Optim Lett
DOI 10.1007/s11590-016-1026-4

ORIGINAL PAPER

Stochastic subgradient descent method for large-scale robust chance-constrained support vector machines

Ximing Wang[1] · Neng Fan[2] · Panos M. Pardalos[1]

Received: 30 June 2015 / Accepted: 29 February 2016

Abstract Robust chance-constrained Support Vector Machines (SVM) with second-order moment information can be reformulated into equivalent and tractable Semidefinite Programming (SDP) and Second Order Cone Programming (SOCP) models. However, practical applications involve processing large-scale data sets. For the reformulated SDP and SOCP models, existed solvers by primal-dual interior method do not have enough computational efficiency. This paper studies the stochastic subgradient descent method and algorithms to solve robust chance-constrained SVM on large-scale data sets. Numerical experiments are performed to show the efficiency of the proposed approaches. The result of this paper breaks the computational limitation and expands the application of robust chance-constrained SVM.

Keywords Support vector machines · Robust chance constraints · Primal-dual interior method · Stochastic subgradient descent method · Large-scale data

✉ Ximing Wang
x.wang@ufl.edu

Neng Fan
nfan@email.arizona.edu

Panos M. Pardalos
pardalos@ufl.edu

[1] Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL 32611, USA

[2] Department of Systems and Industrial Engineering, University of Arizona, Tucson, AZ 85721, USA

Published online: 15 March 2016

1 Introduction

Machine learning is the exploration of models and algorithms to learn from and make predictions based on data. Classified into supervised and unsupervised learning depending on the nature of data, most machine learning techniques are optimization problems. In supervised learning, the data has both features as the attributes and labels describing the class, and thus the learning process is to find the general rule that maps the data features to their labels. For unsupervised learning, data labels are not given, and the features are the only input to discover hidden patterns or structures in data.

As one of the well-known supervised learning algorithms, Support Vector Machines (SVM) is gaining more and more attention. It was proposed by Vapnik [22,23] as a maximum-margin classifier. Tutorials on SVM could refer to [1,2,8,9]. It has wide applications in many fields during recent years and also many algorithmic and modeling variations [21].

Basic SVM models are dealing with the situation where the exact values of the data points are known. When uncertainties exist in data, robust SVM, dealing with the worst-case scenario for data with supporting sets, and chance-constrained SVM, ensuring the small probability of misclassification for the uncertain data, are proposed in literatures, as reviewed in [25].

Recently, robust chance-constrained SVM model was proposed to have benefits of both robust and chance-constrained SVM [3,4,17,24]. For data with uncertainties with second-order moment information, it was reformulated into equivalent Semidefinite Programming (SDP) and Second Order Cone Programming (SOCP) models [24]. The SDP and SOCP reformulation models can be solved by conic linear programming solvers, such as SeDuMi [18] via primal-dual interior method [14,19,20]. However, practical problems require processing large-scale data sets, while the existed solvers do not have enough computational efficiency.

Different solving methods have been proposed for large-scale linear SVM. The dual coordinate descent method [12] considers the dual problem of the soft-margin linear SVM and can obtain an ϵ-accuracy solution for the dual problem in $O(\log(1/\epsilon))$ iterations with the cost per iteration $O(m\bar{n})$, where m is the number of training points and $\bar{n}$ is the average number of nonzero elements per feature. Stochastic subgradient descent method [6,26] considers the primal soft-margin linear SVM problem. As an application of stochastic subgradient descent method, Pegasos [16] can obtain an ϵ-accuracy solution for the primal problem in $\tilde{O}(1/\epsilon)$ iterations with the cost per iteration $O(n)$, where n is the feature dimension. To the best of our knowledge, no numerical methods have been proposed so far to solve robust chance-constrained SVM and its reformulations.

Depending on the specific structure of the reformulations for robust chance-constrained SVM, this paper proposes a method based on stochastic subgradient descent for solving robust chance-constrained SVM on large-scale data sets. We also perform numerical experiments to show the efficiency of the proposed approaches. The computational limitation has blocked the application of robust chance-constrained SVM. The result of this paper breaks the limitation and expands robust chance-constrained SVM to large-scale data sets.

The remainder of this paper is as follows. Section 2 introduces the robust chance-constrained SVM and its SDP and SOCP reformulations. Section 3 presents the stochastic subgradient descent method to solve large-scale linear SVM and its application Pegasos. The followed Sects. 4 and 5 are core sections of this paper. Section 4 proposes the approach to solve robust chance-constrained SVM on large-scale data sets. Section 5 contains the numerical experiments of the proposed method. Finally Sect. 6 concludes this paper.

2 Robust chance-constrained SVM and reformulations

For a two-class dataset of m data points $\{\mathbf{x}_i, y_i\}_{i=1}^m$ with n-dimensional features $\mathbf{x}_i \in \mathbb{R}^n$ and respective class labels $y_i \in \{+1, -1\}$, if they are linearly separable, there exists a hyperplane $\mathbf{w}^\top \mathbf{x} + b = 0$ to separate the two classes and the corresponding classification rule is based on $\text{sign}(\mathbf{w}^\top \mathbf{x} + b)$. SVM constructs maximum-margin classifiers [22,23] such that the distance between the hyperplane and the support vectors is maximized. When the two classes are not linearly separable, soft-margin SVM introduces non-negative slack variables ξ_i to allow mislabeled samples, and ξ_i measures the distance of within-margin or misclassified data $\mathbf{x}_i$ to the margin line with the correct label with the expression $\xi_i = \max\{0, 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b)\}$. When $0 < \xi_i < 1$, the data is within margine but correctly classified; when $\xi_i > 1$, the data is misclassified. The optimization is a trade off between a large margin and a small error penalty. The soft margin SVM formulation [10] is:

$$(\text{SVM} - \text{SoftMargin})$$

$$\min_{\mathbf{w},b,\xi_i} \quad \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^m \xi_i \tag{1a}$$

$$\text{s.t.} \quad y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, m \tag{1b}$$

where C is the trade-off parameter.

When uncertainties exist in the data points, suppose there are m training data points in $\mathbb{R}^n$, use $\tilde{\mathbf{x}}_i = [\tilde{x}_{i1}, \ldots, \tilde{x}_{in}]^\top \in \mathbb{R}^n$, $i = 1, \ldots, m$ to denote the uncertain training data points and $y_i \in \{+1, -1\}$, $i = 1, \ldots, m$ to denote the respective class labels. The chance-constrained program is to ensure the small probability of misclassification for the uncertain data. In practice, the exact probability distribution of the random variables are often unknown and hard to obtain. Only some properties of the distribution could be acquired, such as the first and second moments. To deal with the uncertainty in probability distribution, the distributionally robust or ambiguous chance constraint is developed and adopted to represent a conservative approximation of the original problem. Let $\mathcal{P}$ be the set of all probability distributions that have the known properties of $\mathbb{P}$, then the distributionally robust chance-constrained SVM formulation is [3,4,17]:

$$(\text{SVM} - \text{RCCP})$$

$$\min_{\mathbf{w},b,\xi_i} \quad \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^m \xi_i \tag{2a}$$

$$\text{s.t.} \quad \sup_{\mathbb{P}\in\mathcal{P}} \mathbb{P}\Big\{y_i(\mathbf{w}^\top \tilde{\mathbf{x}}_i + b) \le 1 - \xi_i\Big\} \le \varepsilon, \quad \xi_i \ge 0, \quad i = 1, \ldots, m \tag{2b}$$

where $0 < \varepsilon < 1$ is a prameter given by the user and close to 0, $\mathbb{P}\{\cdot\}$ is the probability distribution. This model ensures an upper bound on the misclassification probability over all distributions in $\mathcal{P}$.

The chance constraints are typically non-convex. As infinite number of distributions could have the known properties, the robust chance-constrained SVM requires efficient transformations of the chance constraints to make the problem solvable. Previous research showed that when second-order moment information, the mean vector $\boldsymbol{\mu}_i = \mathbf{E}[\tilde{\mathbf{x}}_i] \in \mathbb{R}^n$ and the covariance matrix $\boldsymbol{\Sigma}_i = \mathbf{E}\big[(\tilde{\mathbf{x}}_i - \boldsymbol{\mu}_i)(\tilde{\mathbf{x}}_i - \boldsymbol{\mu}_i)^\top\big] \in \mathbb{S}^n$ of the random variable $\tilde{\mathbf{x}}_i$ are known, robust chance-constrained SVM can be reformulated into equivalent SDP and SOCP models [24]. The SDP model is as follows

$$(\text{SVM} - \text{SDP})$$

$$\min_{\mathbf{w}, b, \xi_i, \mathbf{N}_i, \alpha_i} \quad \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{m} \xi_i \tag{3a}$$

$$\text{s.t.} \;\; \alpha_i - \frac{1}{\varepsilon}\text{Trace}(\Omega_i \mathbf{N}_i) \ge 0, \;\; \xi_i \ge 0 \tag{3b}$$

$$\mathbf{N}_i \succeq 0, \;\; \mathbf{N}_i + \begin{bmatrix} 0 & \frac{1}{2} y_i \mathbf{w} \\ \frac{1}{2} y_i \mathbf{w}^\top & y_i b + \xi_i - 1 - \alpha_i \end{bmatrix} \succeq 0 \tag{3c}$$

where $\Omega_i = \begin{bmatrix} \boldsymbol{\Sigma}_i + \boldsymbol{\mu}_i \boldsymbol{\mu}_i^\top & \boldsymbol{\mu}_i \\ \boldsymbol{\mu}_i^\top & 1 \end{bmatrix}$; and for matrix $\mathbf{A}$, $\mathbf{A} \succeq 0$ means $\mathbf{A}$ is positive semi-definite.

The equivalent SOCP model is as follows

$$(\text{SVM} - \text{SOCP})$$

$$\min_{\mathbf{w}, b, \xi_i} \quad \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{m} \xi_i \tag{4a}$$

$$\text{s.t.} \quad y_i(\mathbf{w}^\top \boldsymbol{\mu}_i + b) \ge 1 - \xi_i + \sqrt{\frac{1-\varepsilon}{\varepsilon}} \left\| \boldsymbol{\Sigma}_i^{\frac{1}{2}} \mathbf{w} \right\|_2 \tag{4b}$$

$$\xi_i \ge 0, \;\; i = 1, \ldots, m \tag{4c}$$

The SOCP model also has a nice geometric interpretation that for each point $\mathbf{x}_i$, it is no longer a single point, but an ellipsoid centered at $\boldsymbol{\mu}_i$, and shaped with the covariance matrix $\sqrt{\frac{1-\varepsilon}{\varepsilon}}\boldsymbol{\Sigma}_i^{\frac{1}{2}}$:

$$\mathscr{E}\Big(\boldsymbol{\mu}_i, \sqrt{\frac{1-\varepsilon}{\varepsilon}}\boldsymbol{\Sigma}_i^{\frac{1}{2}}\Big) = \Big\{\mathbf{x} = \boldsymbol{\mu}_i + \sqrt{\frac{1-\varepsilon}{\varepsilon}}\boldsymbol{\Sigma}_i^{\frac{1}{2}}\mathbf{a} \;:\; ||\mathbf{a}||_2 \le 1\Big\}$$

The SOCP constraint (4b) is satisfied if and only if

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \forall \mathbf{x}_i \in \mathscr{E}\left(\boldsymbol{\mu}_i, \sqrt{\frac{1-\varepsilon}{\varepsilon}}\boldsymbol{\Sigma}_i^{\frac{1}{2}}\right)$$

Therefore, this constraint is defining an uncertainty set $\mathscr{E}\left(\boldsymbol{\mu}_i, \sqrt{\frac{1-\varepsilon}{\varepsilon}}\boldsymbol{\Sigma}_i^{\frac{1}{2}}\right)$ for each uncertain training data point $\mathbf{x}_i$. If all the points in the uncertainty set satisfy $y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i$, then the chance-constraint is guaranteed to be satisfied. This transforms the RCCP into a robust optimization problem over the uncertainty set.

The SDP and SOCP models can be solved by conic linear programming solvers like SeDuMi [18] using primal-dual interior method [14,19,20]. However, it is far from sufficient to obtain solutions for large-scale data. Even though basic soft-margin linear SVM model is a convex quadratic programming model and much simpler and easier to solve than SDP and SOCP models, quadratic solvers are still facing difficulty for large-scale data. Different solving methods have been proposed for large-scale linear SVM including dual coordinate descent method [12] and stochastic subgradient descent method [6,16,26]. According to the structure of robust chance-constrained SVM reformulations, stochastic subgradient descent method is chosen to be the foundation for solving robust chance-constrained SVM with large-scale data.

3 Stochastic subgradient descent method

For soft-margin SVM model (SVM-SoftMargin), the non-negative slack variables $\xi_i = \max\{0, 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b)\}$. Then the model can be expressed in this particular form:

$$\min_{\mathbf{w},b} \quad f(\mathbf{w}, b) = \frac{1}{2}\mathbf{w}^\top \mathbf{w} + C\sum_{i=1}^{m} \max\{0, 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b)\} \tag{5}$$

The second term in (5) is also called penalty function. The function $L(\mathbf{x}_i, y_i) = \max\{0, 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b)\}$ decreases linearly for $y_i(\mathbf{w}^\top \mathbf{x}_i + b) \leq 1$ and then remains 0. This is called hinge function, and its value is called the hinge loss. When $y_i(\mathbf{w}^\top \mathbf{x}_i + b)$ is 1 or more, the value of L is 0. L increases linearly as $y_i(\mathbf{w}^\top \mathbf{x}_i + b)$ decreases for smaller values.

The soft-margin SVM is a quadratic program and can be solved by classic quadratic programming approaches. But for large-scale data, subgradient descent method has advantages as shown in [15]. To minimize $f(\mathbf{w}, b)$, the subgradient of the equation with respect to $\mathbf{w}$ and b are computed, and then the current $\mathbf{w}$ and b are moved in the opposite direction of the subgradient. Adding an extra dimension to the feature vector of each data point with value 1, b can be made part of the weight vector $\mathbf{w}$.

A constant η_t is chosen as the fraction of the subgradient that to be moved in each iteration. The subgradient descent iteration is:

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \eta_t \nabla_{\mathbf{w}} f \tag{6}$$

The subgradient of the loss function $L(\mathbf{x}_i, y_i)$ with respect to $\mathbf{w}$ is discontinuous. When $y_i(\mathbf{w}^\top \mathbf{x}_i + b) < 1$, it is $-y_i\mathbf{x}_i$; when $y_i(\mathbf{w}^\top \mathbf{x}_i + b) > 1$, it is 0. Then the subgradient of f with respect to $\mathbf{w}$ can be computed as

$$\begin{aligned} \nabla_{\mathbf{w}} f &= \mathbf{w} + C \sum_{i=1}^{m} \nabla_{\mathbf{w}} L(\mathbf{x}_i, y_i) \\ &= \mathbf{w} + C \sum_{i=1}^{m} \begin{cases} 0, & \text{if } y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 \\ -y_i\mathbf{x}_i, & \text{otherwise} \end{cases} \end{aligned} \tag{7}$$

The subgradient descent method can be summarized as follows:

```
Batch Subgradient Descent Method
Iterate until convergence:
    Evaluate:  $\nabla_{\mathbf{w}} f = \mathbf{w} + C \sum_{i=1}^{m} \nabla_{\mathbf{w}} L(\mathbf{x}_i, y_i)$
    Update:    $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \eta_t \nabla_{\mathbf{w}} f$
```

This is called batch subgradient descent because all the training points are considered as a batch in each iteration. The problem of the batch subgradient descent method is that to compute $\nabla_{\mathbf{w}} f$, it needs to go over all the m training data points. When the data size is large, it can be too time-consuming to visit every training point and often iterates many times before convergence.

The stochastic subgradient descent [6,26], on the other hand, considers one training point at a time and adjusts the current solution in the direction evaluated by the only training point:

$$\nabla_{\mathbf{w}} f_t = \mathbf{w} + Cm \nabla_{\mathbf{w}} L(\mathbf{x}_{i_t}, y_{i_t}) \tag{8}$$

The training point $(\mathbf{x}_{i_t}, y_{i_t})$ can be selected randomly or according to some fixed strategy.

The stochastic subgradient descent method can be summarized as follows:

```
Stochastic Subgradient Descent Method
Iterate until convergence:
    Evaluate:  $\nabla_{\mathbf{w}} f_t = \mathbf{w} + Cm \nabla_{\mathbf{w}} L(\mathbf{x}_{i_t}, y_{i_t})$
    Update:    $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \eta_t \nabla_{\mathbf{w}} f_t$
```

The batch subgradient descent method improves the value of the objective function at every step. The stochastic subgradient descent method improves the value in a noisy way since it only considers one point at each iteration. The batch subgradient descent method takes fewer iterations to converge, but in each iteration, it takes much longer to compute. In practice, the stochastic subgradient descent method is much faster.

For the batch subgradient descent method, when the initial estimate $\mathbf{w}_0$ is close enough to the optimum, and when the step size controller η_t is sufficiently small, this method achieves linear convergence under sufficient regularity assumptions [11]. For an ϵ-accurate solution $\hat{\mathbf{w}}$ with $f(\hat{\mathbf{w}}) \leq \min_{\mathbf{w}} f(\mathbf{w}) + \epsilon$ and iterations t, it is showed that $-\log \epsilon \sim t$ [6].

Table 1 Computational cost of batch subgradient descent method and stochastic subgradient descent method

	BSGD	SSGD
Time per iteration	mn	n
Iterations to ϵ-accuracy	$\log(1/\epsilon)$	$1/\epsilon$
Time to ϵ-accuracy	$mn\log(1/\epsilon)$	n/ϵ

For the stochastic subgradient descent method, the convergence often requires that η_t decreases and satisfies $\sum_t \eta_t^2 < \infty$ and $\sum_t \eta_t = \infty$ [7]. The convergence speed of the stochastic subgradient descent method is closely related to the step size controller η_t since the stochastic subgradient descent method uses a noisy approximation of the true subgradient. Murata [13] showed that the best convergence speed is achieved with $\eta_t \sim t^{-1}$ under sufficient regularity conditions and the expectation of the residual error decreases with similar speed $\mathbf{E}[\epsilon] \sim t^{-1}$. In [5], the authors explicitly expressed η_t as $1/(t + t_0)$.

The computational cost of the batch subgradient descent method and stochastic subgradient descent method are summarized in Table 1. When the data size m is large, the stochastic subgradient descent method performs asymptotically better than the batch subgradient descent method.

As an application of the stochastic subgradient descent method, Pegasos (Primal Estimated sub-GrAdient SOlver for SVM) [16] studied the SVM problem in this form:

$$\min_{\mathbf{w}} \quad f(\mathbf{w}) = \frac{\lambda}{2}\mathbf{w}^\top \mathbf{w} + \frac{1}{m}\sum_{i=1}^{m} \max\{0, 1 - y_i \mathbf{w}^\top \mathbf{x}_i\} \tag{9}$$

The initial solution $\mathbf{w}_1$ is set to be the zero vector. On iteration t, a random training point $(\mathbf{x}_{i_t}, y_{i_t})$ is chosen uniformly with $i_t \in \{1, \ldots, m\}$. The objective function is approximated with the training point $(\mathbf{x}_{i_t}, y_{i_t})$:

$$f_t(\mathbf{w}) = \frac{\lambda}{2}\mathbf{w}^\top \mathbf{w} + \max\{0, 1 - y_{i_t} \mathbf{w}^\top \mathbf{x}_{i_t}\} \tag{10}$$

And the subgradient is:

$$\nabla_{\mathbf{w}} f_t = \lambda \mathbf{w}_t - 1_{[y_{i_t} \mathbf{w}_t^\top \mathbf{x}_{i_t} < 1]} y_{i_t} \mathbf{x}_{i_t} \tag{11}$$

where $1_{[y_{i_t} \mathbf{w}^\top \mathbf{x}_{i_t} < 1]}$ is the indicator function and the whole term has similar meaning with $\nabla_{\mathbf{w}} L(\mathbf{x}_{i_t}, y_{i_t})$. The step size is set to be $\eta_t = 1/(\lambda t)$ for $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \eta_t \nabla_{\mathbf{w}} f_t$. Then the update could be written as:

$$\mathbf{w}_{t+1} \leftarrow \left(1 - \frac{1}{t}\right)\mathbf{w}_t + \eta_t 1_{[y_{i_t} \mathbf{w}_t^\top \mathbf{x}_{i_t} < 1]} y_{i_t} \mathbf{x}_{i_t} \tag{12}$$

Pegasos can obtain an ϵ-accuracy solution in $\tilde{Q}(1/\epsilon)$ iterations with high probability over the choice of the random training points. In each iteration, it involves only one inner product between $\mathbf{w}$ and $\mathbf{x}$. For n-dimensional data, the overall runtime required to obtain an ϵ-accuracy solution is $\tilde{Q}(n/\epsilon)$, with no direct dependency on the number m of training points, and suited for large-scale data sets.

4 Large-scale robust chance-constrained SVM solution approach

As analyzed in Sect. 2, the robust chance-constrained SVM model can be transformed into equivalent SDP and SOCP models. Traditional SDP and SOCP solvers like SeDuMi utilize the primal-dual interior method and the complexity grows intensively with data size. For large-scale robust chance-constrained SVM, the tools used to solve large-scale SVM are required instead of the normal conic programming solvers.

Since the SDP and SOCP reformulations are equivalent, in this section for solving large-scale robust chance-constrained SVM problems, the (SVM-SOCP) model is the basic model analyzed. Write the model in the form similar to subgradient descent method:

$$\begin{aligned}\min_{\mathbf{w},b}\quad f(\mathbf{w},b) &= \frac{1}{2}\mathbf{w}^\top\mathbf{w}\\ &\quad + C\sum_{i=1}^{m}\max\left\{0,\ 1-y_i(\mathbf{w}^\top\boldsymbol{\mu}_i+b)+\sqrt{\frac{1-\varepsilon}{\varepsilon}\mathbf{w}^\top\boldsymbol{\Sigma}_i\mathbf{w}}\right\} \qquad (13)\end{aligned}$$

We need to obtain the subgradient $\nabla_{\mathbf{w}} f$. For term $g(\mathbf{w})=\sqrt{\frac{1-\varepsilon}{\varepsilon}\mathbf{w}^\top\boldsymbol{\Sigma}_i\mathbf{w}}$, as $\frac{\partial\mathbf{w}^\top\boldsymbol{\Sigma}_i\mathbf{w}}{\partial\mathbf{w}}=(\boldsymbol{\Sigma}_i+\boldsymbol{\Sigma}_i^\top)\mathbf{w}=2\boldsymbol{\Sigma}_i\mathbf{w}$, according to the chain rule, we have

$$\nabla_{\mathbf{w}} g=\sqrt{\frac{1-\varepsilon}{\varepsilon}}\frac{2\boldsymbol{\Sigma}_i\mathbf{w}}{2\sqrt{\mathbf{w}^\top\boldsymbol{\Sigma}_i\mathbf{w}}}=\sqrt{\frac{1-\varepsilon}{\varepsilon}}\frac{\boldsymbol{\Sigma}_i\mathbf{w}}{||\boldsymbol{\Sigma}_i^{\frac{1}{2}}\mathbf{w}||_2} \qquad (14)$$

Then for $L(\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i,y_i)=\max\left\{0,\ 1-y_i(\mathbf{w}^\top\boldsymbol{\mu}_i+b)+\sqrt{\frac{1-\varepsilon}{\varepsilon}\mathbf{w}^\top\boldsymbol{\Sigma}_i\mathbf{w}}\right\}$, the subgradient

$$\nabla_{\mathbf{w}} L(\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i,y_i)=\begin{cases}0, & \text{if } y_i(\mathbf{w}^\top\boldsymbol{\mu}_i+b)\geq 1+\sqrt{\frac{1-\varepsilon}{\varepsilon}}||\boldsymbol{\Sigma}_i^{\frac{1}{2}}\mathbf{w}||_2\\ -y_i\boldsymbol{\mu}_i+\sqrt{\frac{1-\varepsilon}{\varepsilon}}\frac{\boldsymbol{\Sigma}_i\mathbf{w}}{||\boldsymbol{\Sigma}_i^{\frac{1}{2}}\mathbf{w}||_2}, & \text{otherwise}\end{cases} \qquad (15)$$

For stochastic subgradient descent method and its application Pegasos, in each iteration, only one randomly selected training point $(\mathbf{x}_{i_t}, y_{i_t})$ is considered. Comparing the Pegasos model and the SVM model in our expression, for $\mathbf{w}_{t+1}\leftarrow\mathbf{w}_t-\eta_t\nabla_{\mathbf{w}} f_t$, the subgradient considered is $1/(Cm)$ of the original f:

$$\nabla_{\mathbf{w}} f_t=\frac{1}{Cm}\mathbf{w}+\nabla_{\mathbf{w}} L(\boldsymbol{\mu}_{i_t},\boldsymbol{\Sigma}_{i_t},y_{i_t}) \qquad (16)$$

And the step size $\eta_t = 1/(\lambda t)$ where $\lambda = 1/(Cm)$ according to the relationship between Pegasos model and the standard SVM model, therefore, $\eta_t = Cm/t$.

The update is then:

$$\mathbf{w}_{t+1} \leftarrow \left(1-\frac{1}{t}\right)\mathbf{w}_t + \eta_t 1_{\left[y_{i_t}(\mathbf{w}_t^\top \boldsymbol{\mu}_{i_t}+b_t)<1+\sqrt{\frac{1-\varepsilon}{\varepsilon}}||\boldsymbol{\Sigma}_{i_t}^{\frac{1}{2}}\mathbf{w}_t||_2\right]}\left(y_{i_t}\boldsymbol{\mu}_{i_t}-\sqrt{\frac{1-\varepsilon}{\varepsilon}}\frac{\boldsymbol{\Sigma}_{i_t}\mathbf{w}_t}{||\boldsymbol{\Sigma}_{i_t}^{\frac{1}{2}}\mathbf{w}_t||_2}\right) \tag{17a}$$

$$b_{t+1} \leftarrow b_t + \eta_t 1_{\left[y_{i_t}(\mathbf{w}_t^\top \boldsymbol{\mu}_{i_t}+b_t)<1+\sqrt{\frac{1-\varepsilon}{\varepsilon}}||\boldsymbol{\Sigma}_{i_t}^{\frac{1}{2}}\mathbf{w}_t||_2\right]} y_{i_t} \tag{17b}$$

The large-scale robust chance-constrained SVM solving method with T as the predetermined number of iterations and the output as $(\mathbf{w}_{T+1}, b_{T+1})$ can be summarized as follows:

```
SVM-RCCP SSGD Method
Initialize $\mathbf{w}_1 = \mathbf{0}, b_1 = 0$
For $t = 1, 2, \ldots, T$
    Choose $i_t \in \{1, \ldots, m\}$ uniformly at random
    Set $\eta_t = Cm/t$
    If $y_{i_t}(\mathbf{w}_t^\top \boldsymbol{\mu}_{i_t} + b_t) < 1 + \sqrt{\frac{1-\varepsilon}{\varepsilon}}||\boldsymbol{\Sigma}_{i_t}^{\frac{1}{2}}\mathbf{w}_t||_2$, then
        Set $\mathbf{w}_{t+1} \leftarrow (1-1/t)\mathbf{w}_t + \eta_t\left(y_{i_t}\boldsymbol{\mu}_{i_t} - \sqrt{\frac{1-\varepsilon}{\varepsilon}}\frac{\boldsymbol{\Sigma}_{i_t}\mathbf{w}_t}{||\boldsymbol{\Sigma}_{i_t}^{\frac{1}{2}}\mathbf{w}_t||_2}\right)$
            $b_{t+1} \leftarrow b_t + \eta_t y_{i_t}$
    Else $\left(\text{i.e., } y_{i_t}(\mathbf{w}_t^\top \boldsymbol{\mu}_{i_t} + b_t) \geq 1 + \sqrt{\frac{1-\varepsilon}{\varepsilon}}||\boldsymbol{\Sigma}_{i_t}^{\frac{1}{2}}\mathbf{w}_t||_2\right)$
        Set $\mathbf{w}_{t+1} \leftarrow (1-1/t)\mathbf{w}_t$
            $b_{t+1} \leftarrow b_t$
```

It needs to point out that in the literatures of large-scale SVM [6,12,16,26], b is either not considered or included into $\mathbf{w}$, and there is no nonlinear term $\sqrt{\frac{1-\varepsilon}{\varepsilon}}||\boldsymbol{\Sigma}_i^{\frac{1}{2}}\mathbf{w}||_2$ in the constraint. The convergence rate of the original stochastic subgradient descent method is based on convex functions. Adding b and the nonlinear term actually makes the function not convex any more. Nevertheless, the numerical experiments show convincing results for the potential of the above proposed method to process large-scale data sets.

5 Numerical experiments

In numerical experiments, first, the Wisconsin breast cancer data and Ionosphere data from UCI datasets are used. Wisconsin breast cancer data has 683 samples with 9-dimensional features. Ionosphere data has 351 samples with 34-dimensional features. The mean vector $\boldsymbol{\mu}_i$ used in the experiments is the original value of the data $\mathbf{x}_i$. The covariance matrix of the training data in $+1$ class and -1 class are calculated respectively. Then the covariance matrix $\mathbf{\Sigma}_i$ of each training sample $\mathbf{x}_i$ is set to be 1/100 of the covariance matrix of its corresponding class. The robust chance-constraint probability ε is set to be 0.1 in all experiments.

When using the proposed SVM-RCCP SSGD method, the sampling procedure is chosen to be sampling without replacement and new permutation is generated every epoch. This means that, a random permutation is chosen over the training points, then the iterations are performed in accordance to the selected order. After going over all the training data, i.e. completing one epoch, a new random permutation is generated, and the following iterations in this epoch are performed in the new order. In the experiments, the number of iterations is set to be $2m$, $10m$, and $50m$ respectively, where m is the number of training samples. So the iterations goes 2 epochs, 10 epochs, and 50 epochs respectively over the training data. As comparison, SeDuMi is used to solve the SOCP reformulation of robust chance-constrained SVM directly.

The results for Wisconsin breast cancer data over 20 runs with random partitions are shown in Table 2. It can be seen from the results that SVM-RCCP SSGD method runs much faster than SeDuMi while maintaining acceptable Test Set Accuracy (TSA). For $2m$ iterations in SSGD, the average TSA is the smallest and the standard deviation is largest. This means that the results from $2m$ iterations are not reaching good TSA and not stable as well. For $10m$ and $50m$, the TSA are comparable, while $50m$ has less standard deviation. But the running time for $50m$ is almost 5 times of the $10m$. Therefore, for this data, $10m$ is a better choice to obtain both high TSA and short running time. The running time of SSGD $10m$ is more than 20 times less than the SeDuMi while the TSA is almost the same.

The results for Ionosphere data over 20 runs with random partitions are shown in Table 3. SSGD is still getting similar TSA results with SeDuMi while the running time is much less.

Besides these datasets, a larger dataset, MAGIC Gamma Telescope data from UCI datasets is experimented. MAGIC Gamma Telescope data has 19,020 samples with 10-dimensional features. Among these samples, 12332 belong to class gamma (signal), 6688 belong to class hadron (background). Similar uncertainty and sampling settings are used for this data when doing the experiments. The results over 20 runs with random partitions are shown in Table 4.

Table 2 Wisconsin breast cancer data using SeDuMi and SVM-RCCP SSGD

	SeDuMi	SSGD $2m$	SSGD $10m$	SSGD $50m$
20 % training, 80 % test				
Running time	0.857 ± 0.063	0.013 ± 0.002	0.034 ± 0.001	0.146 ± 0.002
Test set accuracy (%)	96.16 ± 0.98	93.90 ± 4.23	96.09 ± 1.27	96.08 ± 0.64
80 % training, 20 % Test				
Running time	2.542 ± 0.094	0.029 ± 0.001	0.117 ± 0.001	0.564 ± 0.007
Test set accuracy (%)	96.72 ± 1.12	95.84 ± 2.12	96.68 ± 1.31	96.70 ± 1.28

Table 3 Ionosphere data using SeDuMi and SVM-RCCP SSGD

	SeDuMi	SSGD $2m$	SSGD $10m$	SSGD $50m$
20 % training, 80 % test				
Running time	0.728 ± 0.053	0.010 ± 0.000	0.024 ± 0.001	0.099 ± 0.003
Test set accuracy (%)	83.56 ± 2.50	76.89 ± 7.96	81.09 ± 4.63	82.85 ± 2.93
80 % training, 20 % test				
Running time	1.961 ± 0.044	0.022 ± 0.001	0.083 ± 0.002	0.394 ± 0.013
Test set accuracy (%)	85.93 ± 3.48	80.71 ± 5.07	85.57 ± 4.14	85.64 ± 4.33

Table 4 MAGIC gamma telescope data using SeDuMi and SVM-RCCP SSGD

	SeDuMi	SSGD $2m$	SSGD $10m$	SSGD $50m$
20 % training, 80 % test				
Running time	41.198 ± 4.558	0.215 ± 0.002	1.031 ± 0.011	5.011 ± 0.115
Test set accuracy (%)	76.80 ± 0.52	64.29 ± 6.61	72.09 ± 4.09	74.35 ± 3.87
80 % training, 20 % test				
Running time	–	0.850 ± 0.028	4.061 ± 0.075	20.105 ± 0.654
Test set accuracy (%)	–	67.47 ± 6.49	72.14 ± 4.71	74.68 ± 3.39

For this data, when choosing 80 % of the samples as training points, SeDuMi cannot get solutions of the robust chance-constrained SVM problem because of the data size. SVM-RCCP SSGD method still obtain reasonable TSA results in acceptable time. Our proposed method has the potential for processing large-scale data sets.

6 Conclusions

Practical problems in data sciences require to process big data. For the SDP and SOCP reformulations of robust chance-constrained SVM, existing solvers like SeDuMi do not have enough computational efficiency. Considering data with uncertainties, this paper adapts the stochastic subgradient descent method to solve robust chance-constrained SVM on large-scale data sets. Numerical experiments show the efficiency of the proposed approach.

For large-scale data, our proposed method has an advantage to allow the data residing on disk, while conic programming solvers normally require to keep all the data in memory. Because of the hidden nature of our proposed method, our method has the potential by parallel implementation to improve the efficiency. In the future, we believe the proposed algorithm with the robust chance-constrained SVM model can have broader applications in many areas.

Acknowledgments Research was partially supported by a DTRA grant and the Paul and Heidi Brown Preeminent Professorship in Industrial and Systems Engineering, University of Florida.

References

1. Abe, S.: Support vector machines for pattern classification. Springer (2010)
2. Ben-Hur, A., Weston, J.: A user's guide to support vector machines. In: Data mining techniques for the life sciences, Springer, pp 223–239 (2010)
3. Ben-Tal, A., Bhadra, S., Bhattacharyya, C., Nath, J.S.: Chance constrained uncertain classification via robust optimization. Math. Program. **127**(1), 145–173 (2011)
4. Bhattacharyya, C., Grate, L.R., Jordan, M.I., El Ghaoui, L., Mian, I.S.: Robust sparse hyperplane classifiers: application to uncertain molecular profiling data. J. Comput. Biol. **11**(6), 1073–1089 (2004)
5. Bordes, A., Bottou, L., Gallinari, P.: Sgd-qn: careful quasi-newton stochastic gradient descent. J. Mach. Learn. Res. **10**, 1737–1754 (2009)
6. Bottou, L.: Large-scale machine learning with stochastic gradient descent. In: Proceedings of COMPSTAT'2010. Springer, pp. 177–186 (2010)
7. Bousquet, O., Bottou, L.: The tradeoffs of large scale learning. In: Advances in neural information processing systems, pp. 161–168 (2008)
8. Burges, C.J.: A tutorial on support vector machines for pattern recognition. Data Min. Knowl. Discov. **2**(2), 121–167 (1998)
9. Chang, C.C., Lin, C.J.: Libsvm: a library for support vector machines. ACM Trans. Intel. Syst. Technol. (TIST) **2**(3), 27 (2011)
10. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)
11. Dennis Jr, J.E., Schnabel, R.B.: Numerical methods for unconstrained optimization and nonlinear equations, vol. 16. Siam (1996)
12. Hsieh, C.J., Chang, K.W., Lin, C.J., Keerthi, S.S., Sundararajan, S.: A dual coordinate descent method for large-scale linear svm. In: Proceedings of the 25th international conference on Machine learning. ACM, pp. 408–415 (2008)
13. Murata, N.: A Statistical Study of On-Line Learning. Online Learning and Neural Networks. Cambridge University Press, Cambridge (1998)
14. Nesterov, Y., Nemirovskii, A., Ye, Y.: Interior-point polynomial algorithms in convex programming, vol. 13. SIAM (1994)
15. Rajaraman, A., Ullman, J.D.: Mining of massive datasets. Cambridge University Press (2011)
16. Shalev-Shwartz, S., Singer, Y., Srebro, N., Cotter, A.: Pegasos: primal estimated sub-gradient solver for svm. Math. Program. **127**(1), 3–30 (2011)
17. Shivaswamy, P.K., Bhattacharyya, C., Smola, A.J.: Second order cone programming approaches for handling missing and uncertain data. J. Mach. Learn. Res. **7**, 1283–1314 (2006)
18. Sturm, J.F.: Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. Optim. Methods Softw. **11**(1–4), 625–653 (1999)
19. Sturm, J.F.: Implementation of interior point methods for mixed semidefinite and second order cone optimization problems. Optim. Methods Softw. **17**(6), 1105–1154 (2002)
20. Sturm, J.F., Zhang, S.: Symmetric primal-dual path-following algorithms for semidefinite programming. Appl. Num. Math. **29**(3), 301–315 (1999)
21. Tian, Y., Shi, Y., Liu, X.: Recent advances on support vector machines research. Technol. Econ. Develop. Econ. **18**(1), 5–33 (2012)
22. Vapnik, V.N.: Statistical Learning Theory. Wiley (1998)
23. Vapnik, V.N.: An overview of statistical learning theory. IEEE Trans. Neural Netw. **10**(5), 988–999 (1999)
24. Wang, X., Fan, N., Pardalos, P.M.: Robust chance-constrained support vector machines with second-order moment information. Ann. Oper. Res. (2015). doi:10.1007/s10479-015-2039-6
25. Wang, X., Pardalos, P.M.: A survey of support vector machines with uncertainties. Ann. Data Sci. **1**(3–4), 293–309 (2014)
26. Zhang, T.: Solving large scale linear prediction problems using stochastic gradient descent algorithms. In: Proceedings of the twenty-first international conference on Machine learning. ACM, pp. 116–123 (2004)