SVM via Saddle Point Optimization: New Bounds and Distributed Algorithms

Yifei Jin

Institute for Interdisciplinary Information Sciences
Tsinghua University, China

Joint work with:

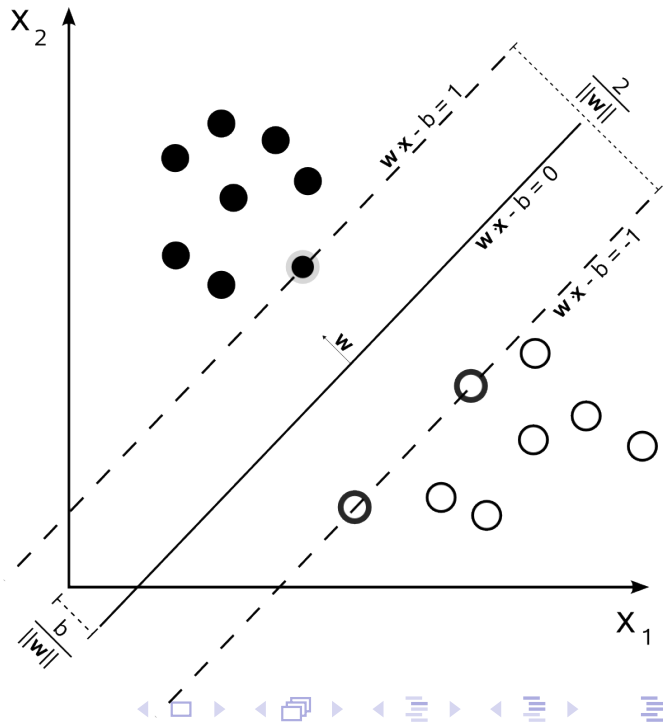
Lingxiao Huang
EPFL

Jian Li
Tsinghua University

Hard-margin SVM

- Given n points $x_i \in \mathbb{R}^d$ for $1 \leq i \leq n$, each x_i has a label $y_i \in \{+1, -1\}$.
- Hard-margin SVM for linearly separable cases: the goal is to find a hyperplane that separates two classes of points and the margin is maximized. [Boser et al. 1992, Cortes and Vapnik 1995]

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}\|w\|^2 \\ \text{s.t.} \quad & y_i(w^{\mathrm{T}} x_i - b) \geq 1, \quad \forall i \end{aligned} \tag{1}$$



SVM variants for non-separable cases

l_2-SVM , C-SVM and ν-SVM.

- The main difference among these variants: use different penalty loss functions for the misclassified points.
- l_2-SVM, as the name implied, uses the l_2 penalty loss.
- C-SVM uses the l_1-loss with penalty coefficient $C \in [0, \infty)$ [Zhu et al. 2004]
- ν-SVM reformulates C-SVM through taking a new regularization parameter $\nu \in (0, 1]$ [Schölkopf et al. 2000]

ν-SVM

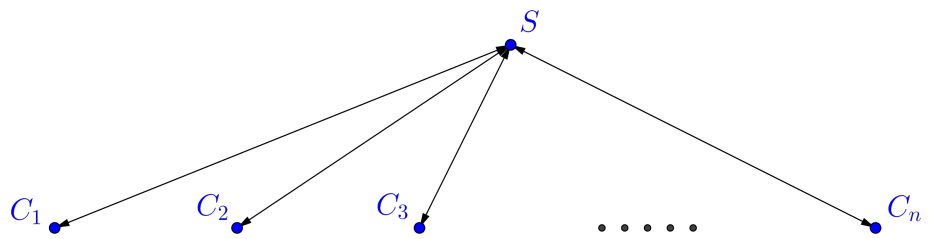
Given n points $x_i \in \mathbb{R}^d$ for $1 \leq i \leq n$, each x_i has a label $y_i \in \{+1, -1\}$.
ν-SVM is the quadratic programming as follows. [Schölkopf et al. 2000]

$$\begin{array}{ll} \min\limits_{w,b,\rho,\delta} & \frac{1}{2}\|w\|^2 - \rho + \frac{\nu}{2}\sum_i \delta_i \\ \text{s.t.} & y_i(w^{\mathrm{T}} x_i - b) \geq \rho - \delta_i, \delta_i \geq 0, \quad \forall i \end{array} \tag{2}$$

- The parameter ν is an upper bound of the fraction of margin errors and a lower bound of the fraction of support vectors.
- $\sum_i \delta_i$ is the sum of slacks.
- If all $\delta_i = 0$, the constraints simply state that the two classes are separated by the margin $2\rho/\|w\|$.

Distributed Settings

- The *server* and *clients* model. Denote the server by S. Let $\mathcal{C}$ be the set of clients and $|\mathcal{C}| = k$.
- Each clients holds a subset of points $\mathcal{P}$.
- Only server can communicate with client.
- The goal is to solve SVM over $\mathcal{P}$ in server.
- Performance measurements of distributed algorithms is the communications cost.



State-of-the-art Algorithms

- Hard-margin SVM: $(1-\epsilon)$-approximation with $O(nd/\epsilon\beta^2)$ time by Gilbert algorithm [Gilbert 1966, Gärtner and Jaggi 2009], where β is the ratio of the minimum distance to the maximum one among the points.
- ν-SVM: $O(n^2 d)$ time algorithm by quadratic programming [Platt 1999, Joachims 1998]

Distributed Setting:

- For hard-margin SVM: Liu et al. proposed a distributed algorithm with $O(kd/\epsilon)$ communication cost. [Liu et al. 2016]
- Lower bound: the communication cost to achieve a $(1-\epsilon)$-approximation of the distributed SVM problem is at least $\Omega(k \min\{d, 1/\epsilon\})$ for any $\epsilon > 0$. [Liu et al. 2016]

Our Contributions

- $(1-\epsilon)$-approximations with running time $\tilde{O}(nd + n\sqrt{d/\epsilon})$ for both hard-margin SVM and ν-SVM.
  - Compared to Gilbert algorithm, our algorithm improves the running time by a factor of $\sqrt{d}/\sqrt{\epsilon}$ for hard-margin SVM.
  - To the best of our knowledge, it is the first nearly linear time algorithm for ν-SVM.
- $\tilde{O}(k(d + \sqrt{d/\epsilon}))$ communication cost for both ν-SVM and hard-margin SVM in distributed setting
  - It is almost optimal according to the lower bound.
  - For the hard-margin SVM, compared with the current best algorithm [Liu et al. 2016] with $O(kd/\epsilon)$ communication cost, our algorithm is more suitable when ϵ is small and d is large.
  - For ν-SVM, our algorithm is the first practical distributed algorithm.

Overview of the algorithms

Hard-Margin SVM:

- We regard hard-margin SVM as computing the polytope distance between two classes of points.
- Then we translate the problem to a saddle point optimization problem using the properties of the geometric structures (Lemma 2), and provide an algorithm to solve the saddle point optimization.

Overview of the algorithms

ν-SVM:

- It is known that ν-SVM is equivalent to computing the distance between two reduced polytopes [David et al. 1999, Bennett and Bredensteiner 2000]
- However, the number of vertices in the reduced polytopes may be exponentially large.
- In our framework, we only need to implicitly represent the reduced polytopes.
- Using the similar saddle point optimization framework, together with a nontrivial projection method, ν-SVM can be solved in nearly linear time.

Hard-margin SVM via Saddle Point Optimization
Step 1: Dual Form

The dual problem of hard-margin SVM is equivalent to finding the minimum distance between the two convex hulls of two classes of points when they are linearly separable [Bennett and Bredensteiner, 2000].

$$\begin{aligned} \min_{\eta,\xi} \quad & \frac{1}{2}\|A\eta - B\xi\|^2 \\ \text{s.t.} \quad & \|\eta\|_1 = 1, \|\xi\|_1 = 1. \quad \eta \geq 0, \xi \geq 0. \end{aligned} \tag{3}$$

where A and B are the matrices in which each column represents a vector of a point with label $+1$ or -1 respectively. We call the problem the C-Hull problem.

Step 2: Saddle Point Optimization Form

Lemma

Problem C-Hull (3) *is equivalent to the saddle point optimization* (4).

$$\text{OPT} = \max_{w} \min_{\eta \in \Delta_{n_1}, \xi \in \Delta_{n_2}} w^{\mathrm{T}} A\eta - w^{\mathrm{T}} B\xi - \frac{1}{2}\|w\|^2 \tag{4}$$

Note that $\phi(w, \eta, \xi)$ is only linear w.r.t. η and ξ.

Step 3: Strongly Convex Trick

- For faster algorithms, we hope that the objective function is strongly convex with respect to η and ξ
- We can add a small regularization term which ensures that the objective function is strongly convex. This is a commonly used approach in optimization (very similar to [Allen-Zhu et al. 2016]).

$$\max_{w} \min_{\eta \in \Delta_{n_1}, \xi \in \Delta_{n_2}} \quad w^{\mathrm{T}} A\eta - w^{\mathrm{T}} B\xi + \gamma H(\eta) + \gamma H(\xi) - \frac{1}{2}\|w\|^2, \tag{5}$$

where $\gamma = \epsilon\beta/2\log n$ and $H(u) := \sum_i u_i \log u_i$.

Why we can introduce regularization terms?

Saddle point optimizations (4) and (5) obtain almost the same results.

Lemma

Let (w^*, η^*, ξ^*) *and* $(w^\circ, \eta^\circ, \xi^\circ)$ *be the optimal solution of saddle point optimizations* (4) *and* (5) *respectively. Define* OPT *as in* (4)*. Define*

$$g(w) := \min_{\eta \in \Delta_{n_1}, \xi \in \Delta_{n_2}} w^{\mathrm{T}} A\eta - w^{\mathrm{T}} B\xi - \frac{1}{2}\|w\|^2.$$

Then $g(w^*) - g(w^\circ) \leq \epsilon \mathrm{OPT}$ *(note that* $g(w^*) = \mathrm{OPT}$*).*

By this lemma, we can focus on solving (5) now.

Saddle-SVC: Algorithms for saddle point optimization

- Preparation step: to achieve a fast running time, we hope that

$$\forall i \in [n],\ \|x_i\|_\infty \leq O(\sqrt{\log n/d}).$$

  - By a pre-processing procedure in [Ailon and Chazelle, 2010, Allen-Zhu et al. 2016], this property holds with high probablity.
- Roughly, Saddle-SVC alternatively maximizes the objective with respect to w and minimizes with respect to η and ξ.

Saddle-SVC: Algorithms for saddle point optimization

Algorithm 1 Update Rules of Saddle-SVC

1: $\gamma \leftarrow \epsilon\beta/2\log n$, $q \leftarrow O(\sqrt{\log n})$, $\tau \leftarrow \frac{1}{2q}\sqrt{d/\gamma}$, $\sigma \leftarrow 1/2q\sqrt{d\gamma}$, $\theta \leftarrow 1-1/(d+q\sqrt{d}/\sqrt{\gamma})$.

2: $w[0] \leftarrow \mathbf{0}^{\mathrm{T}}$, $\eta[-1]=\eta[0] \leftarrow \mathbf{1}^{\mathrm{T}}/n_1$, $\xi[-1]=\xi[0] \leftarrow \mathbf{1}^{\mathrm{T}}/n_2$.

3: Pick an index i^* in $[d]$ uniformly at random

4: $\delta_{i^*}^{+} \leftarrow \langle x_{i^*}^{+}, \eta[t]+\theta(\eta[t]-\eta[t-1])\rangle$, $\delta_{i^*}^{-} \leftarrow \langle x_{i^*}^{-}, \xi[t]+\theta(\xi[t]-\xi[t-1])\rangle$

5: $\forall i \in [d]$, $w_i[t+1] \leftarrow \begin{cases} (w_i[t]+\sigma(\delta_i^{+}-\delta_i^{-}))/(\sigma+1), & \text{if } i=i^* \\ w_i[t], & \text{if } i \neq i^* \end{cases}$

6: $\eta[t+1] \leftarrow$
$\arg\min_{\eta\in\Delta_1} \{\frac{1}{d}(w[t]+d(w[t+1]-w[t]))^{\mathrm{T}}x^{+}\eta + \frac{\gamma}{d}H(\eta) + \frac{1}{\tau}V_{\eta[t]}(\eta)\}$

7: $\xi[t+1] \leftarrow$
$\arg\min_{\xi\in\Delta_2} \{-\frac{1}{d}(w[t]+d(w[t+1]-w[t]))^{\mathrm{T}}x^{-}\xi + \frac{\gamma}{d}H(\xi) + \frac{1}{\tau}V_{\xi[t]}(\xi)\}$

Explanations of Saddle-SVC

Line 5 is equivalent to a variant of the proximal coordinate gradient method with l_2-norm regularization as follows.

$$w_{i^*}[t+1] = \arg\min_{w_{i^*}} \left\{ (\delta_{i^*}^+ - \delta_{i^*}^-) w_{i^*} - w_{i^*}^2/2 - (w_{i^*} - w_{i^*}[t])^2/2\sigma \right\}$$

- $(\delta_{i^*}^+ - \delta_{i^*}^-)$ can be considered as the term $\langle x_{i^*}^+, \eta[t] \rangle - \langle x_{i^*}^-, \xi[t] \rangle$ adding extra momentum terms $\langle x_{i^*}^+, \theta(\eta[t] - \eta[t-1]) \rangle$ and $-\langle x_{i^*}^-, \theta(\xi[t] - \xi[t-1]) \rangle$ for dual variables $\eta[t]$ and $\xi[t]$ respectively
- Further, $(\langle x_{i^*}^+, \eta[t] \rangle - \langle x_{i^*}^-, \xi[t] \rangle) w_{i^*} - w_{i^*}^2/2$ is the term in the objective function which are related to w.
- The $(w_{i^*} - w_{i^*}[t])^2/2\sigma)$ is a l_2-norm regularization term.

Explanations of Saddle-SVC

$$\eta[t+1] \leftarrow \arg\min_{\eta \in \Delta_1} \{ \frac{1}{d}(w[t] + d(w[t+1] - w[t]))^{\mathrm{T}} x^{+} \eta + \frac{\gamma}{d} H(\eta) + \frac{1}{\tau} V_{\eta[t]}(\eta) \}$$

- The update rule for η (or ξ) is the proximal gradient method with a Bergman divergence regularization term $\frac{1}{\tau} V_{\eta[t]}(\eta)$, where

$$V_x(y) = H(y) - \langle \nabla H(x), y - x \rangle - H(x)$$

- we also add a momentum term $d(w[t+1] - w[t])$ for primal variable w when updating η and ξ.

Case: ν-SVM

ν-SVM can be reduced to solving the saddle point optimization as follows.

$$\max_{w} \min_{\eta \in \mathcal{D}_{n_1}, \xi \in \mathcal{D}_{n_2}} \quad w^{\mathrm{T}} A\eta - w^{\mathrm{T}} B\xi + \gamma H(\eta) + \gamma H(\xi) - \tfrac{1}{2}\|w\|^2. \tag{6}$$

where $\gamma = \epsilon\beta/(2\log n)$,

- $\mathcal{D}_{n_1}$ is $\{\eta \mid \|\eta\|_1 = 1, 0 \le \eta_i \le \nu, \forall i\}$
- $\mathcal{D}_{n_2}$ is $\{\xi \mid \|\xi\|_1 = 1, 0 \le \xi_j \le \nu, \forall j\}$.

The only difference with the hard-margin SVM is the domains $\mathcal{D}_{n_1}$ and $\mathcal{D}_{n_2}$.

Update rule for ν-SVM

The only difference of the update rule with hard-margin SVM is an additional domain constraint.

$$\eta[t+1] := \arg \min_{\eta \in \mathcal{D}_{n_1}} \left\{ \frac{1}{d}(w[t] + d(w[t+1] - w[t]))^{\mathrm{T}} X\eta + \frac{\gamma}{d} H(\eta) + \frac{1}{\tau} V_{\eta[t]}(\eta) \right\}$$

Update rule for ν-SVM

It can be proved that the update rule is equivalent to the following procedure that takes $O(n)$ time.

- Step 1:

$$\eta_i := Z^{-1} \exp\{(\gamma + d\tau^{-1})^{-1}(d\tau^{-1} \log \eta_i[t] - \langle w[t] + d(w[t+1] - w[t]), x_{\cdot i}\rangle)\}$$

  for each $i \in [n_1]$, where Z ensures $\sum_i \eta_i = 1$.

- Step 2:

$$\begin{array}{lll} \textbf{while} & \varsigma := \sum_{\eta_i > \nu}(\eta_i - \nu) \neq 0 : & \\ & \Omega = \sum_{\eta_i < \nu} \eta_i & \\ & \forall i, \quad \textbf{if } \eta_i \geq \nu, & \textbf{then } \eta_i[t+1] = \nu \\ & \forall i, \quad \textbf{if } \eta_i < \nu, & \textbf{then } \eta_i[t+1] = \eta_i(1 + \varsigma/\Omega) \end{array}$$

Time Complexity

Saddle-SVC computes $(1-\epsilon)$-approximate solutions for HM-Saddle and ν-Saddle in $\tilde{O}(d+\sqrt{d/\epsilon\beta})$ iterations. Moreover, it takes $O(n)$ time for each iteration.

Theorem

A $(1-\epsilon)$-approximation for either hard-margin SVM or ν-SVM can be computed in $\tilde{O}(n(d+\sqrt{d/\epsilon\beta}))$ time.

Extension to Distributed Settings: Saddle-DSVC

Key: Estimate each iteration of Saddle-SVC in the distributed setting. The communication cost is $O(k)$ for each iteration.

Theorem

The communication cost of Saddle-DSVC is $\tilde{O}(k(d+\sqrt{d/\epsilon}))$.

If $d = \Theta(1/\epsilon)$, the communication lower bound is $\Omega(k(d+\sqrt{d/\epsilon}))$ which matches the communication cost of Saddle-DSVC.

Thank you!