal{R}(\mathcal{C}) - \mathcal{R}(f_c)$. Without loss generality, we assume X to be a simplex on \mathbb{R}^d , which is defined by

$$X = \{x \in \mathbb{R}^d : x(i) \geq 0, 1 \leq i \leq d, 1 - |x| \geq 0\},$$

where $x = (x(1), x(2), \dots, x(d)) \in \mathbb{R}^d$, $|x| = \sum_{i=1}^d x(i)$.

To present the generalization error, we should introduce some assumptions on the data. The first one is the Tsybakov noise condition [46] shown as follows.

Definition 1. Let $0 \leq q \leq \infty$. We say that ρ satisfies the Tsybakov noise condition with exponent q if there exists a constant \hat{c}_q such that

$$\rho_X(\{x \in X : |2\eta(x) - 1| \leq \hat{c}_q t\}) \leq t^q, \quad \forall t > 0. \quad (19)$$

The Tsybakov noise condition reflects the close extent of the margin from the hard margin to the soft margin. Such a condition plays an important role in reducing the variance of the relative loss with its expectation and has been adopted in [38], [49], [44], [22] to quantify learning rates of SVM.

Different from the standard smoothness assumption [15], [51], [55], the other condition in this paper is the geometric noise assumption proposed in [38]. Denote by

$$f_\rho(x) := \arg \min_{t \in \mathbb{R}} \int_Y (1 - yt)_+ d\rho(y|x). \quad (20)$$

Write $X_{-1} := \{x \in X : f_\rho(x) < 0\}$, $X_1 := \{x \in X : f_\rho(x) > 0\}$, and $X_0 := \{x \in X : f_\rho(x) = 0\}$. We get a distance function $x \mapsto \tau_x$ by

$$\tau_x := \begin{cases} \text{dist}(x, X_0 \cup X_1), & \text{if } x \in X_{-1}, \\ \text{dist}(x, X_0 \cup X_1), & \text{if } x \in X_1, \\ 0, & \text{otherwise,} \end{cases}$$

where $\text{dist}(x, A)$ denotes the distance of x to a set A with respect to the Euclidean norm. With this function, we can define the following geometric noise condition.

Definition 2. Let $\alpha > 0$. We say that ρ satisfies the geometric noise condition with exponent α if there exists a constant $c > 0$ such that

$$\int_X |f_\rho(x)| \exp\left(-\frac{\tau_x^2}{t}\right) d\rho_X \leq ct^\alpha \quad (21)$$

holds for all $t > 0$.

The geometric noise condition describes the concentration of the measure $|f_\rho(x)| d\rho_X$ near the decision boundary and does not imply any smoothness of f_c or regularity of ρ_X with respect to the Lebesgue measure on X . As shown in [38, Theorem 2.6], if ρ has a Tsybakov noise exponent q and satisfies the envelope condition $|f_\rho(x)| \leq c_\gamma \tau_x^\gamma$ for some constants γ and c_γ , then ρ has a geometric noise exponent $\alpha = \frac{q+1}{2}\gamma$ if $q \geq 1$.

In the following theorem, we derive the generalization error bounds of FPC under assumptions (19) and (21).

Theorem 3. Assume that ρ satisfies noise assumptions (19) and (21) with exponent $q > 0$ and $\alpha > 0$. Let $\theta^* = \frac{q+1}{\alpha(q+2)+d(q+1)}$, $s = \lceil (m/\log m)^{\theta^*} \rceil$, where $[a]$ represents the integer part of $a > 0$. Then for all $0 < \delta < 1$, with confidence at least $1 - \delta$, we have

$$\mathcal{R}(\text{sgn}(f_{D,n})) - \mathcal{R}(f_c) \leq C(m/\log m)^{-\alpha\theta^*} \log\left(\frac{4}{\delta}\right),$$

where C is a positive constant independent of m or δ .

The proof of this theorem will be provided in Appendix C. Studying the generalization performance in the framework

of learning theory is a classical topic in machine learning. In Steinwart's pioneer work [37], the universal consistency of SVM with different kernels were presented. With this, [3] provided generalization error for SVM with q -hinge loss under some smoothness assumptions for the Bayes decision function. For the polynomial kernel, following the work of [54], [43], [44] proved that under the same condition as Theorem 3, SVM with polynomial kernels can reach a learning rate of order $m^{-\frac{\alpha(q+1)}{\alpha(q+2)+(d+1)(q+1)}}$. Specifically, ignoring the logarithmic term in Theorem 3, the exponent of the learning rate of FPC is $-\frac{\alpha(q+1)}{\alpha(q+2)+d(q+1)}$, which is better than that of SVM with polynomial kernels, i.e., $\frac{\alpha(q+1)}{\alpha(q+2)+(d+1)(q+1)}$ in [44]. It is also comparable with SVM with Gaussian kernels in [38]. If $\alpha = \infty$, ignoring the logarithmic term, the learning rate derived in Theorem 3 is of order $m^{-\frac{q+1}{q+2}}$. This rate coincides with the optimal learning rates $m^{-\frac{q+1}{q+2}}$ for certain classifiers based on empirical risk minimization in [46].

V. TOY SIMULATIONS

In this section, we present toy simulations to demonstrate the effect of algorithmic parameters of FPC and verify the developed theoretical assertions. All the numerical experiments are carried out in Matlab R2015b environment running Windows 8, Intel(R) Xeon(R) CPU E5-2667 v3 @ 3.2GHz 3.2GH. The code is available in the web site: <https://github.com/JinshanZeng/FPC>.

A. Experimental setting

The settings of simulations are described as follows.

Samples: In simulations, the training samples were generated as follows. Let

$$h(t) = ((1 - 2t)_+^5 (32t^2 + 10t + 1) + 1) / 2, \quad t \in [0, 1]$$

be a nonlinear Bayes rule. Let $\mathbf{x} = \{x_i\}_{i=1}^m \subset ([0, 1] \times [0, 1])^m$ be drawn i.i.d. according to the uniform distribution with size m . Then we labeled the samples lying in the epigraph of function $h(t)$ as the positive class, while the rest were labeled as the negative class, that is, given an $x_i = (x_i(1), x_i(2))$, its label $y_i = 1$ if $x_i(2) \geq h(x_i(1))$, and otherwise, $y_i = -1$. Moreover, for the training samples, we added $r\%$ noise, that is, we selected $m * r\%$ training samples via a uniformly random way and reversed their labels. The testing samples $\mathbf{x}' = \{x'_i\}_{i=1}^{m'} \subset ([0, 1] \times [0, 1])^{m'}$ were generated according to the same procedure of the training samples without adding noise.

Implementation and Evaluation: We implemented six simulations to verify the theoretical assessments and show the effectiveness of FPC. For each simulation, we repeated $\ell \in \mathbb{N}$ times of experiments and recorded its training and test error, which are defined as the ratios of the number of wrong training (test) labels learned to the training (test) sample size, respectively. The first one is to suggest an efficient stopping criterion of the suggested ADMM algorithm. The second and third ones aim to show the sensitivity of the suggested ADMM to its parameters including the proximal parameter α and Lagrangian parameter β , respectively. The

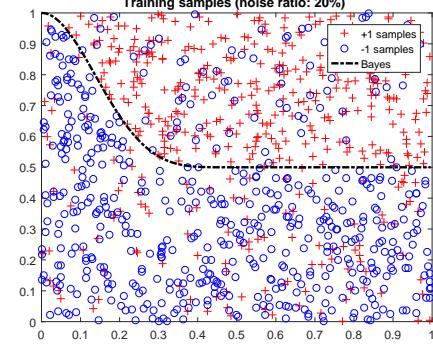


Fig. 5: The generated data used in simulations. The red points are labeled as “+1” class, while the blue points are labeled as “-1” class.

fourth one is to study the role of kernel parameter s in FPC via comparing it with SVM using polynomial kernel (called *SVM-Poly* for short). The fifth one is to suggest some efficient center generation mechanisms in terms of the generalization ability, and the final one is to show the robustness of the proposed learning scheme to different types of noise.

B. Simulation results

In this part, we report the experimental results and present some discussions.

Simulation 1: On stopping criterion. The initialization and stopping criterion are crucial for the practical implementation of an iterative algorithm. In this simulation, we aim to provide an effective initialization as well as a stopping criterion. Specifically, we set $m = m' = 1000$, $\ell = 50$ and $r = 10$. As demonstrated by Theorem 2, the sequence $\|p^{k+1} - p^k\|_H^2$ is monotonically decreasing. We suggest using $\|p^{k+1} - p^k\|_H^2 < \text{Tolerance}$ for some $\text{Tolerance} > 0$ as the stopping criteria. Moreover, we suggest setting the initialization $p^0 = (0, y, 0)$. Under the above settings, in this simulation, we verified 25 *Tolerance*'s in the form of $10^{-\gamma}$, where γ was taken as the equispaced points from 0 to 5 with the stepsize 0.2. The other parameters were set as

$$s = 9, n = \binom{11}{9} = 55, \alpha = 1, \beta = 1.$$

The centers $\{\eta_j\}_{j=1}^n$ were selected as the first n inputs of $\{x_i\}_{i=1}^m$.

The trends of test error, training error, and the required maximal number of iterations are depicted in Figure 6. From Figure 6, the test error of FPC is stable for the choice of tolerance. However, when the tolerance is less than $10^{-3.2}$ ($\approx 6.3 \times 10^{-4}$), the number of iterations required to achieve the given tolerance increases dramatically from 3 to 1734 as the tolerance decreases to 10^{-5} , yet the test error only changes slightly from 0.01235 to 0.01153. Therefore, in terms of both test error and computational cost, we suggest *Tolerance* = 5×10^{-4} as the default stopping criterion of Algorithm 1 in practice.

Simulation 2: On effect of proximal parameter α . In this simulation, we verify the effect of the proximal parameter α of the ADMM algorithm. The settings of this simulation are

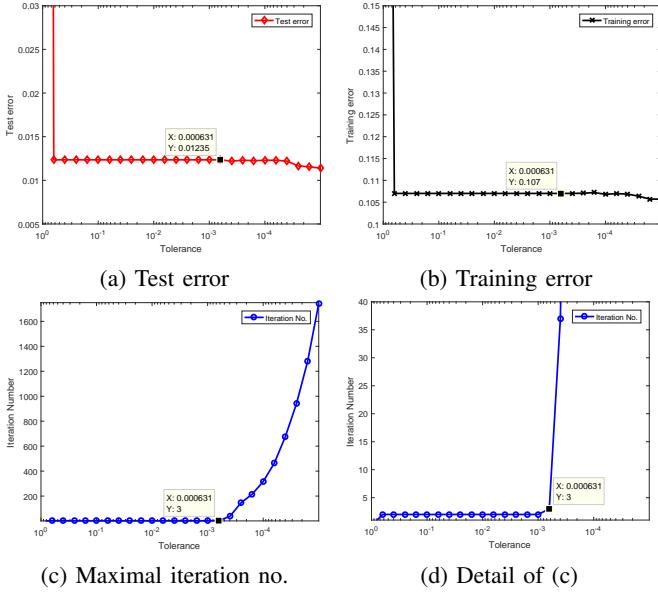


Fig. 6: The effect of stopping criterion for FPC.

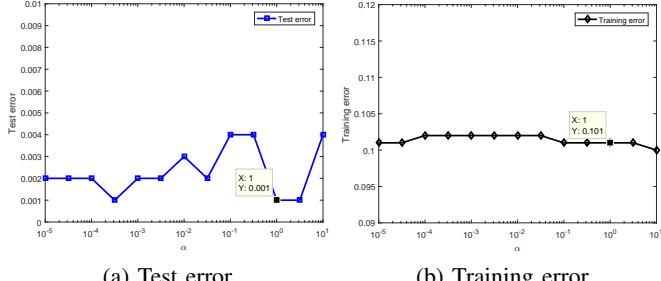


Fig. 7: The effect of parameter α for FPC.

similar to those in **Simulation 1** except α is varied in the form of 10^γ , where γ is taken as the equispaced point of the interval $[-5, 1]$ with the step 0.5, and the stopping criterion is fixed as $Tolerance = 5 \times 10^{-4}$. The trends of test and training errors as the varying of α are presented in Figure 7. From Figure 7, the proposed algorithm is stable for the choice of parameter α . Thus, we suggest using $\alpha = 1$ as the default setting in practice for the computational convenience.

Simulation 3: On effect of parameter β . As shown in Theorem 1, the suggested ADMM algorithm converges for arbitrary positive β . In this simulation, we study the numerical effect of the Lagrangian parameter β of the ADMM algorithm. The settings of this simulation are similar to those in **Simulation 2** except β vary in the form of 10^γ with γ being taken as the equispaced point of the interval $[-2, 2]$ with the step 0.2, while $\alpha = 1$ is fixed. The trends of test and training errors as the varying of β are presented in Figure 8. From Figure 8, the choice of parameter β has a little effect on the performance of FPC in terms of test error, and the numerical performance of FPC is usually stable around 1. Thus, we suggest using $\beta = 1$ as the default setting in practice.

Simulation 4: On effect of kernel parameter s . In this simulation, we studied the importance of the parameter s in SVM and FPC. We set $m = m' = 1000$, $\ell = 50$ and

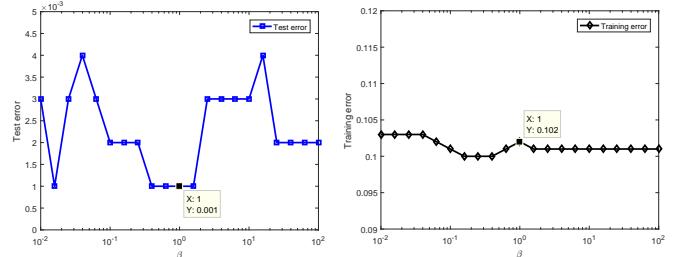


Fig. 8: The effect of parameter β for FPC.

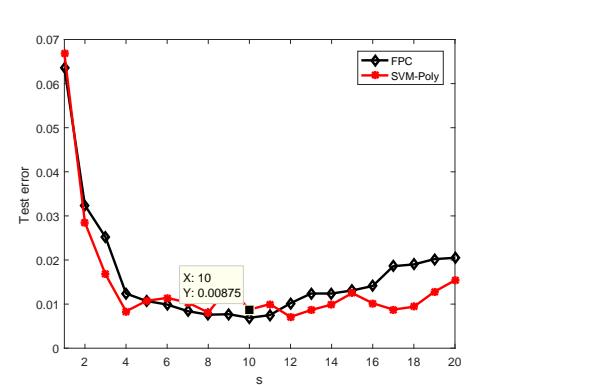


Fig. 9: The effect of s for both SVM-Poly (22) and FPC.

$r = 10$. For comparison, we consider the following SVM with polynomial kernels (SVM-Poly)

$$f_{D,s,\lambda} = \arg \min_{f \in \mathcal{H}_s} \left\{ \frac{1}{m} \sum_{i=1}^m (1 - y_i f(x_i))_+ + \lambda \|f\|_s^2 \right\}. \quad (22)$$

To solve (22), a quadratic programming (QP) of size m should be solved via a numerical optimizer, such as SMO [31] and *libsvm* toolbox used in this paper. The computational complexity of such QP problem generally depends on the extent of ill-conditionedness of its associated coefficient matrix, i.e., $\mathcal{K} + \lambda m \mathbf{I}_m$, where $\mathcal{K} := (K_s(x_i, x_j))_{i,j=1}^m$. In our running, the parameters of ADMM are set as follows: $\alpha = 1$, $\beta = 1$, $Tolerance = 5 \times 10^{-4}$ and $p^0 = (0, y, 0)$. Our aim is to study whether there are additional requirements for s in FPC. For this purpose, we record the test errors of SVM and FPC with s being varied from the range $[1, 20]$. For (22), λ is selected to be optimal for a given s , according to the test samples directly. We take test error as a function of s . The simulation results are reported in Figure 9. By Figure 9, there exist optimal degrees s for SVM-Poly and FPC, both of which are around 10. Moreover, trends of the effect of s for both algorithms are similar, showing that the polynomial kernel parameter s plays almost the same role and removing the regularization parameter in SVM does not bring additional difficulty in selecting s .

Simulation 5: On center generation mechanism. In this simulation, we set m varying from 5×10^3 to 5×10^4 , $\ell = 50$ and $r = 10$. Our aim is to study the effect of different center generation mechanisms for FPC. Specifically, we consider the following three schemes for choosing η in (6). Scheme

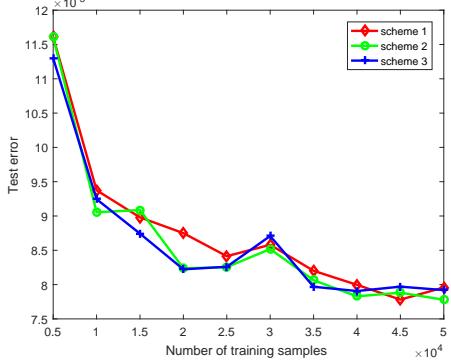


Fig. 10: The effect of selection schemes for centers.

1 denotes that $\eta = \{\eta_i\}_{i=1}^n$ are drawn i.i.d according to the uniform distribution. Scheme 2 denotes that $\{\eta_i\}_{i=1}^n$ are selected as the first n inputs of samples. Scheme 3 denotes that $\{\eta_i\}_{i=1}^n$ are selected randomly as the n input of samples. We figured out the test errors of these three approaches with different sizes of samples (from 5×10^3 to 5×10^4) and optimal s (selected according to the test samples). The parameters of ADMM used in three schemes were the same as those in **Simulation 4**. The experimental results are reported in Figure 10. It can be found in Figure 10 that for a suitable s , the performance of three schemes are almost the same. Thus, in practice, we usually suggest using *Scheme 2* to generate the centers $\{\eta_j\}_{j=1}^n$, due to they are determined directly by n , and thus, there does not need additional storage to store them, once the training samples are given.

Simulation 6: On effect of noise. In this simulation, we considered the effect of three different types of noise: the noisy samples concentrated within a band of the Bayes (see, Figure 11(a)); the noisy samples lying in the region that is far from the Bayes (see, Figure 11(c)); and the noisy samples lying randomly in all the region $[0, 1] \times [0, 1]$ (see, Figure 11(b)). In the following, we considered different levels for each noise type. For noise type 1 and type 2, we consider different widths of the banded region and noise ratio within the banded region. For noise type 3, we only considered different noise levels. For all these three types of noise, the parameters of FPC are set the same as the following: $\alpha = 1$, $\beta = 1$, $Tolerance = 5 \times 10^{-4}$, $p^0 = (0, y, 0)$, $s = 9$, $n = \binom{s+d}{s} = 55$, and the centers $\{\eta_i\}_{i=1}^n$ are set according to *Scheme 2* as suggested in **Simulation 5**. The trends of training and test errors with respect to the noise ratio are depicted in Figure 12 (a)-(f). Particularly, the noise ratios for noise type 1 and type 2 are defined as the ratio of the number of noisy samples to the size of training samples. For noise type 1, the noise level roughly equals to the multiplication of the band width and the specified noise ratio in the considered region, while for noise type 2, the noise level roughly equals to the multiplication of one minus the band width and the specified noise ratio in the considered region.

From Figure 12, the training error of FPC is almost equal to the noise level, which means that FPC preserves these three different types of noise approximately linearly. Thus, FPC is generally robust to these types of noise. By Figure 12 (b), (d)

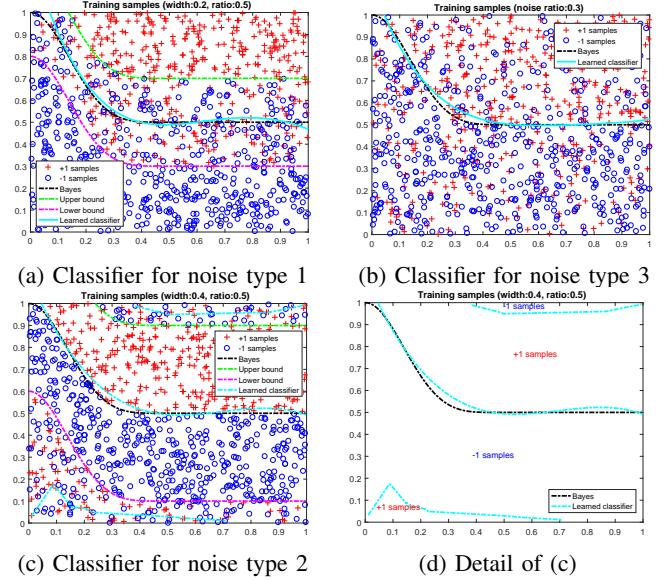


Fig. 11: The classifiers learned by FPC under different types of noise, where the cyan lines are the learned classifiers of FPC, and the black one is the ground truth of Bayes. For the first two types of noise, the **width** is defined as the distance between the boundaries of noise and Bayes, that is, the distance between the magenta and black lines for noise type 1, or the distance between the green and black lines for noise type 2, respectively, and the **ratio** is defined as the ratio of noisy samples in the specified regions, that is, the region nearby the Bayes for noise type 1 and the region away from the Bayes for noise type 2, in (a) and (c) of this figure. The **noise ratio** for the third type of noise is defined as the number of noisy samples with “wrong” labels divided by the total number of training samples.

and (f), on one hand, it is expected that the trend of test error gets worse as the increasing of noise level. On the other hand, given a level of training error (which approximately reflects the noise level), FPC generally performs the best for noise type 3, while similarly for the other two types of noise. Some learned classifiers by FPC for three different types of noise are depicted in Figure 11. From Figure 11(a) and (b), the learned classifiers for the first and third types of noise are still preserved well under some moderate noise levels, as the learned classifiers are nearby the Bayes and keep the similar trends as the Bayes. However, for the second type of noise, the learned classifier is generally divided into three pieces, that is, the main piece still nearby the Bayes, the other two pieces lie in the two outlier regions, as shown in Figure 11. Such phenomenon is reasonable because the second type of noise actually cannot be simply regarded as random noise, but it will be more suitable to be considered as some outliers, which changes the classifiers.

VI. REAL DATA EXPERIMENTS

In this section, we show the effectiveness of the proposed method via a series of experiments on nine UCI data sets covering various areas with medium sizes, and two massive

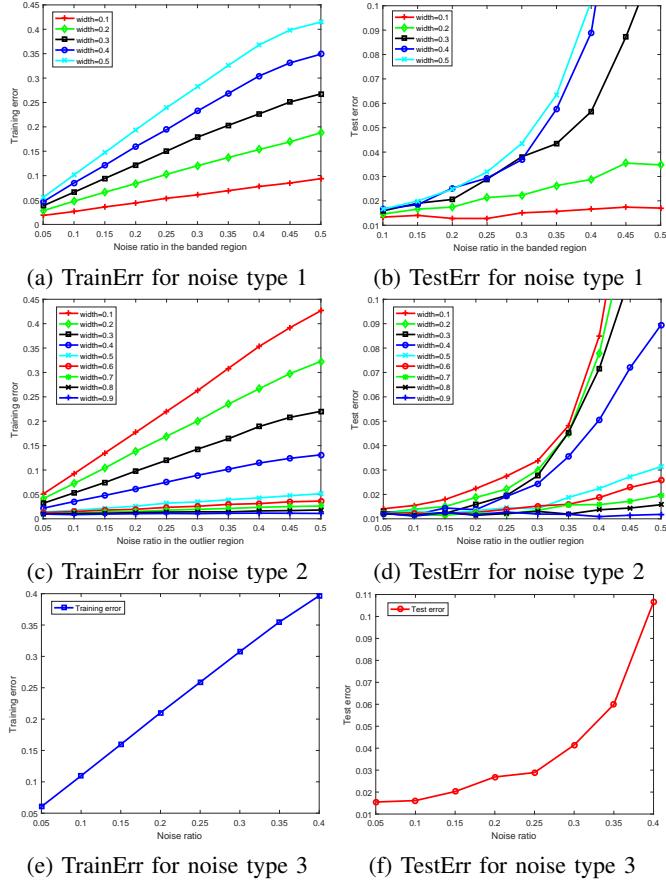


Fig. 12: Performance of FPC for three types of noise with different levels.

data arisen from the exotic particle discovery application in high-energy physics, as well as a real application, i.e., image classification.

A. UCI data sets

1) *Experimental setting:* In the following, we describe the setting of the experiments.

Samples: All data is from: <https://archive.ics.uci.edu/ml/datasets.html>. The sizes of data sets are listed in Table I. For each data set, we used 50%, 25% and 25% samples as the training, validation and testing sets, respectively.

Competitors: We evaluate the effectiveness of FPC via comparing with the baselines and three state-of-the-art methods including two typical support vector machine (SVM) methods with radial basis function (*SVM-RBF*) and polynomial (*SVM-Poly*) kernels and the random forest (*RF*) [2]. We used the well-known *libsvm* toolbox to implement these SVM methods, from the website: <https://www.csie.ntu.edu.tw/~cjlin/libsvm/>.

Implementation: For FPC, we set $\alpha = 1$, $\beta = 1$ and the initialization $p^0 = (0, y, 0)$; the stopping criterion of FPC was set as the maximal iterations less than 5, which is generally adequate as shown in the previous simulations; as shown by Theorem 3, the polynomial degree s is empirically selected from the range $(1, s_{\max})$, where $s_{\max} :=$

TABLE I: Sizes of UCI data sets. In the latter tables, we use the first vocabulary of the name of the data set for short.

Data sets	Data size	#Attributes
breast_cancer	683	9
banknote_authentication	1,372	4
seismic_bumps	2,584	18
musk2	6,598	166
HTRU2	17,898	8
MAGIC_Gamma_Telescope	19,020	10
occupancy	20,560	5
default_of_credit_card_clients	30,000	24
Skin_NonSkin	245,057	3

$\min \left\{ \left\lceil \left(\frac{m}{\log m} \right)^{1/d} \right\rceil, 10 \right\}$; once s is given, the number of centers n is set as $n = \min \left\{ \binom{s+d}{s}, m \right\}$; the centers $\{\eta_i\}_{i=1}^n$ are selected as the first n inputs of training samples.

For both *SVM-RBF* and *SVM-Poly*, the ranges of parameters (c, g) involved in *libsvm* are determined via a grid search on the region $[2^{-5}, 2^5] \times [2^{-5}, 2^5]$ in the logarithmic scale, while for *SVM-Poly*, the kernel parameter s is selected from the interval $[1, 10]$ via a grid search with 10 candidates, i.e., $\{1, 2, \dots, 10\}$.

For *RF*, the number of trees used is determined from the interval $[2, 20]$ via a grid search with 10 candidates, i.e., $\{2, 4, \dots, 20\}$. For each data set, we run 50 times for all algorithms, and then record their averages and standard deviations (std) of test accuracies (*TestAcc*) *, the averages of training time (*TrainTime*) and testing time (*TestTime*), as well as the average of the *sparsity levels* †.

2) *Experimental results:* The experimental results of UCI data are reported in Tables II–IV. As shown in Table II, *TestAcc* of all the mentioned methods are similar in most of data sets except *musk2*. For this data set, the test accuracy of FPC is significantly better than those of competitors and also baselines provided by the source website. Since *TestTime* depends on the sparsity level of the estimator, we also give a comparison of the sparsity level of the mentioned methods in Table IV. In a nutshell, as far as the generalization capability is concerned, all of these methods are of high quality, usually slightly better than the baseline. However, as far as the computational burden is concerned, FPC is superior to others. Furthermore, compared to *SVM-Poly*, FPC can usually deduce more sparse estimators. Note in Table III that the training time of FPC is much less than that of the classical kernel approaches, especially when the size of data exceeds ten thousands. From these experimental results, FPC provides a possibility to tackle massive data.

B. Massive data sets: SUSY and Higgs

The field of high-energy physics is devoted to the study of elementary constituents of matter. By investigating the structure of matter and the laws that govern its interactions, this field strives to discover fundamental

* *TestAcc* is defined as the percentage of the correct classification.

† For SVMs, the sparsity level is the number of the support vectors, for FPC, the sparsity level is the number of selected centers, and for random forest, the sparsity level is the number of selected trees.

TABLE II: Test accuracies and their standard deviations in parentheses. The best results are marked in bold, and the second best results are marked via blue color.

Data sets	SVM-RBF	SVM-Poly	RF	FPC	Baseline
breast	97.19 (0.71)	96.84 (1.05)	96.81 (1.4)	96.78 (0.90)	96.20
banknote	98.07 (0.71)	97.72 (0.99)	98.99 (0.57)	98.15 (0.73)	95.81
seismic	93.84 (0.37)	93.59 (0.74)	92.88 (0.78)	93.68 (0.84)	88.00
musk2	91.11 (0.41)	92.82 (0.32)	96.56 (0.51)	99.08 (0.32)	90.30
HTRU2	97.53 (0.06)	97.42 (0.08)	97.88 (0.18)	97.26 (0.23)	99.00
MAGIC	85.69 (0.26)	86.00 (0.11)	86.90 (0.53)	85.10 (0.53)	86.34
occupancy	98.63 (0.08)	98.95 (0.08)	99.14 (0.10)	98.77 (0.13)	97.16
default	81.60 (1.05)	82.10 (0.32)	81.01 (0.40)	80.51 (0.29)	82.00
Skin	98.80 (0.02)	99.06 (0.01)	99.94 (0.001)	98.83 (0.14)	98.09

TABLE III: Training time in seconds.

Data sets	SVM-RBF	SVM-Poly	RF	FPC
breast	5.42	1.10	0.99	0.003
banknote	8.35	6.07	1.21	0.05
seismic	30.12	12.94	1.86	0.04
musk2	1,350.4	845.89	7.36	6.44
HTRU2	167.29	78.55	8.11	0.51
MAGIC	1,026.9	12,011.5	18.18	0.91
occupancy	660.04	2,991.6	6.57	1.43
default	17,998.9	35,902.2	30.65	0.43
Skin	1,862.2	851.62	64.23	36.64

properties of the physical universe. The primary tools of experimental high-energy physicists are modern accelerators, which collide protons and/or antiprotons to create exotic particles that occur only at extremely high-energy densities. Observing these particles and measuring their properties may yield critical insights about the very nature of matter. Finding these rare particles requires solving difficult signal-versus-background classification problems. In the following, we consider two benchmark massive data sets, i.e., supersymmetry particles (SUSY) and Higgs bosons (HIGGS), as studied in [1]. These two data sets are available from the links: <https://archive.ics.uci.edu/ml/datasets/SUSY>, and <https://archive.ics.uci.edu/ml/datasets/HIGGS>, respectively. For both data sets, the parameter settings of FPC are the same as those in Section VI-A, except n is tuned from some ranges via a grid search (the ranges in SUSY and Higgs are [1500, 1700] and [300, 500], respectively, with 11 candidates). Since the training processes of SVM-RBF, SVM-Poly and random forest are very time-consuming for these two massive data sets, we only implement FPC and compare the performance of FPC with the state-of-the-art results in the recent literature.

Benchmark case for supersymmetry particles. The first benchmark classification task is to distinguish between a process where new supersymmetric particles (SUSY) are produced, leading to a final state, in which some particles are detectable and others are invisible to the experimental apparatus, and a background process with the same detectable particles but fewer invisible particles and distinct kinematic features. The SUSY data set includes **5 million sample points** produced using Monte Carlo simulations. The first 8 features are kinematic properties measured by the particle detectors in the accelerator. The last 10 features are functions

of the first 8 features. They are high-level features derived by physicists to help discriminate between the two classes. For better comparison, we only consider the first 8 low-level features for the classification as studied in the recent literature [26]. Specifically, we use **the last 500,000 sample points** as a test set. The rest sample points are divided into the training and validation sets, where the numbers of training and validation sets are **4 million** and **500,000**, respectively. We compare with many state-of-the-art methods including the k-nearest neighbors (kNN) [34], stochastic gradient descent (SGD), Hoeffding tree (HT) [9], and a set of ensemble methods suggested in [26] such as the Leveraging Bagging Hoeffding tree (LB-HT) [26], Hoeffding tree with kNN (HT-kNN), kNN with random feature (kNN-F), SGD with random feature (SGD-F), Leveraging Bagging SGD with random feature (LB-SGD-F), Hoeffding tree SGD with random feature (HT-SGD-F), where LB-HT is the state-of-the-art method for SUSY data set according to [26], in terms of the test accuracy (TestAcc) and training time (TrainTime) are presented in Table V.

From Table V, in terms of test accuracy, FPC outperforms all these state-of-the-art methods, while in the perspective of training time, FPC lies in the medium position with an affordable training time, though they are implemented on different platforms. Specifically, the first nine methods were implemented on GPU with 2880 simultaneous thresholds via CUDA, while FPC was implemented on a single CPU with one threshold via Matlab. Noting that the maximal iterations of the ADMM algorithm used in FPC is only 5, the computational complexity of the proposed method is about $\mathcal{O}(mn^2 + n^3 + 5mn)$ matrix-vector multiplies, where $m = 4,000,000$ is the number of training sample points, and $n = 1652$ is the number of centers selected. Since n is much less than m and $n^2 \approx m$, these imply that the total computational complexity of FPC is generally only about $\mathcal{O}(m^2)$, which shall be significantly lower than those of the other state-of-the-art methods.

Benchmark case for Higgs bosons. The second benchmark classification task is to distinguish between a signal process where new theoretical Higgs bosons (HIGGS) are produced, and a background process with the identical decay products but distinct kinematic features. The HIGGS data set includes **11 million sample points** produced using Monte Carlo simulations. For each individual sample point, there are 28 features, where the first 21 features are kinematic properties measured by the particle detectors in the accelerator, and the last seven features are functions of the first 21

TABLE IV: Testing time (in second), and sparsity levels of the estimated parameters for different algorithms.

Data sets	TestTime				Average sparsity			
	SVM-RBF	SVM-Poly	RF	FPC	SVM-RBF	SVM-Poly	RF	FPC
breast	0	0	0.03	0	60.6	136.6	13	41.25
banknote	0	0	0.03	0	281.6	273.4	12.7	58.5
seismic	0.02	0.02	0.042	0	339.0	236.8	11.3	27.55
musk2	0.66	0.61	0.11	0.002	1,314.2	1,184	15.5	3,299
HTRU2	0.14	0.09	0.20	0.006	638.8	699.6	13.7	165
MAGIC	1.26	0.83	0.28	0.002	4,561.8	4,504.3	18.5	286
occupancy	0.20	0.047	0.25	0	8,46.0	395.4	12.3	20.5
default	11.88	0.11	0.47	0.002	9,817.2	164.0	18.5	325
Skin	24.85	10.23	1.99	0.028	9,760.2	7,005.3	14.0	253

TABLE V: Performance comparison of different algorithms on SUSY data with 8 low-level features. The first nine columns are from the recent literature [26] as baselines, implemented on *GPU* in a parallel way (that is, *NVIDIA Tesla K40c with 12GB of RAM each, 15 SMX and up to 2880 simultaneous thresholds and CUDA 7.0*). The last column is the results of FPC, implemented on *single CPU* using *MATLAB* (that is, *Intel Xeron(R) CPU E5-2667 PC, RAM 256G, MATLAB R2015b*). The optimal parameters (s, #centers) for FPC in average are (5, 1652).

Methods	kNN	SGD	HT	LB-HT	HT-kNN	kNN-F	SGD-F	LB-SGD-F	HT-SGD-F	FPC
TestAcc (%)	67.5	76.5	78.2	78.7	77.2	71.2	77.7	77.7	78.4	78.99
TrainTime (seconds)	1,464	25	45	530	1,428	4,714	118	1,040	159	732.8

features. These are high-level features derived by physicists to help discriminate between the two classes. We use **the last 500,000 sample points** as a test set. The rest sample points are divided into the training and validation sets, where the numbers of training and validation sets are **10 million** and **500,000**, respectively. We compare with many state-of-the-art methods including the linear SVM (Linear SVM), logistic regression (Logit), alternating direction method of multipliers (ADMM) for deep neural networks (called, *ADMM-DNN* for short) [41], and cartesian genetic programming (CGP) [19], where the performance of ADMM-DNN can be regarded as the baseline. The test accuracy, training time and the associated computational platform including both software and hardware are presented in Table VI.

From Table VI, in terms of test accuracy, FPC outperforms the state-of-the-art methods including the classical linear SVM, logistic regression, the deep learning method solved via ADMM as the baseline, and the advanced evolutionary algorithm, i.e., CGP. In the perspective of computational effectiveness, the training time of FPC is far less than those of linear SVM and logistic regression implemented on a fast distributed system (Apache Spark built on Hadoop 2.0), and also that of CGP implemented on a fast evolutionary computational platform (ECJ), while FPC can be easily implemented via Matlab on a single PC only requiring enough RAM. Although the training time of ADMM-DNN seems much less than that of FPC, it is implemented on multi-core (up to 7200) CPUs, yet FPC is implemented with single CPU core. According to the similar analysis in the SUSY experiment, the computational complexity of FPC in this experiment is also about $\mathcal{O}(mn^2 + n^3 + 5mn)$ matrix-vector multiplies, where $m = 10,000,000$ is the number of training sample points, $n = 498$ is the number of centers selected, and 5 is the maximal iterations. Since n is much smaller than the training sample points m , the computational complexity of

FPC is only about $\mathcal{O}(mn^2)$. Particularly, for Higgs data set, the computational complexity of FPC is about 10^{13} matrix-vector multiplies, which shall be much less than those of the other methods.

C. Image classification on Dogs vs. Cats competition data

The image classification problem plays a very important role in modern fields of computer vision and machine learning. In order to exploit the machine learning methods efficiently, it commonly requires large amount of manually labeled training data. However, labeling images in a manual way is generally very expensive and time-cost. To overcome this difficulty, we can usually exploit the internet and obtain lots of labeled images via searching in the search engine according to some specified categories. Although this way is relatively cheaper and easier, there may be many noisy images with uncorrect labels and even some of them are outliers, as shown in Figure 13(a) and (b), where the false images are marked in red boxes. Thus, in practice, it is urgent to find an efficient classification method to automatically recognize correct labels of a given image, instead of doing in a manual way.

Dogs vs. Cats [†] is a famous competition in the Kaggle to classify whether images contain either a dog or a cat. This competition data contains 12,500 images of dogs and 12,500 images of cats. In our experiment, we use totally 25,000 labeled images as the testing set, and then search some images of dogs and cats respectively using the search engine to build up the training set (3,911 images in total). Note that the training set includes some noisy images or outliers, that is, there shall be some false images in each category. Some training images are shown in Figure 13(a) and (b). We compare with many state-of-the-art methods including boosting, ϵ -boosting, re-scaled boosting [47], SVM-RBF, SVM-Poly, and

[†]<https://www.kaggle.com/c/dogs-vs-cats>.

TABLE VI: Performance comparison of different algorithms on Higgs data. The optimal parameters (s , #centers) in average for FPC are (2, 498).

Methods	Linear SVM [35]	Logit [35]	ADMM-DNN [41]	CGP [19]	FPC
TestAcc (%)	52.0	60.8	64.0	64.6	65.39
TrainTime (seconds)	18,337.8	14,364.4	7.8	9,000	275.73
Software	Apache Spark	Apache Spark	Python, MPI	ECJ	Matlab R2015b
Hardware	Hadoop 2.0	Hadoop 2.0	7200 CPUs , Cray XC30	single CPU	single CPU

random forest, where re-scaled boosting is the state-of-the-art method for this data set. The parameter settings of FPC are the same as those in Higgs data set, while the optimal number of trees used for random forest (RF) is tuned via a grid search from the range [4, 40] with 10 candidates, i.e., $\{4, 8, \dots, 40\}$. The test accuracy and training time are presented in Table VI.

Different from the previous UCI data sets, such an image classification task cannot be handled directly in the pixel level but should be implemented in the feature representation level. Thus, we firstly exploit the famous googLeNet [40] to translate the training and testing images into feature representation with 1024 dimensions (corresponding to 32×32 image patches). Then among the training images, we randomly select 2,000 images for training via a uniform distribution, while the rest 1,911 images are used as the validation set to tune parameters of different algorithms. Finally, we evaluate the performance of each fine-tuned algorithms on 25,000 test images. For all the methods, we repeat 50 times of experiments, and record the test accuracies and the average training time, reported in Table VII.

From Table VII, the test accuracy of our proposed method is comparable to those of the random forest and boosting-type algorithms, which are viewed as the state-of-the-art algorithms for this data set. Particularly, FPC is much faster than the boosting-type algorithms and two classical SVM methods, and slightly faster than random forest. Some predictions on the test images are shown in Figure 13(c), where two false images among 64 test images are marked in the red boxes.

VII. CONCLUSION

The design of efficient classification methods for massive data classification is an import topic in the era of big data. Classical kernel approaches are not scalable enough to handle massive data, mainly because they are required to map the original data into a very high dimensional space whose dimension is the size of samples, which is commonly “unnecessary high” for classification. Due to such a kernel trick, the computational burden and storage of the classical kernel approaches are generally unaffordable when dealing with the massive data classification problem. In this paper, we propose a fast, efficient learning scheme called *FPC* for massive data classification based on polynomial kernels. We exploit some subsampling scheme, based on which an effective feature mapping can be constructed from polynomial kernels. Instead of mapping the original data into the very high dimensional space, the constructed feature mapping generally maps the original data into a relatively low dimensional feature spaces



Fig. 13: Experiment results for the set of Dogs vs. Cats. The details are better seen by zooming on a computer scene.

efficiently via exploiting the alternating direction method of multipliers (ADMM). The effectiveness of the proposed learning scheme is verified in both theory and numerical experiments. Theoretically, we justify that the suggested learning scheme preserves almost the same generalization power of SVM. Numerically, the proposed FPC is much faster than SVM but does not sacrifice its generalization. This shows that FPC brings some extent of possibility for handling massive data classification problems.

ACKNOWLEDGMENTS

The authors would like to thank Drs. Yao Wang and Xu Liao for sharing the *cats vs dogs* competition data set. The work of J. Zeng and M. Wu was partially supported by the National Natural Science Foundation of China [Grants No. 61603162, 61977038]. The work of S.B. Lin was supported by the National Natural Science Foundation of China [Grant No.

TABLE VII: Performance comparison of different algorithms on Dogs vs. Cats competition data. The first three columns are from [47] implemented on the same platform of this paper. The optimal parameters (s , #centers) for FPC in average are $(2, 401.6)$, and the optimal number of trees for RF in average is 33.52 .

Methods	Boosting	Re-scaled boosting	ϵ -boosting	SVM-RBF	SVM-Poly	RF	FPC
TestAcc (%)	95.12	96.98	96.71	96.25	96.53	96.74	96.83
TrainTime (seconds)	158.4	165.6	180.5	8,348.6	6,220.8	4.35	1.24

61876133], and the work of D.X. Zhou was partially supported by the Research Grant Council of Hong Kong [Project No. CityU11338616].

APPENDIX A. PROOF OF THEOREM 1

According to [10], it is easy to derive the variational inequality (VI) reformulation of (8) as follows: Find $p^* = (u^*, v^*, w^*) \in \Omega := \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^m$ such that for any $p \in \Omega$,

$$\text{VI}(\Omega, F, f) := f(v) - f(v^*) + (p - p^*)^T F(p^*) \geq 0, \quad (23)$$

where

$$p = \begin{pmatrix} u \\ v \\ w \end{pmatrix}, \quad F(p) = \begin{pmatrix} A^T w \\ -w \\ v - Au \end{pmatrix}. \quad (24)$$

Note that the mapping $F(p)$ is monotone because it is affine with a skew-symmetric matrix. We denote by Ω^* the solution set of $\text{VI}(\Omega, F, f)$. By [10], the VI formulation (23) is equivalent to the constrained formulation (8) in the sense that for any $p^* \in \Omega^*$, (u^*, v^*) is a minimizer of (8) and vice versa, that is, if (u^*, v^*) is a minimizer of (8), then there exists a w^* such that $p^* = (u^*, v^*, w^*) \in \Omega^*$. Furthermore, note from the relation between the problems (6) and (8), it is obvious that $\text{VI}(\Omega, F, f)$ is equivalent to the original unconstrained problem (6).

In order to prove Theorem 1, we need the following lemmas.

Lemma 2. Let $\{p^k\}$ be the sequence generated by (10), (11), (12). Then for any $k \in \mathbb{N}$ and $p \in \mathbb{R}^{n+2m}$, there holds

$$\begin{aligned} & f(v) - f(v^{k+1}) + \\ & (p - p^{k+1})^T [F(p^{k+1}) + \eta(v^k, v^{k+1}) + H(p^{k+1} - p^k)] \geq 0, \end{aligned} \quad (25)$$

where H is defined in (18), and

$$\eta(v^k, v^{k+1}) := \beta \begin{pmatrix} -A^T \\ \mathbf{I}_m \\ 0 \end{pmatrix} (v^k - v^{k+1}). \quad (26)$$

Proof: By the u^{k+1} -update (10), the optimality of u^{k+1} implies

$$\begin{aligned} 0 &= A^T [\beta(Au^{k+1} - v^k) + w^k] + \alpha(u^{k+1} - u^k) \\ &= A^T w^{k+1} - \beta A^T (v^k - v^{k+1}) + \alpha(u^{k+1} - u^k). \end{aligned} \quad (27)$$

By the v^{k+1} -update (11), the optimality of v^{k+1} implies

$$0 \in \partial f(v^{k+1}) - [\beta(Au^{k+1} - v^{k+1}) + w^k] = \partial f(v^{k+1}) - w^{k+1}$$

By the convexity of f , the above equality shows for any $k \in \mathbb{N}$,

$$f(v) - f(v^{k+1}) - (v - v^{k+1})^T w^{k+1} \geq 0, \quad \forall v \in \mathbb{R}^n. \quad (28)$$

It follows from (12) that

$$\beta^{-1}(w^{k+1} - w^k) - (Au^{k+1} - v^{k+1}) = 0. \quad (29)$$

Combining (27), (28) and (29) together, for any p , we have

$$\begin{aligned} & f(v) - f(v^{k+1}) \\ & + \begin{pmatrix} u - u^{k+1} \\ v - v^{k+1} \\ w - w^{k+1} \end{pmatrix}^T \left\{ \begin{pmatrix} A^T w^{k+1} - \beta A^T (v^k - v^{k+1}) \\ -w^{k+1} \\ v^{k+1} - Au^{k+1} \end{pmatrix} \right. \\ & \left. + \begin{pmatrix} \alpha(u^{k+1} - u^k) \\ 0 \\ \beta^{-1}(w^{k+1} - w^k) \end{pmatrix} \right\} \geq 0, \end{aligned}$$

which can be rewritten as

$$\begin{aligned} & f(v) - f(v^{k+1}) \\ & + \begin{pmatrix} u - u^{k+1} \\ v - v^{k+1} \\ w - w^{k+1} \end{pmatrix}^T \left\{ \begin{pmatrix} A^T w^{k+1} \\ -w^{k+1} \\ v^{k+1} - Au^{k+1} \end{pmatrix} \right. \\ & \left. + \beta \begin{pmatrix} -A^T (v^k - v^{k+1}) \\ v^k - v^{k+1} \\ 0 \end{pmatrix} \right. \\ & \left. + \begin{pmatrix} \alpha(u^{k+1} - u^k) \\ \beta(v^{k+1} - v^k) \\ \beta^{-1}(w^{k+1} - w^k) \end{pmatrix} \right\} \geq 0. \end{aligned} \quad (30)$$

By the notations of $F(p)$, $\eta(v^k, v^{k+1})$ and H , we get (25) immediately. ■

Lemma 3. Let $\{p^k\}$ be the sequence generated by (10), (11), (12), then for any $k \in \mathbb{N}$, and $p^* \in \Omega^*$, there holds

$$\|p^{k+1} - p^*\|_H^2 \leq \|p^k - p^*\|_H^2 - \|p^k - p^{k+1}\|_H^2. \quad (31)$$

Proof: Since $p^* \in \Omega^*$, it follows from (23) that

$$f(v^{k+1}) - f(v^*) + (p^{k+1} - p^*)^T F(p^*) \geq 0.$$

By the monotonicity of F , we have

$$f(v^{k+1}) - f(v^*) + (p^{k+1} - p^*)^T F(p^{k+1}) \geq 0. \quad (32)$$

Note that (28) is satisfied for both k and $k+1$, that is, for any $v \in \mathbb{R}^n$,

$$\begin{aligned} & f(v) - f(v^{k+1}) - (v - v^{k+1})^T w^{k+1} \geq 0, \\ & f(v) - f(v^k) - (v - v^k)^T w^k \geq 0. \end{aligned}$$

Setting $v = v^k$ and $v = v^{k+1}$ in the first and second inequalities, respectively, and then adding them yields

$$\langle v^k - v^{k+1}, w^k - w^{k+1} \rangle \geq 0. \quad (33)$$

By using the notation of $\eta(v^k, v^{k+1})$, the relation $v^* = Au^*$, and the w^k -update (12), we have

$$\begin{aligned} & (p^{k+1} - p^*)^T \eta(v^k - v^{k+1}) \\ &= \beta(v^k - v^{k+1})^T [-A(u^{k+1} - u^*) + (v^{k+1} - v^*)] \\ &= \beta(v^k - v^{k+1})^T (v^{k+1} - Au^{k+1}) \\ &= (v^k - v^{k+1})^T (w^k - w^{k+1}) \geq 0, \end{aligned} \quad (34)$$

where the final inequality follows from (33).

Setting $p = p^*$ in (25), we have

$$\begin{aligned} & (p^{k+1} - p^*)^T H(p^k - p^{k+1}) \\ &\geq f(v^{k+1}) - f(v^*) + (p^{k+1} - p^*)^T F(p^{k+1}) \\ &\quad + (p^{k+1} - p^*)^T \eta(v^k, v^{k+1}) \\ &\geq 0, \end{aligned} \quad (35)$$

where the final inequality holds due to (32) and (34).

By (35), we have

$$\begin{aligned} \|p^k - p^*\|_H^2 &= \|(p^{k+1} - p^*) + (p^k - p^{k+1})\|_H^2 \\ &= \|p^{k+1} - p^*\|_H^2 + \|p^k - p^{k+1}\|_H^2 \\ &\quad + 2(p^{k+1} - p^*)^T H(p^k - p^{k+1}) \\ &\geq \|p^{k+1} - p^*\|_H^2 + \|p^k - p^{k+1}\|_H^2. \end{aligned}$$

This finishes the proof of this lemma. \blacksquare

Proof of Theorem 1: By Lemma 3, the generated sequence $\{p^k\}$ is bounded. Actually, it is contained by the compact set

$$\{p \in \mathbb{R}^{n+2m} : \|p - p^*\|_H^2 \leq \|p^0 - p^*\|_H^2\}$$

where p^* is an arbitrary point in Ω^* , $p^0 \in \mathbb{R}^{n+2m}$ is the initialization. Therefore, it is obvious that $\{p^k\}$ has at least one cluster point, say p^∞ , and we assume that a subsequence $\{p^{k_j}\}_{j \in \mathbb{N}}$ converges to p^∞ . Note that (31) directly implies $\|p^{k_j+1} - p^{k_j}\|_H^2 \rightarrow 0$ when $k_j \rightarrow \infty$. Thus, taking the limit over $k_j \rightarrow \infty$ in (25), we have that p^∞ is a solution of VI(Ω, F, f) defined in (23), and thus, u^∞ is a global minimizer of (6). Again by (31), it implies that p^∞ is the unique cluster point of the sequence $\{p^k\}$. Thus, $\{p^k\}$ converges to p^∞ , a solution of (23), starting from any initial point p^0 . As a consequence, $\{u^k\}$ converges to a global minimizer of the original problem (6). This finishes the proof of this theorem. \blacksquare

APPENDIX B. PROOF OF THEOREM 2

For the convenience of analysis, we introduce some notations

$$M = \begin{pmatrix} \mathbf{I}_n & 0 & 0 \\ 0 & \mathbf{I}_m & 0 \\ 0 & -\beta \mathbf{I}_m & \mathbf{I}_m \end{pmatrix}, \quad (36)$$

$$Q = \begin{pmatrix} \alpha \mathbf{I}_n & 0 & 0 \\ 0 & \beta \mathbf{I}_m & 0 \\ 0 & -\mathbf{I}_m & \beta^{-1} \mathbf{I}_m \end{pmatrix}. \quad (37)$$

From the definitions of M and Q , it is easy to show that

$$Q = HM, \quad (Q^T + Q) - M^T HM \succeq 0. \quad (38)$$

Moreover, we introduce an auxiliary variable \tilde{p}^k defined as

$$\tilde{p}^k = \begin{pmatrix} \tilde{u}^k \\ \tilde{v}^k \\ \tilde{w}^k \end{pmatrix} = \begin{pmatrix} u^{k+1} \\ v^{k+1} \\ w^k + \beta(Au^{k+1} - v^k) \end{pmatrix}, \quad (39)$$

then we have

$$p^{k+1} = p^k - M(p^k - \tilde{p}^k). \quad (40)$$

We still need the following lemma to show the monotonicity of the sequence $\{\|p^{k+1} - p^k\|_H^2\}$.

Lemma 4. Let $\{p^k\}$ be the sequence generated by Algorithm 1, then for any $k \geq 1$, there holds

$$\|p^{k+1} - p^k\|_H^2 \leq \|p^k - p^{k-1}\|_H^2. \quad (41)$$

Proof: By using the definition (39) of \tilde{p}^k , and the facts

$$\begin{aligned} \beta^{-1}(w^k - \tilde{w}^k) &= -(Au^{k+1} - v^k) \\ &= -(A\tilde{u}^k - \tilde{v}^k) - (\tilde{v}^k - v^k), \end{aligned}$$

(30) can be rewritten as

$$\begin{aligned} f(v) - f(\tilde{v}^k) + (p - \tilde{p}^k)^T \left\{ \begin{pmatrix} A^T \tilde{w}^k \\ -\tilde{w}^k \\ \tilde{v}^k - A\tilde{u}^k \end{pmatrix} \right. \\ \left. + \begin{pmatrix} \alpha(\tilde{u}^k - u^k) \\ \beta(\tilde{v}^k - v^k) \\ -(\tilde{v}^k - v^k) + \beta^{-1}(\tilde{w}^k - w^k) \end{pmatrix} \right\} \geq 0, \quad \forall p \in \mathbb{R}^{n+2m}. \end{aligned}$$

By the definition (37) of Q and (24) of $F(p)$, the above inequality yields for any $p \in \Omega$,

$$f(v) - f(\tilde{v}^k) + (p - \tilde{p}^k)^T [F(\tilde{p}^k) + Q(\tilde{p}^k - p^k)] \geq 0. \quad (42)$$

Note that (42) also holds when k is replaced by $k+1$, and thus we have

$$\begin{aligned} f(v) - f(\tilde{v}^{k+1}) + (p - \tilde{p}^{k+1})^T [F(\tilde{p}^{k+1}) + Q(\tilde{p}^{k+1} - p^{k+1})] \geq 0. \end{aligned} \quad (43)$$

Setting $p = \tilde{p}^{k+1}$ and $p = \tilde{p}^k$ in (42) and (43), respectively, we have

$$f(\tilde{v}^{k+1}) - f(\tilde{v}^k) + (\tilde{p}^{k+1} - \tilde{p}^k)^T [F(\tilde{p}^{k+1}) + Q(\tilde{p}^{k+1} - p^{k+1})] \geq 0,$$

and

$$\begin{aligned} f(\tilde{v}^k) - f(\tilde{v}^{k+1}) \\ + (\tilde{p}^k - \tilde{p}^{k+1})^T [F(\tilde{p}^{k+1}) + Q(\tilde{p}^{k+1} - p^{k+1})] \geq 0. \end{aligned}$$

Adding the above two inequalities and using the monotonicity of F yields

$$(\tilde{p}^k - \tilde{p}^{k+1})^T Q [(p^k - p^{k+1}) - (\tilde{p}^k - \tilde{p}^{k+1})] \geq 0. \quad (44)$$

Adding the term

$$[(p^k - p^{k+1}) - (\tilde{p}^k - \tilde{p}^{k+1})]^T Q [(p^k - p^{k+1}) - (\tilde{p}^k - \tilde{p}^{k+1})]$$

to both sides of (44) and using $p^T Q p = \frac{1}{2} p^T (Q^T + Q) p$, we have

$$\begin{aligned} & (p^k - p^{k+1})^T Q [(p^k - p^{k+1}) - (\tilde{p}^k - \tilde{p}^{k+1})] \\ & \geq \frac{1}{2} \|(p^k - p^{k+1}) - (\tilde{p}^k - \tilde{p}^{k+1})\|_{Q^T + Q}^2 \\ & = \frac{1}{2} \|(p^k - \tilde{p}^k) - (\tilde{p}^{k+1} - \tilde{p}^{k+1})\|_{Q^T + Q}^2. \end{aligned}$$

By (38) and (40), the above inequality implies

$$\begin{aligned} & (p^k - \tilde{p}^k)^T M^T H M [(p^k - \tilde{p}^k) - (p^{k+1} - \tilde{p}^{k+1})] \\ & \geq \frac{1}{2} \|(p^k - \tilde{p}^k) - (p^{k+1} - \tilde{p}^{k+1})\|_{Q^T + Q}^2. \end{aligned} \quad (45)$$

Setting $a = M(p^k - \tilde{p}^k)$ and $b = M(p^{k+1} - \tilde{p}^{k+1})$ in the identity

$$\|a\|_H^2 - \|b\|_H^2 = 2a^T H(a - b) - \|a - b\|_H^2,$$

we obtain

$$\begin{aligned} & \|M(p^k - \tilde{p}^k)\|_H^2 - \|M(p^{k+1} - \tilde{p}^{k+1})\|_H^2 \\ & = 2(p^k - \tilde{p}^k)^T M^T H M [(p^k - \tilde{p}^k) - (p^{k+1} - \tilde{p}^{k+1})] \\ & \quad - \|M[(p^k - \tilde{p}^k) - (p^{k+1} - \tilde{p}^{k+1})]\|_H^2. \end{aligned}$$

Plugging (45) into the above inequality yields

$$\begin{aligned} & \|M(p^k - \tilde{p}^k)\|_H^2 - \|M(p^{k+1} - \tilde{p}^{k+1})\|_H^2 \\ & \geq \|(p^k - \tilde{p}^k) - (p^{k+1} - \tilde{p}^{k+1})\|_{[Q^T + Q]}^2 \\ & \quad - \|M[(p^k - \tilde{p}^k) - (p^{k+1} - \tilde{p}^{k+1})]\|_H^2 \\ & = \|(p^k - \tilde{p}^k) - (p^{k+1} - \tilde{p}^{k+1})\|_{[(Q^T + Q) - M^T H M]}^2 \\ & \geq 0, \end{aligned}$$

where the last inequality is due to $(Q^T + Q) - M^T H M \succeq 0$ via (38). Furthermore, by (40), the above inequality implies (41). This finishes the proof of this lemma. ■

Based on Lemmas 3 and 4, we can prove Theorem 2 as follows.

Proof of Theorem 2: By Lemma 3, we have

$$\sum_{t=0}^{\infty} \|p^t - p^{t+1}\|_H^2 \leq \|p^0 - p^*\|_H^2, \quad \forall p^* \in \Omega^*. \quad (46)$$

According to Lemma 4, the sequence $\{\|p^t - p^{t+1}\|_H^2\}$ is non-increasing. Thus, by [7, Lemma 1.1], we can get the $o(1/k)$ convergence rate of $\|p^t - p^{t+1}\|_H^2$. This finishes the proof. ■

APPENDIX C. PROOF OF THEOREM 3

Denote $\phi(t) := (1-t)_+$ and $\mathcal{E}(f) := \int_Z \phi(yf(x)) d\rho$ for $f \in L^2_{\rho_X}$. It can be found in [52] that

$$\mathcal{R}(\text{sgn}(f)) - \mathcal{R}(f_c) \leq \mathcal{E}(f) - \mathcal{E}(f_\rho), \quad (47)$$

where f_ρ is defined by (20). Thus, to prove Theorem 3, it suffices to bound $\mathcal{E}(\pi f_{D,n}) - \mathcal{E}(f_\rho)$, where $\pi t = \text{sgn}(t) \cdot \min\{1, t\}$ denotes the truncation of $t \in \mathbb{R}$ to $[-1, 1]$, and $\text{sgn}(t)$ is defined as follows

$$\text{sgn}(t) := \begin{cases} 1, & \text{if } t \geq 0, \\ -1, & \text{if } t < 0. \end{cases}$$

Define the empirical version of $\mathcal{E}(f)$ as $\mathcal{E}_D(f) := \frac{1}{m} \sum_{i=1}^m \phi(y_i f(x_i))$. Then, we can deduce the following error decomposition easily.

Proposition 3. Let $f_{D,n}$ be defined by (5). Then for $f_0 \in \mathcal{H}_{\eta,n}$, there holds

$$\mathcal{E}(\pi f_{D,n}) - \mathcal{E}(f_\rho) \leq \mathcal{D}(f_0) + \mathcal{S}_D(f_0) - \mathcal{S}_D(\pi f_{D,n}), \quad (48)$$

where

$$\mathcal{D}(f_0) := \mathcal{E}(f_0) - \mathcal{E}(f_\rho), \quad (49)$$

and

$$\mathcal{S}_D(f) := [\mathcal{E}_D(f) - \mathcal{E}_D(f_\rho)] - [\mathcal{E}(f) - \mathcal{E}(f_\rho)]. \quad (50)$$

Proof: Direct computations yield

$$\begin{aligned} & \mathcal{E}(\pi f_{D,n}) - \mathcal{E}(f_\rho) = \mathcal{E}(f_0) - \mathcal{E}(f_\rho) - \mathcal{E}(f_0) + \mathcal{E}_D(f_0) \\ & \quad + \mathcal{E}_D(\pi f_{D,n}) - \mathcal{E}_D(f_0) - \mathcal{E}_D(\pi f_{D,n}) + \mathcal{E}(\pi f_{D,n}). \end{aligned}$$

Then, it follows from (5) that

$$\mathcal{E}_D(\pi f_{D,n}) \leq \mathcal{E}_D(f_{D,n}) \leq \mathcal{E}_D(f_0).$$

Therefore,

$$\begin{aligned} & \mathcal{E}(\pi f_{D,n}) - \mathcal{E}(f_\rho) \leq \mathcal{E}(f_0) - \mathcal{E}(f_\rho) - \mathcal{E}(f_0) + \mathcal{E}_D(f_0) \\ & \quad + \mathcal{E}(\pi f_{D,n}) - \mathcal{E}_D(\pi f_{D,n}) = \mathcal{D}(f_0) + \mathcal{S}_D(f_0) - \mathcal{S}_D(\pi f_{D,n}). \end{aligned}$$

This completes the proof of Proposition 3. ■

Based on the above error decomposition, we need to bound $\mathcal{D}(f_0)$, $\mathcal{S}_D(f_0)$ for some $f_0 \in \mathcal{H}_{\eta,n}$ and $-\mathcal{S}_D(\pi f_{D,n})$ respectively. The first two bounds are easy, which can be found in [44, Proposition 3] and [44, Proposition 4].

Proposition 4. If ρ satisfies (21), then

$$\mathcal{D}(B_s(f_\rho)) := \mathcal{E}(B_s(f_\rho)) - \mathcal{E}(f_\rho) \leq 4c_1 5^\alpha s^{-\alpha},$$

where $B_s(f)$ is the Bernstein polynomial for a function f on the simplex X defined as

$$B_s(f)(x) := B_{s,d}(f, x) = \sum_{|k| \leq s} f\left(\frac{k}{s}\right) P_{k,s}(x), \quad x \in X,$$

$$\text{where } P_{k,s}(x) = \binom{s+d}{d} x^k (1-|x|)^{s-|k|}.$$

Proposition 5. If ρ satisfies (19), then for any $0 < \delta < 1$, with the confidence at least $1 - \delta/2$, there holds

$$\mathcal{S}_D(B_s(f_\rho)) \leq \frac{8 \log(2/\delta)}{3m} + \left(\frac{2c_q \log(2/\delta)}{m} \right)^{\frac{q+1}{q+2}} + \frac{1}{2} \mathcal{D}(B_s(f_\rho)).$$

The most challenging part in our analysis is to bound $-\mathcal{S}_D(\pi f_{D,n})$, for which we adopt the approaches in our recent work [24] and [22]. To this end, we need four lemmas and some definitions concerning the capacity of a space. Let \mathcal{B} be a Banach space and \mathcal{V} a compact subset of \mathcal{B} . The quantity $H_\varepsilon(\mathcal{V}, \mathcal{B}) = \log_2 \mathcal{N}_\varepsilon(\mathcal{V}, \mathcal{B})$, where $\mathcal{N}_\varepsilon(\mathcal{V}, \mathcal{B})$ is the least number of elements in an ε -net of \mathcal{V} , is called ε -entropy of \mathcal{V} in \mathcal{B} . The quantity $\mathcal{N}_\varepsilon(\mathcal{V}, \mathcal{B})$ is called the ε -covering number of \mathcal{V} . If a vector $\mathbf{t} = (t_1, \dots, t_n)$ belongs to \mathbb{R}^n , then we denote by $\text{sgn}(\mathbf{t})$ the vector $(\text{sgn}(t_1), \dots, \text{sgn}(t_n))$. The VC dimension [16] of a set \mathcal{V} over X , denoted as $VCdim(\mathcal{V})$, is defined as the maximal natural number l such that there exists a collection (ξ_1, \dots, ξ_l) in X such that the cardinality of the sgn -vector set

$$S = \{(\text{sgn}(v(\xi_1)), \dots, \text{sgn}(v(\xi_l))) : v \in \mathcal{V}\}$$

equals to 2^l , that is, the set S coincides with the set of all vertexes of unit cube in \mathbb{R}^l . The quantity

$$Pdim(\mathcal{V}) := \max_g VCdim(\mathcal{V} + g),$$

is called the pseudo-dimension [25] of the set \mathcal{V} over X , where g runs over the set of all functions defined on X and $\mathcal{V} + g = \{v + g : v \in \mathcal{V}\}$.

The first lemma establishes some important relations among the pseudo-dimension, ε -entropy and VC-dimension, which can be found in [29].

Lemma 5. *Let \mathcal{V}_R be a class of functions which consists of all functions $f \in \mathcal{V}$ satisfying $|f(x)| \leq R$ for all $x \in X$. Then*

$$VC(\mathcal{V}) \leq Pdim(\mathcal{V})$$

and

$$H_\varepsilon(\mathcal{V}_R, L^2(X)) \leq c_2 Pdim(\mathcal{V}_R) \log_2 \left(\frac{R}{\varepsilon} \right),$$

where c_2 is an absolute positive constant.

The second one is a covering number estimate of $\mathcal{H}_{\eta,n}$.

Lemma 6. *Let $\mathcal{H}_{\eta,n}$ be the space defined by (4) with $n = \binom{s+d}{s}$. Define further $\pi\mathcal{H}_{\eta,n} := \{\pi f : f \in \mathcal{H}_{\eta,n}\}$. Then for any $\varepsilon > 0$,*

$$\mathcal{H}_\varepsilon(\pi\mathcal{H}_{\eta,n}, C(X)) \leq c'_2 s^d \log \frac{1}{\varepsilon},$$

where $c'_2 > 0$ depends only on d and $C(X)$ represents the set of all continuous functions defined on X .

Proof: Noting $\mathcal{H}_{\eta,n}$ is a linear space with dimension at most n , then it follows from [25] that $Pdim(\mathcal{H}_{\eta,n}) \leq n$. Since $\pi\mathcal{H}_{\eta,n} \subseteq \mathcal{H}_{\eta,n}$, By the definition of pseudo-dimension, we then have $Pdim(\pi\mathcal{H}_{\eta,n}) \leq n$ [25, pp. 297]. Then Lemma 5 implies

$$H_\varepsilon(\pi\mathcal{H}_{\eta,n}, L^2(X)) \leq c_2 n \log_2 \left(\frac{1}{\varepsilon} \right).$$

But $\|\cdot\|_{L^2} \leq \|\cdot\|_\infty$ implies

$$\mathcal{N}_\varepsilon(\pi\mathcal{H}_{\eta,n}, C(X)) \leq \mathcal{N}_\varepsilon(\pi\mathcal{H}_{\eta,n}, L^2(X)).$$

The desired estimates then follows from $n \sim s^d$. ■

The third one is a classical concentration inequality based on the covering number estimates established by [48].

Lemma 7. *Let \mathcal{G} be a set of functions on Z . If for some $B \geq 0$, $0 \leq \alpha \leq 1$ and $c \geq 0$, every $g \in \mathcal{G}$ and $|g - \mathbf{E}g| \leq B$ almost everywhere and $\mathbf{E}(g^2) \leq c(\mathbf{E}g)^\alpha$, then for any $\varepsilon > 0$,*

$$\begin{aligned} \mathbf{P} \left\{ \sup_{g \in \mathcal{G}} \frac{\mathbf{E}g - \frac{1}{m} \sum_{i=1}^m g(z_i)}{\sqrt{(\mathbf{E}g)^\alpha + \varepsilon^\alpha}} > \varepsilon^{1-\frac{\alpha}{2}} \right\} \\ \leq \mathcal{N}_\varepsilon(\mathcal{G}, C(X)) \exp \left\{ -\frac{m\varepsilon^{2-\alpha}}{2(c + \frac{1}{3}B\varepsilon^{1-\alpha})} \right\}. \end{aligned}$$

The last one is an estimates for the solution to some equation, which was provided in [43, Lemma 4.2].

Lemma 8. *Let $a_1, a_2 > 0$ and $r_1 > r_2 > 0$. Then the equation*

$$t^{r_1} - a_1 t^{r_2} - a_2 = 0$$

has a unique positive zero t^* . In addition

$$t^* \leq \max \left\{ (2c_1)^{\frac{1}{r_1-r_2}}, (2c_2)^{\frac{1}{r_1}} \right\}.$$

Based on the above lemmas, we can bound $-\mathcal{S}_D(\pi f_{D,n})$ as follows.

Proposition 6. *Let $0 < \delta < 1$ and $f_{D,n}$ be defined by (5) with $n = \binom{s+d}{s}$, then with confidence at least $1 - \delta/2$, there holds*

$$-\mathcal{S}_D(\pi f_{D,n}) \leq \frac{1}{2} (\mathcal{E}(\pi f_{D,n}) - \mathcal{E}(f_\rho)) + c_3 \left[\frac{s^d \log m}{m} \right]^{\frac{q+1}{q+2}} \log \frac{4}{\delta},$$

where c_3 is a constant independent of m , s or δ .

Proof: Set

$$\mathcal{F}'_1 := \{\phi(yf) - \phi(yf_\rho) : f \in \pi\mathcal{H}_{\eta,n}\}.$$

Then for any $g \in \mathcal{F}'_1$, there exists an $f \in \pi\mathcal{H}_{\eta,n}$ such that $g(x) = \phi(yf(x)) - \phi(yf_\rho(x))$. Therefore,

$$\mathbf{E}g = \mathcal{E}(f) - \mathcal{E}(f_\rho) \geq 0, \quad \frac{1}{m} \sum_{i=1}^m g(z_i) = \mathcal{E}_D(f) - \mathcal{E}_D(f_\rho).$$

It is easy to check that $|g(z)| \leq 2$ and $|g - \mathbf{E}g| \leq 4$. Furthermore, for any $f \in \pi\mathcal{H}_{\eta,n}$, it can be found in [38] that under condition (19), there exists an absolute constant $c_4 \geq 1$ such that

$$\mathbf{E} \left\{ [\phi(y\pi f(x)) - \phi(yf_\rho(x))]^2 \right\} \leq c_4 [\mathcal{E}(\pi f) - \mathcal{E}(f_\rho)]^{\frac{q}{q+1}}.$$

Then Lemma 7 with $\alpha = \frac{q}{q+1}$, $c = c_4$ and $B = 4$ together with the definition of $\pi\mathcal{H}_{\eta,n}$ yields

$$\begin{aligned} \mathbf{P} \left\{ \sup_{f \in \mathcal{H}_{\eta,n}} \mathcal{Q}_{D,\varepsilon} > \varepsilon^{\frac{q+2}{2q+2}} \right\} \\ \leq \mathcal{N}_\varepsilon(\mathcal{F}'_1, C(X)) \exp \left\{ -\frac{m\varepsilon^{\frac{q+2}{q+1}}}{2(c_4 + \frac{4}{3}\varepsilon^{\frac{1}{q+1}})} \right\}, \end{aligned} \quad (51)$$

where

$$\mathcal{Q}_{D,\varepsilon}(f) = \frac{\mathcal{E}(\pi f) - \mathcal{E}(f_\rho) - (\mathcal{E}_D(\pi f) - \mathcal{E}_D(f_\rho))}{\sqrt{(\mathcal{E}(\pi f) - \mathcal{E}(f_\rho))^{\frac{q}{q+1}} + \varepsilon^{\frac{q}{q+1}}}}.$$

Observing that for any $f_1, f_2 \in \mathcal{H}_{\eta,n}$,

$$\begin{aligned} & |(\phi(y\pi f_1(x)) - \phi(yf_\rho(x))) - (\phi(y\pi f_2(x)) - \phi(yf_\rho(x)))| \\ &= |\phi(y\pi f_1(x)) - \phi(y\pi f_2(x))| \\ &\leq |\pi f_1(x) - \pi f_2(x)| \leq \|f_1 - f_2\|_\infty, \end{aligned}$$

we obtain

$$\mathcal{N}_\varepsilon(\mathcal{F}'_1, C(X)) \leq \mathcal{N}_\varepsilon(\pi\mathcal{H}_{\eta,n}, C(X)).$$

Inserting the above estimate into (51) and noting Lemma 6, we get

$$\begin{aligned} \mathbf{P} \left\{ \sup_{f \in \mathcal{H}_{\eta,n}} \mathcal{Q}_{D,\varepsilon} > \varepsilon^{\frac{q+2}{2q+2}} \right\} \\ \leq \exp \left\{ c_5 s^d \log \frac{1}{\varepsilon} \right\} \exp \left\{ -\frac{m\varepsilon^{\frac{q+2}{q+1}}}{2(c_4 + \frac{4}{3}\varepsilon^{\frac{1}{q+1}})} \right\}. \end{aligned} \quad (52)$$

for some constant c_5 depending only on d . Let $\psi(\varepsilon) := \log(\frac{1}{\varepsilon})$ and $\mathcal{A} := c_5 s^d$. We can define a function $l : \mathbb{R}_+ \rightarrow \mathbb{R}$ by

$$l(\varepsilon) := \mathcal{A}\psi(\varepsilon) - \frac{m\varepsilon^{\frac{q+2}{q+1}}}{2(c_4 + \frac{4}{3}\varepsilon^{\frac{1}{q+1}})}.$$

Since ψ is decreasing, we obtain that $l(\cdot)$ is decreasing. Thus, there exists a unique solution β^* to the equation

$$l(\beta) = \log \frac{\delta}{2}.$$

For any $\beta \geq m^{-\theta}$ with $\theta = \frac{q+1}{q+2}$, we have

$$l(\beta) \leq \mathcal{A}\psi(m^{-\theta}) - \frac{m\beta^{\frac{q+2}{q+1}}}{2(c_4 + \frac{4}{3}\beta^{\frac{1}{q+1}})} =: l_1(\beta). \quad (53)$$

Take β_1 to be the positive number satisfying

$$l_1(\beta_1) = \log \frac{\delta}{2}.$$

Then

$$\begin{aligned} \beta_1^{\frac{q+2}{q+1}} - \frac{8(\mathcal{A}\psi(m^{-\theta}) + \log \frac{2}{\delta})}{3m} \beta_1^{\frac{1}{q+1}} \\ - \frac{2c_4(\mathcal{A}\psi(m^{-\theta}) + \log \frac{2}{\delta})}{m} = 0. \end{aligned}$$

Using Lemma 8 with $r_1 = \theta$, $r_2 = \frac{1}{q+1}$,

$$a_1 = \frac{8(\mathcal{A}\psi(m^{-\theta}) + \log \frac{2}{\delta})}{3m},$$

and

$$a_2 = \frac{2c_4(\mathcal{A}\psi(m^{-\theta}) + \log \frac{2}{\delta})}{m},$$

we get

$$\beta_1 \leq (6 + 4c_4) \left[\frac{(\mathcal{A}\psi(m^{-\theta}) + \log \frac{2}{\delta})}{m} \right]^{\frac{q+1}{q+2}}. \quad (54)$$

Setting $c_6 := (6 + 4c_4) + (\mathcal{A}\theta)^{\theta}$, we obtain

$$\beta_1 \leq c_6 \left[\frac{(\mathcal{A}\psi(m^{-\theta}) + 1)}{m} \right]^{\frac{q+1}{q+2}} \log \frac{4}{\delta} =: \beta_2.$$

It is easy to check that $\beta_2 \geq m^{-\theta}$. Then (53) implies that

$$l(\beta_2) \leq l_1(\beta_2) \leq l_1(\beta_1) = \log \frac{\delta}{2} = l(\beta^*).$$

Hence the monotone decreasing property of $l(\cdot)$ yields $\beta^* \leq \beta_2$. The above estimate together with (52) implies that with confidence at least $1 - \frac{\delta}{2}$,

$$\begin{aligned} -\mathcal{S}_D(\pi f_{D,n}) &= [\mathcal{E}(\pi f_{D,n}) - \mathcal{E}(f_\rho)] - [\mathcal{E}_D(\pi f_{D,n}) - \mathcal{E}_D(f_\rho)] \\ &\leq \frac{q}{2q+2}(\mathcal{E}(\pi f_{D,n}) - \mathcal{E}(f_\rho)) + \frac{q}{2q+2}\beta^* + \frac{q+2}{2q+2}\beta^* \\ &\leq \frac{1}{2}(\mathcal{E}(\pi f_{D,n}) - \mathcal{E}(f_\rho)) + c_6 \left[\frac{(\mathcal{A}\psi(m^{-\theta}) + 1)}{m} \right]^{\frac{q+1}{q+2}} \log \frac{4}{\delta}, \end{aligned}$$

where the first inequality holds for the Young's inequality. Plugging the definitions of $\mathcal{A} = c_5 s^d$ and $\psi(\epsilon) = \log(1/\epsilon)$ and $\theta = \frac{q+1}{q+2}$ into the above inequality, this finishes the proof of Proposition 6. ■

With the above tools, we can prove Theorem 3 as follows.

Proof of Theorem 3: Combining Proposition 3 with Proposition 4, Proposition 5, and Proposition 6, with confidence $1 - \delta$, there holds

$$\begin{aligned} \mathcal{E}(\pi f_D) - \mathcal{E}(f_\rho) &\leq 6c_1 5^\alpha s^{-\alpha} \\ &+ \frac{8 \log(4/\delta)}{3m} + \left(\frac{2c_q \log(4/\delta)}{m} \right)^{\frac{q+1}{q+2}} \\ &+ \frac{1}{2}(\mathcal{E}(\pi f_D) - \mathcal{E}(f_\rho)) + c_3 \left[\frac{s^d \theta \log m}{m} \right]^{\frac{q+1}{q+2}} \log \frac{4}{\delta}. \end{aligned}$$

Thus, with confidence $1 - \delta$,

$$\begin{aligned} \mathcal{E}(\pi f_D) - \mathcal{E}(f_\rho) \\ \leq c_4 \left[s^{-\alpha} + m^{-1} \log(4/\delta) + (m^{-1} \log(4/\delta))^{\frac{q+1}{q+2}} \right. \\ \left. + s^{\frac{d(q+1)}{q+2}} (m^{-1} \log m)^{\frac{q+1}{q+2}} \log \frac{4}{\delta} \right], \end{aligned}$$

for some constant $c_4 > 0$. Thus, if $s \sim \left(\frac{m}{\log m} \right)^{\frac{q+1}{\alpha(q+2)+d(q+1)}}$, then

$$\mathcal{E}(\pi f_D) - \mathcal{E}(f_\rho) \leq c \left(\frac{m}{\log m} \right)^{-\frac{\alpha(q+1)}{\alpha(q+2)+d(q+1)}} \log \left(\frac{4}{\delta} \right)$$

for some constant $c > 0$. Then Theorem 3 follows from (47). ■

APPENDIX D. PROOF OF LEMMA 1

Proof: The identity (13) can be derived from (10) directly.

Define

$$\text{Hinge}_\gamma(\xi, \zeta)$$

$$= \arg \min_{v \in \mathbb{R}^m} \left\{ \sum_{i=1}^m (1 - \xi(i) \cdot v(i))_+ + \frac{\gamma}{2} \|v - \zeta\|_2^2 \right\}$$

We can derive (14) directly. Let

$$\text{hinge}_\gamma(a, b) = \arg \min_{z \in \mathbb{R}} \left\{ \max\{0, 1 - a \cdot z\} + \frac{\gamma}{2}(z - b)^2 \right\},$$

for some $\gamma > 0$ and $a, b \in \mathbb{R}$, then we have

$$\begin{aligned} \text{Hinge}_\gamma(\xi, \zeta) \\ = (\text{hinge}_\gamma(\xi(1), \zeta(1)), \dots, \text{hinge}_\gamma(\xi(m), \zeta(m)))^T, \end{aligned}$$

and

$$\begin{aligned} \text{hinge}_\gamma(a, b) = \\ \begin{cases} b, & \text{if } a = 0, \\ b + \gamma^{-1}a, & \text{if } a \neq 0 \text{ and } ab \leq 1 - \gamma^{-1}a^2, \\ a^{-1}, & \text{if } a \neq 0 \text{ and } 1 - \gamma^{-1}a^2 < ab < 1, \\ b, & \text{if } a \neq 0 \text{ and } ab \geq 1. \end{cases} \end{aligned}$$

The only remainder is to prove (15). Given $a, b \in \mathbb{R}$, and $\gamma > 0$, let

$$g(z) := \max\{0, 1 - a \cdot z\} + \frac{\gamma}{2}(z - b)^2.$$

We consider the minimization problem in the following three different cases: (1) $a > 0$, (2) $a = 0$ and (3) $a < 0$.

Case 1. $a > 0$: In this case,

$$g(z) = \begin{cases} 1 - az + \frac{\gamma}{2}(z - b)^2, & z < a^{-1}, \\ \frac{\gamma}{2}(z - b)^2, & z \geq a^{-1}. \end{cases}$$

It is easy to show that the solution of the problem is

$$z^* = \begin{cases} b + \gamma^{-1}a, & \text{if } a > 0 \text{ and } b \leq a^{-1} - \gamma^{-1}a, \\ a^{-1}, & \text{if } a > 0 \text{ and } a^{-1} - \gamma^{-1}a < b < a^{-1}, \\ b, & \text{if } a > 0 \text{ and } b \geq a^{-1}. \end{cases}$$

Case 2. $a = 0$: It is obvious that

$$z^* = b.$$

Case 3. $a < 0$: Similar to Case 1,

$$g(z) = \begin{cases} 1 - az + \frac{\gamma}{2}(z - b)^2, & z \geq a^{-1}, \\ \frac{\gamma}{2}(z - b)^2, & z < a^{-1}. \end{cases}$$

Similarly, it is easy to show that the solution of the problem is

$$z^* = \begin{cases} b + \gamma^{-1}a, & \text{if } a < 0 \text{ and } b \geq a^{-1} - \gamma^{-1}a, \\ a^{-1}, & \text{if } a < 0 \text{ and } a^{-1} < b < a^{-1} - \gamma^{-1}a, \\ b, & \text{if } a < 0 \text{ and } b \leq a^{-1}. \end{cases}$$

Thus, we finish the proof of this lemma. ■

APPENDIX E. PROOF OF PROPOSITION 2

Proof: It is obvious that the problem (6) is equivalent to the following constrained optimization problem

$$\begin{aligned} \min_{u \in \mathbb{R}^n, \xi \in \mathbb{R}^m} \quad & \frac{1}{m} \sum_{i=1}^m \xi_i \\ \text{s.t. } & y_i \sum_{j=1}^n A_{ij} u_j \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, 2, \dots, m. \end{aligned}$$

The Lagrangian function of the above problem is

$$\begin{aligned} \mathcal{L}(u, \xi, \mathbf{a}, \mathbf{c}) & \quad (55) \\ & = \frac{1}{m} \sum_{i=1}^m \xi_i + \sum_{i=1}^m a_i \left(1 - \xi_i - y_i \sum_{j=1}^n A_{ij} u_j \right) - \sum_{i=1}^m c_i \xi_i, \end{aligned}$$

where $a_i, c_i \geq 0$, $i = 1, \dots, m$ are the multipliers. Let $\mathbf{a} := (a_1, a_2, \dots, a_m)^T$, $\mathbf{c} := (c_1, c_2, \dots, c_m)^T$. Taking derivatives of the Lagrangian function \mathcal{L} with u and ξ , and letting them equal to zero yields

$$A^T \text{Diag}(y) \mathbf{a} = 0, \quad (56)$$

$$\mathbf{a} + \mathbf{c} = \frac{1}{m} \mathbf{1}_m. \quad (57)$$

Plugging the above two equations into (55), then the Lagrangian function \mathcal{L} (55) becomes $\mathbf{1}_m^T \mathbf{a}$. This, together with the equations (56) and (57) yield the dual form (7) presented in Proposition 2. ■

REFERENCES

- [1] P. Baldi, P. Sadowski, and D. Whiteson, Searching for exotic particles in high-energy physics with deep learning, *Nature Communication*, 5: 4308, 2014.
- [2] L. Breiman, Random forests, *Machnie Learning*, 45: 5-32, 2001.
- [3] D. R. Chen, Q. Wu, Y. M. Ying, and D. X. Zhou, Support vector machine soft margin classifiers: error analysis, *J. Mach. Learn. Res.*, 5: 1143-1175, 2004.
- [4] M. Chen, S. Mao, Y. Liu, Big data: A survey, *Mobile Netw. Appl.*, 19: 171-209, 2014.
- [5] L.J. Chien, C.C. Chang, Y.J. Lee, Variant methods of reduced set selection for reduced support vector machines, *J. Inf. Sci. Eng.*, 26(1): 183-196, 2010.
- [6] F. Cucker, and D. X. Zhou, Learning theory: an approximation theory viewpoint. *Cambridge University Press*, Cambridge, 2007.
- [7] W. Deng, M. Lai, Z. Peng, and W. Yin, Parallel multi-block ADMM with $o(1/k)$ convergence, *Journal of Scientific Computing*, 71: 712-736, 2017.
- [8] F. Dimpert and A. Christmann, Universal consistency and robustness of localized support vector machines, *Neurocomputing*, 315: 96-106, 2018.
- [9] P. Domingos, and G. Hulten, Mining high-speed data streams. in *Proceed. of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 71-80, 2000.
- [10] F. Facchinei, and J.S. Pang, Finite-dimensional variational inequalities and complementarity problem, Vol. I. *Springer Ser. Oper. Res.*, Springer-Verlag, New York, 2003.
- [11] P. A. Forero, A. Cano and G. B. Giannakis, Consensus-based distributed support vector machines, *J. Mach. Learn. Res.*, 11: 1663-1707, 2010.
- [12] Y. Freund and R. E. Schapire, A decision-theoretic generalization of online learning and an application to boosting, *J. Comput. Syst. Sci.*, 55(1): 119-139, 1997.
- [13] D. Gabay, and B. Mercier, A dual algorithm for the solution of nonlinear variational problems via finite-element approximations, *Comp. Maths. with Appl.*, 2: 17-40, 1976.
- [14] A. Gittens and M. W. Mahoney, Revisiting the Nyström method for improved large scale machine learning, *J. Mach. Learn. Res.*, 17: 1-65, 2016.
- [15] Z. C. Guo, D. H. Xiang, X. Guo, and D. X. Zhou, Thresholded spectral algorithms for sparse approximations, *Anal. Appl.*, 15: 433-455, 2017.
- [16] L. Györfy, M. Kohler, A. Krzyzak, and H. Walk, A distribution-free theory of nonparametric regression, *Springer*, Berlin, Germany, 2002.
- [17] M. Hagan, M. Beale, and H. Demuth, Neural network design, *PWS Publishing Company*, Boston, 1996.
- [18] B. He, and X. Yuan, On non-ergodic convergence rate of Douglas-Rachford alternating direction method of multipliers, *Numer. Math.*, 50(130): 567-577, 2015.
- [19] H. Hmida, S.B. Hamida, A. Borgi, and M. Rukoz, Scale genetic programming for large data sets: case of Higgs Bosons classification, in *Proceed. of the International Conference on Knowledge Based and Intelligent Information and Engineering Systems, KES2018*, 3-5 September 2018, Belgrade, Serbia.
- [20] W.C. Kao, K.M. Chung, C.L. Sun, and C.J. Lin. Decomposition methods for linear support vector machines, *Neural Computation*, 16: 1689-1704, 2004.
- [21] S. B. Lin, X. Guo and D. X. Zhou, Distributed learning with regularized least squares, *J. Mach. Learn. Res.*, 18: 1-31, 2017.
- [22] S. Lin, J. Zeng, and X. Chang, Learning rates for classification with Gaussian kernels, *Neural Comput.*, 29(12): 3353-3380, 2017.
- [23] S. B. Lin and D. X. Zhou, Distributed kernel-based gradient descent algorithms, *Constr. Approx.*, 47: 249-276, 2018.
- [24] S. Lin, and J. Zeng, Fast learning with polynomial kernels, *IEEE Transactions on Cybernetics*, 49(10): 3780-3792, 2019.
- [25] V. Maiorov and J. Ratsaby, On the degree of approximation by manifolds of finite pseudo-dimension, *Constructive Approximation*, 15: 291-300, 1999.
- [26] D. Marron, J. Read, A. Bieft, and N. Navarro, Data stream classification using random feature functions and novel method combinations, *The Journal of Systems and Software*, 127: 195-204, 2017.
- [27] M. Meister and I. Steinwart, Optimal learning rates for localized SVMs, *J. Mach. Learn. Res.*, 17: 1-44, 2016.
- [28] S. Menard, Applied logistic regression analysis, *SAGE*, Loudon, 2002.
- [29] S. Mendelson, and R. Vershynin, Entropy and the combinatorial dimension, *Invent. Math.*, 125: 37-55, 2003.
- [30] Y. Nesterov, Smooth minimization of non-smooth functions, *Math. Program.*, 103(1): 127-152, 2005.

- [31] C. Platt, Fast training of support vector machines using sequential minimal optimization, in *B. Scholkopf, C.J.C. Burges, A.J. Smola (Eds.), Advances in Kernel Methods: Support Vector Learning*, MIT Press, 185-208, 1999.
- [32] S. Rosset, J. Zhu and T. Hastie, Boosting as a regularized path to a maximum margin classifier, *J. Mach. Learn. Res.*, 5:941-973, 2004.
- [33] A. Rudi, R. Camoriano and L. Rosasco, Less is more: Nyström computational regularization, in *Proceed. of the Twenty-ninth Conference on Neural Information Processing Systems (NIPS)*, 1657-1665, 2015.
- [34] A. Shaker, and E. Hullermeier, Instance-based classification and regression on data streams, in *Proceed. of the Learning in Non-stationary Environments*, Springer New York, pp. 185-201, 2012.
- [35] B.M. Shashidhara, S. Jain, V.D. Rao, N. Patil, and G.S. Raghavendra, Evaluation of machine learning frameworks on Bank Marketing and Higgs datasets, in *Proced. of the 2nd International Conference on Advances in Computing and Communication Engineering*, 2015.
- [36] X.J. Shen, L. Mu, Z. Li, H.X. Wu, J.P. Gou, and X. Chen, Large-scale support vector machine classification with redundant data reduction, *Neurocomputing*, 172: 189-197, 172.
- [37] I. Steinwart, On the influence of the kernel on the consistency of support vector machines, *J. Mach. Learn. Res.*, 2: 67-93, 2001.
- [38] I. Steinwart, and C. Scovel, Fast rates for support vector machines using Gaussian kernels, *Ann. Stat.*, 35: 575-607, 2007.
- [39] I. Steinwart, and A. Christmann, Support vector machines, *Springer*, New York, 2008.
- [40] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, Going deeper with convolutions, in *Proceed. of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [41] G. Taylor, R. Burmeister, Z. Xu, B. Singh, A. Patel, T. Goldstein, Training neural networks without gradients a scalable ADMM approach, in *Proceed. of the 33rd International Conference on Machine Learning (ICML)*, New York, USA, 2016.
- [42] J. S. Taylor and N. Cristianini, Kernel methods for pattern analysis, *Cambridge University Press*, Cambridge, 2004.
- [43] H. Z. Tong, D. R. Chen and Z. P. Li, Learning rates for regularized classifiers using multivariate polynomial kernels, *J. Complex.*, 24: 619-631, 2008.
- [44] H. Z. Tong, A note on support vector machines with polynomial kernels, *Neural Comput.*, 28: 71-88, 2016.
- [45] I. W. Tsant, J.T. Kwok, and P.M. Cheung, Core vector machines: fast svm training on very large data sets, *J. Mach. Learn. Res.*, 6: 363-392, 2005.
- [46] A. Tsybakov, Optimal aggregation of classifiers in statistical learning, *Ann. Stat.*, 32: 575-607, 2004.
- [47] Y. Wang, X. Liao, and S. Lin, Re-scaled boosting in classification, *IEEE Transactions on Neural Networks and Learning Systems*, 2019 (to appear).
- [48] Q. Wu, and D. X. Zhou, SVM soft margin classifiers: linear programming versus quadratic programming, *Neural Comput.*, 17: 1160-1187, 2005.
- [49] D. H. Xiang, and D. X. Zhou, Classification with Gaussians and convex loss, *J. Mach. Learn. Res.*, 10: 1447-1468, 2009.
- [50] Y. Ye, Interior Point Algorithms: Theory and Analysis, *Wiley*, New York, 1997.
- [51] Y. Ying and D. X. Zhou, Unregularized online learning algorithms with general loss functions, *Appl. Comput. Harmon. Anal.*, 42: 224-244, 2017.
- [52] T. Zhang, Statistical behavior and consistency of classification methods based on convex risk minimization, *Ann. Stat.*, 32: 56-85, 2004.
- [53] Y. C. Zhang, J. Duchi and M. Wainwright, Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates, *J. Mach. Learn. Res.*, 16: 3299-3340, 2015.
- [54] D. X. Zhou, and K. Jetter, Approximation with polynomial kernels and SVM classifiers, *Adv. Comput. Math.*, 25: 323-344, 2006.
- [55] D. X. Zhou, Deep distributed convolutional neural networks: Universality, *Anal. Appl.*, 16: 895-919, 2018.
- [56] Z. H. Zhou, N. V. Chawla, Y. Jin and G. J. Williams, Big data opportunities and challenges: discussions from data analytics perspectives, *IEEE Comput. Intel. Magaz.*, 9: 62-74, 2014.