

Building an SVM Solver

CS771: Introduction to Machine Learning

Purushottam Kar

Topics to be Covered

- Using Lagrangian duals to deal with constraints
- Deriving the SVM dual problem
- Applying GD variants to solve primal and dual SVM problems
- Recall we have objective fn $f: \mathbb{R}^d \rightarrow \mathbb{R}$ and constraint set $\mathcal{C} \subseteq \mathbb{R}^d$
$$\begin{array}{ll} \min_{\mathbf{w} \in \mathbb{R}^d} & f(\mathbf{w}) \\ \text{s.t.} & \mathbf{w} \in \mathcal{C} \end{array}$$
- We previously saw that interior point methods and projected GD were two nice ways to deal with constrained problems



Constrained Optimization

Method 3: Creating a Dual Problem

Suppose we wish to solve

$$\min_{\mathbf{x} \in \mathbb{R}^d} f(\mathbf{x})$$

Trick: sneak this constraint into the objective

Construct a barrier (indicator) for $r(\mathbf{x})$ so that $r(\mathbf{x}) = 0$ if \mathbf{x} is feasible, and $r(\mathbf{x}) = \infty$ otherwise, and have the

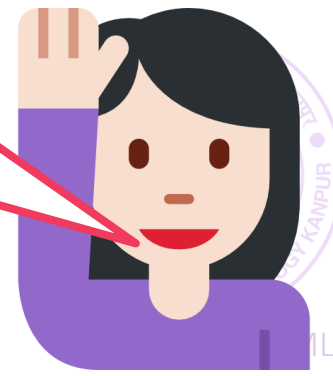
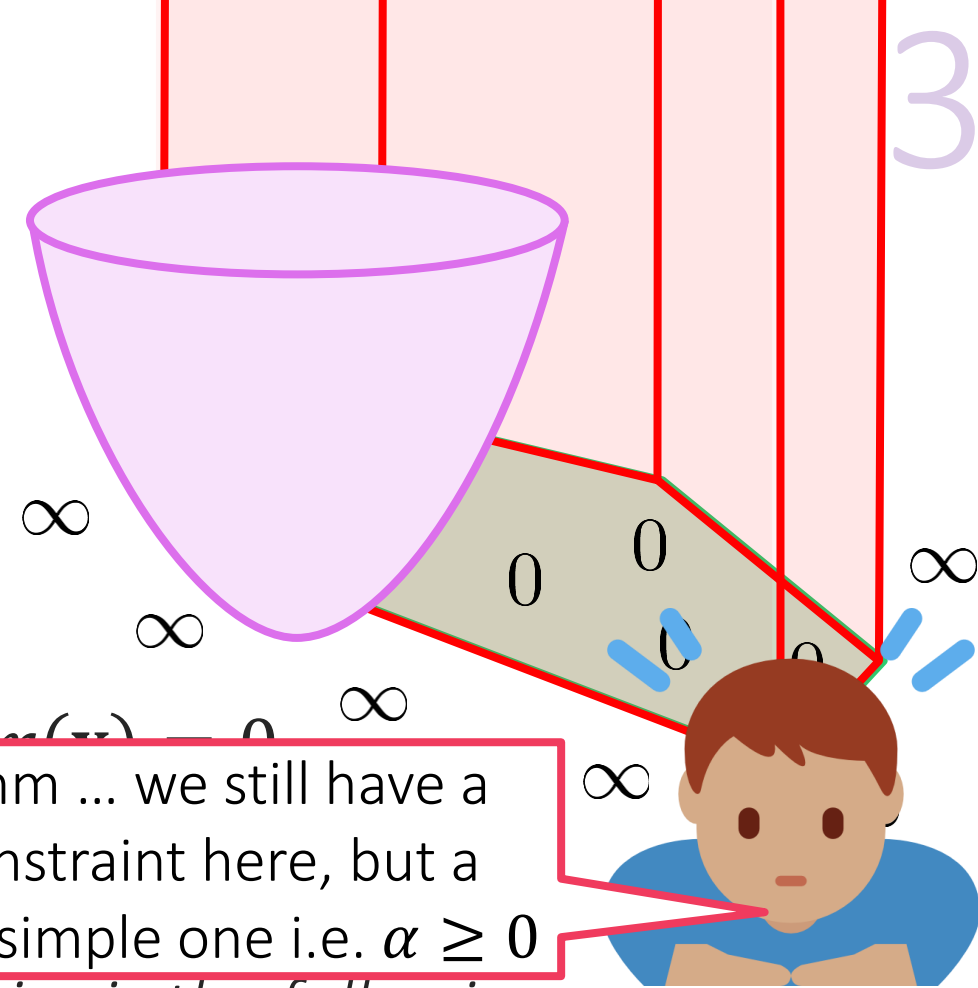
Let us see how to handle multiple constraints and equality constraints

Hmm ... we still have a constraint here, but a very simple one i.e. $\alpha \geq 0$

One very elegant way to construct such a barrier is the following

$$\text{Same as } \min_{\mathbf{x} \in \mathbb{R}^d} \left\{ \max_{\alpha \geq 0} \{f(\mathbf{x}) + \alpha \cdot g(\mathbf{x})\} \right\}$$

Thus, we want to solve



A few Cleanup Steps

4

Step 1: Convert your problem to a minimization problem

$$\max f(\mathbf{x}) \rightarrow \min -f(\mathbf{x})$$

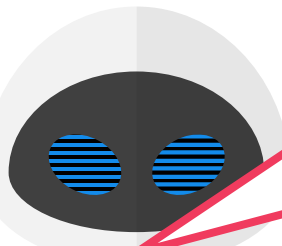
Step 2: Convert all inequality constraints to \leq constraints

$$g(\mathbf{x}) \geq 0 \rightarrow -g(\mathbf{x}) \leq 0$$

Step 3: Convert all equality constraints to two inequality constraints

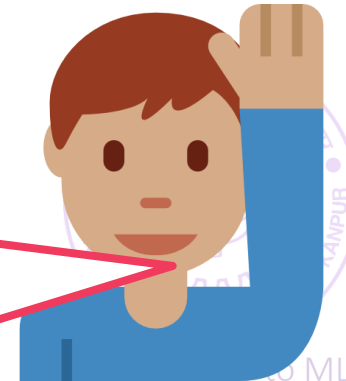
$$s(\mathbf{x}) = 0 \rightarrow s(\mathbf{x}) \leq 0, -s(\mathbf{x}) \leq 0$$

Step 4: For each constraint we now have, introduce a new variable
e.g. if we have C inequality constraints $g_1(\mathbf{x}) \leq 0, \dots, g_C(\mathbf{x}) \leq 0$,
introduce C new variables $\alpha_1, \dots, \alpha_C$



The variables of the original optimization problem, e.g. \mathbf{x} in this case, are called the *primal variables* by comparison

These new variables are called *dual variables* or sometimes even called *Lagrange multipliers*



The Lagrangian

5

$$\begin{array}{ll}\min_{\mathbf{x}} & f(\mathbf{x}) \\ \text{s.t.} & g_1(\mathbf{x}) \leq 0 \\ & g_2(\mathbf{x}) \leq 0 \\ & \vdots \\ & g_C(\mathbf{x}) \leq 0\end{array}$$

$\mathcal{L}(\mathbf{x}, \boldsymbol{\alpha}) = f(\mathbf{x}) + \sum_{c=1}^C \boldsymbol{\alpha}_c \cdot g_c(\mathbf{x})$ called the *Lagrangian* of the problem

If \mathbf{x} violates even one constraint, we have

$$\max_{\substack{\boldsymbol{\alpha} \in \mathbb{R}^C \\ \boldsymbol{\alpha}_c \geq 0}} \{\mathcal{L}(\mathbf{x}, \boldsymbol{\alpha})\} = \infty$$

If \mathbf{x} satisfies every single constraint, we have

This is just a nice way of rewriting the above problem

$$\min_{\mathbf{x} \in \mathbb{R}^d} \left\{ \max_{\substack{\boldsymbol{\alpha} \in \mathbb{R}^C \\ \boldsymbol{\alpha}_c \geq 0}} \left\{ f(\mathbf{x}) + \sum_{c=1}^C \boldsymbol{\alpha}_c \cdot g_c(\mathbf{x}) \right\} \right\}$$



The Dual Problem

6

The original optimization problem is also called the *primal problem*

Recall: variables of the original problem e.g. \mathbf{x} called *primal variables*

Using the Lagrangian, we rewrote the primal problem as

$$\min_{\mathbf{x} \in \mathbb{R}^d} \left\{ \max_{\substack{\boldsymbol{\alpha} \in \mathbb{R}^C \\ \alpha_c \geq 0}} \left\{ f(\mathbf{x}) + \sum_{c=1}^C \alpha_c \cdot g_c(\mathbf{x}) \right\} \right\}$$

The dual problem is obtained by simply switching order of min/max

$$\max_{\substack{\boldsymbol{\alpha} \in \mathbb{R}^C \\ \alpha_c \geq 0}} \left\{ \min_{\mathbf{x} \in \mathbb{R}^d} \left\{ f(\mathbf{x}) + \sum_{c=1}^C \alpha_c \cdot g_c(\mathbf{x}) \right\} \right\}$$

In some cases, the dual problem is easier to solve than the primal



Duality

7

Let $\hat{\mathbf{x}}^P, \hat{\boldsymbol{\alpha}}^P$ be the solutions to the primal problem i.e.

$$(\hat{\mathbf{x}}^P, \hat{\boldsymbol{\alpha}}^P) = \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^d} \left\{ \operatorname{argmax}_{\substack{\boldsymbol{\alpha} \in \mathbb{R}^C \\ \alpha_c \geq 0}} \left\{ f(\mathbf{x}) + \sum_{c=1}^C \alpha_c \cdot g_c(\mathbf{x}) \right\} \right\}$$

Let $\hat{\mathbf{x}}^D, \hat{\boldsymbol{\alpha}}^D$ be the solutions to the dual problem i.e.

$$(\hat{\mathbf{x}}^D, \hat{\boldsymbol{\alpha}}^D) = \operatorname{argmax}_{\substack{\boldsymbol{\alpha} \in \mathbb{R}^C \\ \alpha_c \geq 0}} \left\{ \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^d} \left\{ f(\mathbf{x}) + \sum_{c=1}^C \alpha_c \cdot g_c(\mathbf{x}) \right\} \right\}$$

Strong Duality: $\hat{\mathbf{x}}^P = \hat{\mathbf{x}}^D$ if the original problem is convex and “nice”

Complementary Slackness: $\hat{\alpha}_c^D \cdot g_c(\hat{\mathbf{x}}^D) = 0$ for all constraints c

Note: not complimentary but complementary 😊



Hard SVM without a bias

8

$\min_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{2} \|\mathbf{w}\|_2^2$ such that $1 - y^i \cdot \mathbf{w}^\top \mathbf{x}^i \leq 0$ for all $i \in [n]$

n constraints so we need n dual variables i.e. $\boldsymbol{\alpha} \in \mathbb{R}^n$

Lagrangian: $\mathcal{L}(\mathbf{w}, \boldsymbol{\alpha}) = \frac{1}{2} \|\mathbf{w}\|_2^2 + \sum_{i=1}^n \alpha_i (1 - y^i \cdot \mathbf{w}^\top \mathbf{x}^i)$

Primal problem: $\operatorname{argmin}_{\mathbf{w} \in \mathbb{R}^d} \left\{ \operatorname{argmax}_{\boldsymbol{\alpha} \geq 0} \left\{ \frac{1}{2} \|\mathbf{w}\|_2^2 + \sum_{i=1}^n \alpha_i (1 - y^i \mathbf{w}^\top \mathbf{x}^i) \right\} \right\}$

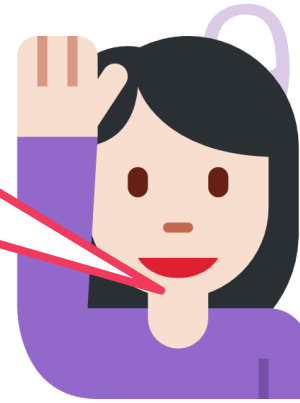
Dual problem: $\operatorname{argmax}_{\boldsymbol{\alpha} \geq 0} \left\{ \operatorname{argmin}_{\mathbf{w} \in \mathbb{R}^d} \left\{ \frac{1}{2} \|\mathbf{w}\|_2^2 + \sum_{i=1}^n \alpha_i (1 - y^i \mathbf{w}^\top \mathbf{x}^i) \right\} \right\}$

The dual problem can be greatly simplified!



Simplifying the Dual Problem

Once you get optimal values of α , use $\mathbf{w} = \sum_{i=1}^n \alpha_i y^i \cdot \mathbf{x}^i$ to get optimal value of \mathbf{w}



$$\operatorname{argmax}_{\alpha \geq 0} \left\{ \operatorname{argmin}_{\mathbf{w} \in \mathbb{R}^d} \left\{ \frac{1}{2} \|\mathbf{w}\|_2^2 + \sum_{i=1}^n \alpha_i (1 - y^i \mathbf{w}^\top \mathbf{x}^i) \right\} \right\}$$

Since this is an unconstrained problem with a convex and differentiable objective, we can apply first order optimality to solve it completely ☺

If we set the gradient to zero, we will get $\mathbf{w} = \sum_{i=1}^n \alpha_i y^i \cdot \mathbf{x}^i$

Substituting this back in the dual problem we get

$$\operatorname{argmax}_{\alpha \geq 0} \left\{ \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y^i y^j \langle \mathbf{x}^i, \mathbf{x}^j \rangle \right\}$$

This is actually the problem several solvers (e.g. libsvm, sklearn) solve



Support Vectors

Recall: we have α_i for every data point

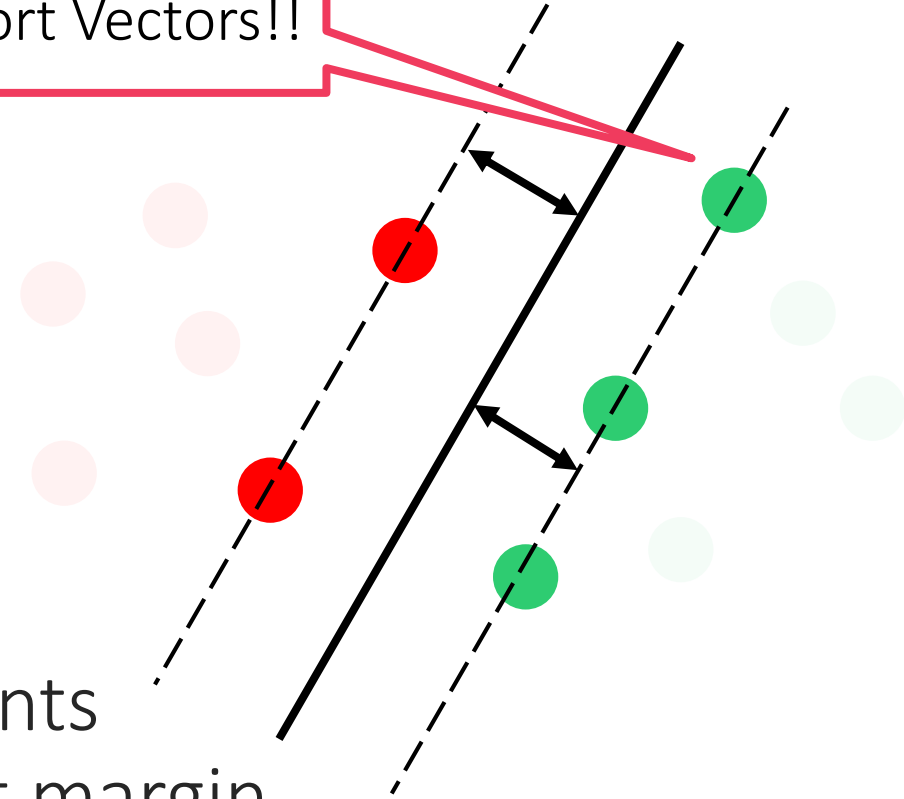
After solving the dual problem, the data points for which $\alpha_i \neq 0$: **Support Vectors**

Usually we have $\ll n$ support vectors

Recall: complementary slackness tells us that $\alpha_i(1 - y^i \mathbf{w}^T \mathbf{x}^i) = 0$ i.e. only those data points can become SVs for which $y^i \mathbf{w}^T \mathbf{x}^i = 1$ i.e. at margin

The reason these are called *support* vectors has to do with a mechanical interpretation of these objects – need to look at CSVM to understand that

Support Vectors!!



Dual for CSVM

11

Similar calculations (see course notes for a derivation) show that if we have a bias term b as well as slack variables, then the dual looks like

$$\begin{aligned} & \underset{\boldsymbol{\alpha} \in \mathbb{R}^n}{\operatorname{argmax}} \left\{ \sum_{i=1}^n \boldsymbol{\alpha}_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \boldsymbol{\alpha}_i \boldsymbol{\alpha}_j y^i y^j \langle \mathbf{x}^i, \mathbf{x}^j \rangle \right\} \\ & \text{s.t. } \boldsymbol{\alpha}_i \in [0, C], \text{ and } \sum_{i=1}^n \boldsymbol{\alpha}_i y^i = 0 \end{aligned}$$

Reason for the name “SVM”: imagine that each data point i is applying a force $\boldsymbol{\alpha}_i$ on the hyperplane in the direction y^i

Then the total force on the hyperplane is equal to zero since $\sum_{i=1}^n \boldsymbol{\alpha}_i y^i = 0$

Also, the condition $\mathbf{w} = \sum_{i=1}^n \boldsymbol{\alpha}_i y^i \cdot \mathbf{x}^i$ can be interpreted to mean that the total torque on the hyperplane is zero as well

Thus, support vectors mechanically support the hyperplane (don't let it shift or rotate around), hence their name 😊



CSVM Dual Problem

12

If we have a bias b , then the dual problem looks like

$$\begin{aligned} & \underset{\boldsymbol{\alpha} \in \mathbb{R}^n}{\operatorname{argmax}} \left\{ \sum_{i=1}^n \boldsymbol{\alpha}_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \boldsymbol{\alpha}_i \boldsymbol{\alpha}_j y^i y^j \langle \mathbf{x}^i, \mathbf{x}^j \rangle \right\} \\ & \text{s.t. } \boldsymbol{\alpha}_i \in [0, C], \text{ and } \sum_{i=1}^n \boldsymbol{\alpha}_i y^i = 0 \end{aligned}$$

The constraint $\sum_{i=1}^n \boldsymbol{\alpha}_i y^i = 0$ links all $\boldsymbol{\alpha}_i$ together. Cannot update a single $\boldsymbol{\alpha}_i$ without disturbing all the others ☹

A more involved algorithm Sequential Minimal Optimization (SMO) by John Platt is needed to solve the version with a bias – updates two $\boldsymbol{\alpha}_i$ at a time!

However, if we omit bias (hide it inside the model vector) the dual is

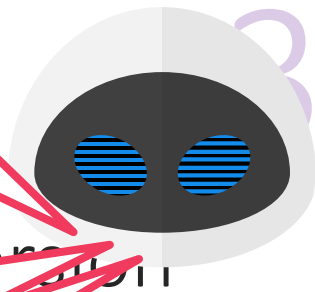
$$\begin{aligned} & \underset{\boldsymbol{\alpha} \in \mathbb{R}^n}{\operatorname{argmax}} \left\{ \sum_{i=1}^n \boldsymbol{\alpha}_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \boldsymbol{\alpha}_i \boldsymbol{\alpha}_j y^i y^j \langle \mathbf{x}^i, \mathbf{x}^j \rangle \right\} \\ & \text{s.t. } \boldsymbol{\alpha}_i \in [0, C] \end{aligned}$$

We will see a method to solve this simpler version of the problem



Solvers for the SVM

Sub-gradient since the primal objective is convex but non-differentiable



We can solve

Yes, coordinate ascent in the dual looks a lot like stochastic gradient descent in the primal! Both work with a single data point at a time

$$\arg\max_{\mathbf{w} \in \mathbb{R}^d} \left(\frac{1}{2} \|\mathbf{w}\|^2 + \sum_{i=1}^n \alpha_i y^i \langle \mathbf{w}, \mathbf{x}^i \rangle \right)$$

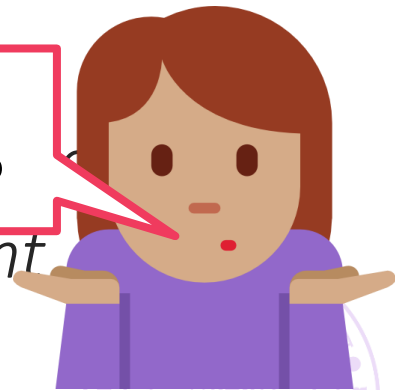
Projected since we have a constraint (albeit a simple one) in the dual

... or the dual version

$$\arg\max_{\boldsymbol{\alpha} \in \mathbb{R}^n} \left\{ \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y^i y^j \langle \mathbf{x}^i, \mathbf{x}^j \rangle \right\} \text{ s.t. } \alpha_i \in [0, C]$$

We may use gradient, coordinate

Does this mean I need to choose one data point at each time step?



For primal, we may use sub-gradient

For dual, we may use (projected) gradient ascent, coordinate ascent

We will actually see how to do coordinate maximization for dual

Since the optimization variable in the dual is $\boldsymbol{\alpha} \in \mathbb{R}^n$, we will need to take one coordinate at each time i.e. choose a different $i \in [n]$ at each time step

SDCM for the constrained problem

Warning: in general, finding an unconstrained solution and doing a projection step **does not** give a true solution



$$\operatorname{argmax}_{\alpha \in \mathbb{R}^n} \left\{ \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j v^i v^j \langle x^i, x^j \rangle \right\}$$

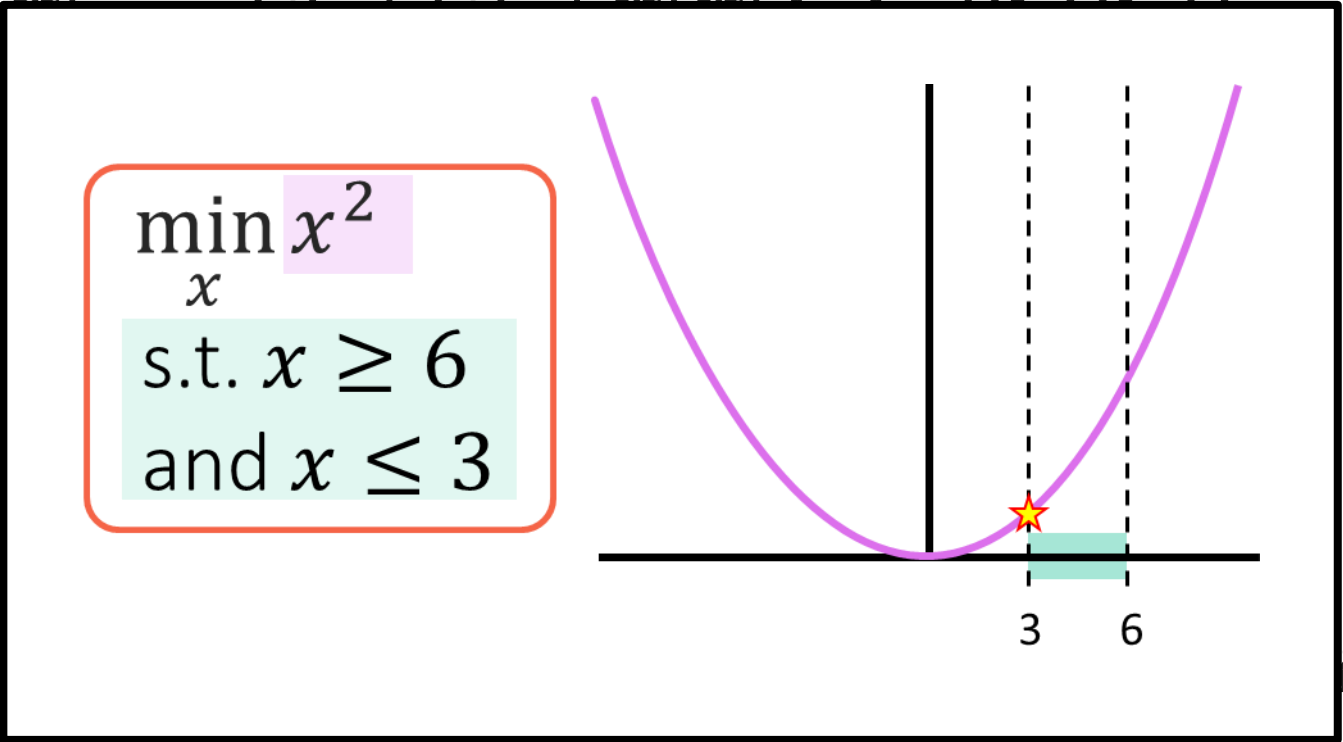
s.t. $\alpha_i \in [0, C]$

Concentrating on

$$\operatorname{argmax}_{\alpha_i \in \mathbb{R}} \left\{ \alpha_i - \frac{1}{2} \sum_{j=1}^n \alpha_i \alpha_j v^i v^j \langle x^i, x^j \rangle \right\}$$

s.t. $\alpha_i \in [0, C]$

Renaming $x =$

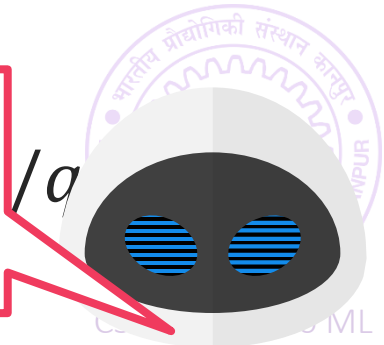


$\langle x^j \rangle$, we get

$$\operatorname{argmin}_x \frac{1}{2} q x^2 - x(1 - p) \text{ s.t. } x \in [0, C]$$

Solution is very simple: find \tilde{x} such that $\tilde{x} \in [0, C]$, solution is \tilde{x}

Indeed! In this special case, our objective had a nice property called *unimodality* which is why this trick works – it won't work in general



Speeding up SDCM computations

15

All that is left is to find how to compute p, q for our chosen i

$q = \|\mathbf{x}^i\|_2^2$ can be easily precomputed for all data points

However, $p = y^i \cdot \sum_{j \neq i} \alpha_j y^j \langle \mathbf{x}^i, \mathbf{x}^j \rangle$ needs $\mathcal{O}(nd)$ time to compute ☹

... only if done naively. Recall that we always have $\mathbf{w} = \sum_{i=1}^n \alpha_i y^i \cdot \mathbf{x}^i$ for the CSVM (even if we have bias and slack variables)

Thus, $\sum_{j \neq i} \alpha_j y^j \langle \mathbf{x}^i, \mathbf{x}^j \rangle = \mathbf{w}^\top \mathbf{x}^i - \alpha_i y^i \|\mathbf{x}^i\|_2^2 = \mathbf{w}^\top \mathbf{x}^i - \alpha_i y^i q$

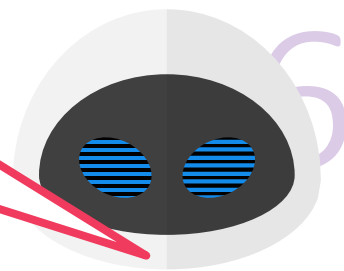
If we somehow had access to \mathbf{w} , then computing $\mathbf{w}^\top \mathbf{x}^i$ would take $\mathcal{O}(d)$ time and computing $\alpha_i y^i q$ would take $\mathcal{O}(1)$ time

All we need to do is create (and update) the \mathbf{w} vector in addition to the α vector and we would be able to find p in just $\mathcal{O}(d)$ time ☺



Which Method

Can you work out the details on how to implement stochastic primal coordinate descent in $\mathcal{O}(n)$ time per update?



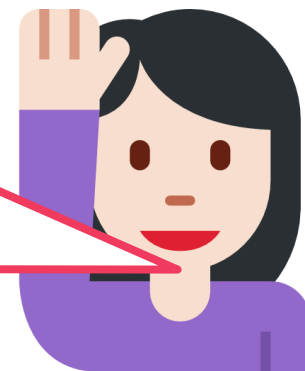
Gradient Methods

Primal

Dual Gradient

Stochastic

Be careful not to get confused with similar sounding terms. Coordinate Ascent takes a small step along one of the coordinates to increase the objective a bit. Coordinate Maximization instead tries to completely maximize the objective along a coordinate



Stochastic Primal Gradient Descent: $\mathcal{O}(d)$ time per update

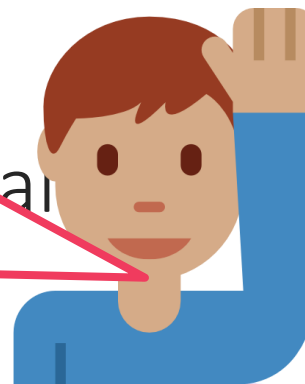
Stochastic Dual Gradient

Coordinate Methods:

Stochastic Primal Coordinate

Stochastic Dual Coordinate Maximization: $\mathcal{O}(d)$ time per update

Also be careful that some books/papers may call a method as “Coordinate Ascent” even when it is really doing Coordinate Maximization. The terminology is unfortunately a bit non-standard



Case 1: $n \gg d$: use SDCM or SPGD ($\mathcal{O}(d)$ time per update)

Case 2: $d \gg n$: use SDGA or SPCD ($\mathcal{O}(n)$ time per update)

