18-661 Introduction to Machine Learning

SVM – II

Fall 2020

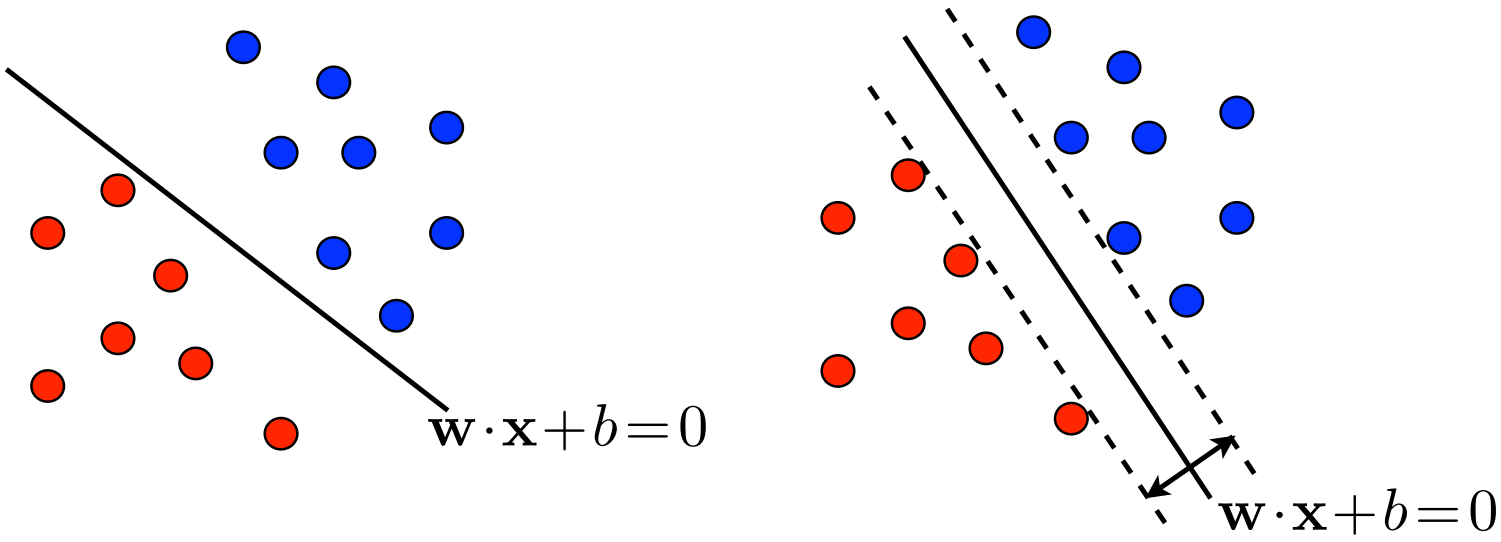
ECE – Carnegie Mellon University

Outline

1. Review of SVM Max Margin Formulation
2. SVM: Hinge Loss Formulation
3. A Dual View of SVMs (the short version)
4. Some Practice Exercises

Review of SVM Max Margin Formulation

Intuition: Where to put the decision boundary?



Find a decision boundary in the '*middle*' of the two classes that:

- Perfectly classifies the training data
- Is as far away from every training point as possible

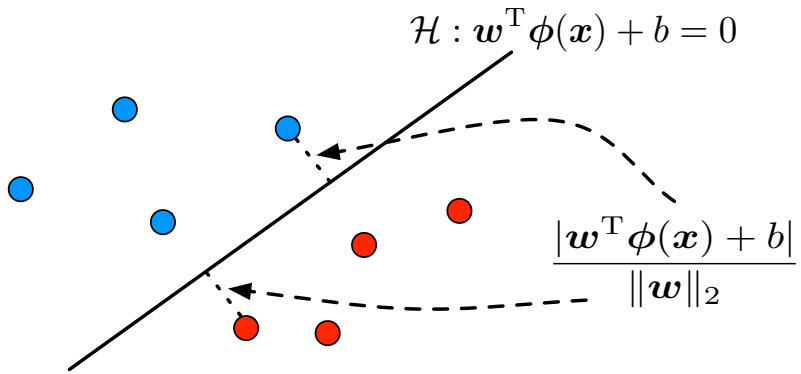
Let us apply this intuition to build a classifier that maximizes the margin between training points and the decision boundary.

Defining the margin

Margin

Smallest distance between the hyperplane and all training points

$$\text{MARGIN}(\boldsymbol{w}, b) = \min_n \frac{y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b]}{\|\boldsymbol{w}\|_2}$$



How can we use this to find the SVM solution?

Rescaled margin

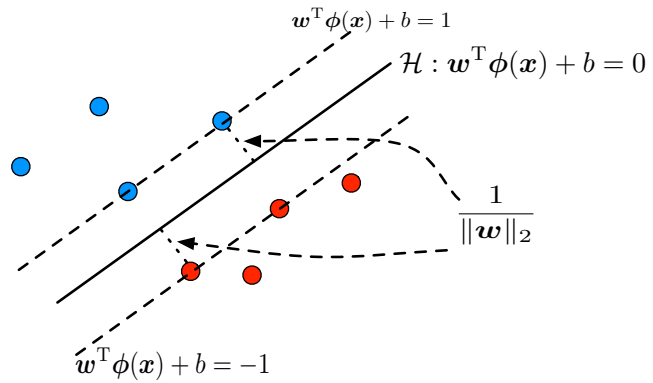
We can further constrain the problem by scaling $(\boldsymbol{w}, b)$ such that

$$\min_n y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] = 1.$$

Note that there always exists a scaling for which this is true. We've fixed the numerator in the $\text{MARGIN}(\boldsymbol{w}, b)$ equation, and we have:

$$\text{MARGIN}(\boldsymbol{w}, b) = \frac{\min_n y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b]}{\|\boldsymbol{w}\|_2} = \frac{1}{\|\boldsymbol{w}\|_2}$$

Hence the points closest to the decision boundary are at distance $\frac{1}{\|\boldsymbol{w}\|_2}$.



SVM: Max margin formulation for separable data

Assuming separable training data, we thus want to solve:

$$\max_{\boldsymbol{w},b} \underbrace{\frac{1}{\|\boldsymbol{w}\|_2}}_{\text{margin}} \quad \text{such that} \quad \underbrace{y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1, \quad \forall \quad n}_{\text{scaling of } \boldsymbol{w}, b}$$

This is equivalent to

$$\begin{aligned} \min_{\boldsymbol{w},b} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\ \text{s.t.} \quad & y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1, \quad \forall \quad n \end{aligned}$$

Given our geometric intuition, SVM is called a **max margin** (or large margin) classifier. The constraints are called **large margin constraints**.

SVM for non-separable data

Constraints in separable setting

$$y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1, \quad \forall \quad n$$

Constraints in non-separable setting

Can we modify our constraints to account for non-separability?
Specifically, we introduce slack variables $\xi_n \geq 0$:

$$y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n, \quad \forall \quad n$$

- For "hard" training points, we can increase ξ_n until the above inequalities are met.
- What does it mean when ξ_n is very large? We have violated the original constraints "by a lot."

Soft-margin SVM formulation

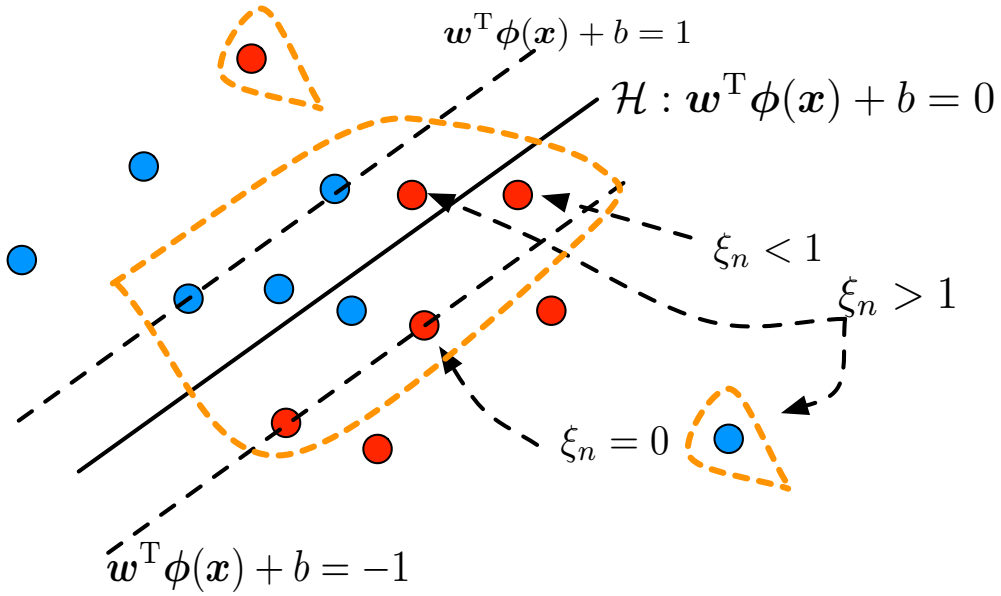
We do not want ξ_n to grow too large, and we can control their size by incorporating them into our optimization problem:

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n \\ \text{s.t.} \quad & y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n, \quad \forall \quad n \\ & \xi_n \geq 0, \quad \forall \; n \end{aligned}$$

What is the role of C?

- User-defined hyperparameter – same idea as the regularization term in ridge regression
- Trades off between the two terms in our objective.
- Increasing C will yield the same or smaller margin, and decreasing C will ignore more outliers and increase the margin.

Visualization of how training data points are categorized



Recall the constraints $y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n$. These define three types of support vectors:

- $\xi_n = 0$: The point is on the boundary.
- $0 < \xi_n \leq 1$: On the correct side, but inside the margin.
- $\xi_n > 1$: On the wrong side of the boundary.

Outline

1. Review of SVM Max Margin Formulation
2. SVM: Hinge Loss Formulation
3. A Dual View of SVMs (the short version)
4. Some Practice Exercises

SVM: Hinge Loss Formulation

SVM vs. Logistic regression

SVM soft-margin formulation

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n \\ \text{s.t.} \quad & y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n, \ \forall\ n \\ & \xi_n \geq 0, \quad \forall\ n \end{aligned}$$

Logistic regression formulation

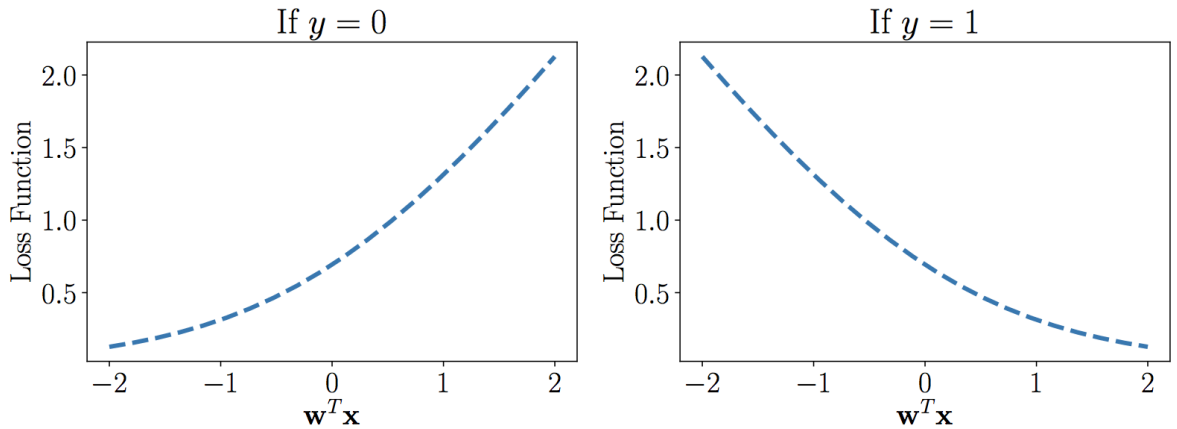
$$\begin{aligned} \min_{\boldsymbol{w}} -\sum_n \{ & y_n \log \sigma(\boldsymbol{w}^\top \boldsymbol{x}_n) \\ & + (1 - y_n) \log[1 - \sigma(\boldsymbol{w}^\top \boldsymbol{x}_n)]\} \\ & + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2 \end{aligned}$$

- Logistic regression defines a loss for each data point and minimizes the total loss plus a regularization term.
- This is convenient for assessing the "goodness" of the model on each data point.
- Can we write SVMs in this form as well?

Logistic regression loss: Illustration

$$\mathcal{L}(\boldsymbol{w}) = -\sum_n \{y_n \log \sigma(\boldsymbol{w}^\top \boldsymbol{x}_n) + (1 - y_n) \log[1 - \sigma(\boldsymbol{w}^\top \boldsymbol{x}_n)]\}$$

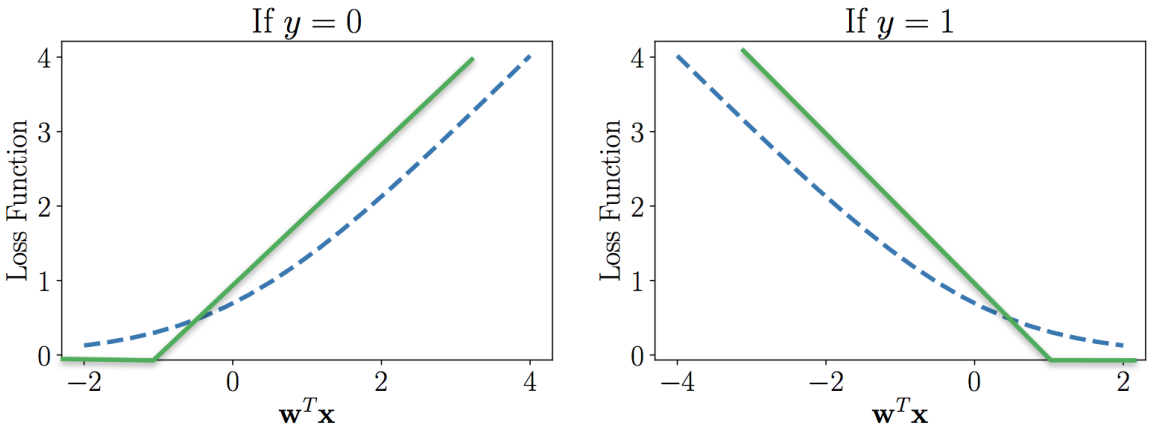
$$-\log\left(\frac{e^{-\mathbf{w}^T\mathbf{x_n}}}{1+e^{-\mathbf{w}^T\mathbf{x_n}}}\right) \qquad -\log\left(\frac{1}{1+e^{-\mathbf{w}^T\mathbf{x_n}}}\right)$$



- Alternative: Hinge Loss Function

Hinge loss: Illustration

$\max\left(\mathbf{w}^T\mathbf{x_n} + 1, 0\right)$ $\max\left(1 - \mathbf{w}^T\mathbf{x_n}, 0\right)$

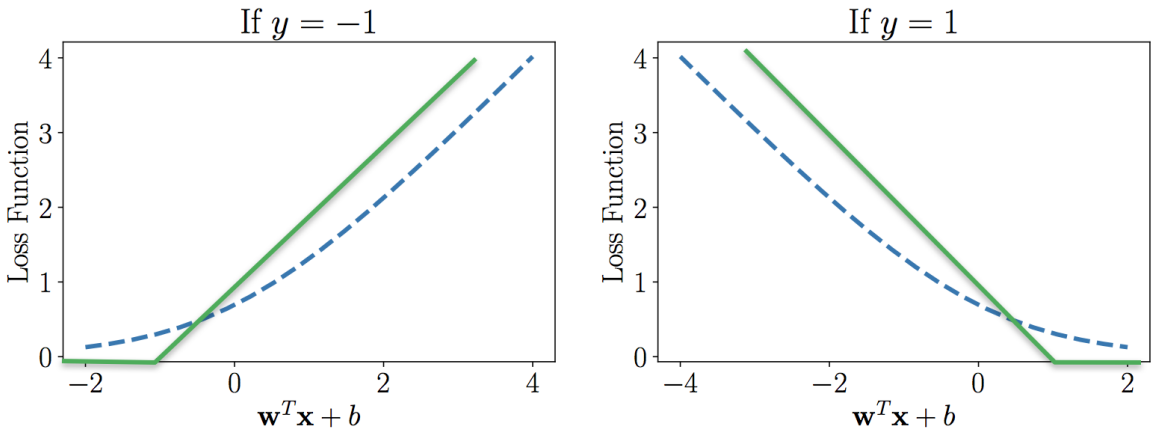


- Loss grows linearly as we move away from the boundary.
- No penalty if a point is more than 1 unit from the boundary.
- Makes the search for the boundary easier (as we will see later).

Hinge loss: Mathematical expression

$$\mathcal{L}(\boldsymbol{w}) = -\sum_{n} \max(0, 1 - y_n(\boldsymbol{w}^\top \boldsymbol{x}_n + b))$$

$\max\left(\mathbf{w}^T\mathbf{x_n} + b + 1, 0\right)$ $\max\left(1 - \mathbf{w}^T\mathbf{x_n} - b, 0\right)$



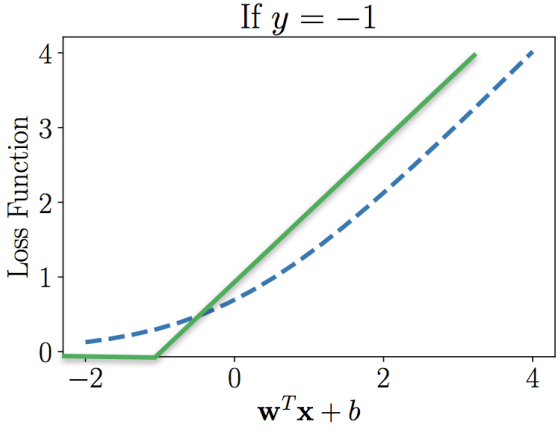
- Change of notation $y = 0 \rightarrow y = -1$
- Separate the bias term b from $\mathbf{w}$
- Makes the mathematical expression more compact

Hinge loss: Mathematical expression

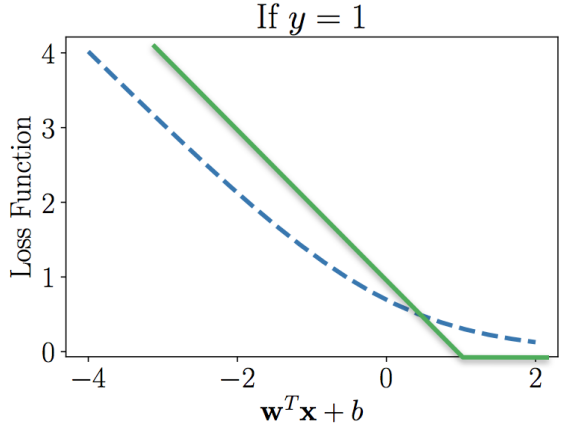
Assume $y \in \{-1, 1\}$ and the decision rule is $h(\boldsymbol{x}) = \text{SIGN}(\boldsymbol{w}^\top \boldsymbol{x} + b)$ with $f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b$,

$$\ell^{\text{HINGE}}(f(\boldsymbol{x}), y) = \begin{cases} 0 & \text{if } yf(\boldsymbol{x}) \geq 1 \\ 1 - yf(\boldsymbol{x}) & \text{otherwise} \end{cases}$$

$\max\left(\mathbf{w}^T\mathbf{x_n} + b + 1, 0\right)$



$\max\left(1 - \mathbf{w}^T\mathbf{x_n} - b, 0\right)$



Hinge loss

Assume $y \in \{-1, 1\}$ and the decision rule is $h(\boldsymbol{x}) = \text{SIGN}(f(\boldsymbol{x}))$ with $f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b$,

$$\ell^{\text{HINGE}}(f(\boldsymbol{x}), y) = \begin{cases} 0 & \text{if } yf(\boldsymbol{x}) \geq 1 \\ 1 - yf(\boldsymbol{x}) & \text{otherwise} \end{cases}$$

Intuition

- No penalty if raw output, $f(\boldsymbol{x})$, has same sign and is far enough from decision boundary (i.e., if 'margin' is large enough)
- Otherwise pay a growing penalty, between 0 and 1 if signs match, and greater than one otherwise

Convenient shorthand

$$\ell^{\text{HINGE}}(f(\boldsymbol{x}), y) = \max(0, 1 - yf(\boldsymbol{x})) = (1 - yf(\boldsymbol{x}))_+$$

Hinge loss SVM formulation

Minimizing the total hinge loss on all the training data

$$\min_{\boldsymbol{w},b} \sum_n \underbrace{\max(0, 1 - y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b])}_{\text{hinge loss for sample } n} + \underbrace{\frac{\lambda}{2}\|\boldsymbol{w}\|_2^2}_{\text{regularizer}}$$

Analogous to regularized least squares or logistic regression, as we balance between two terms (the loss and the regularizer).

- Can solve using gradient descent to get the optimal $\mathbf{w}$ and b
- Gradient of the first term will be either 0, $\mathbf{x}_n$ or $-\mathbf{x}_n$ depending on y_n and $\boldsymbol{w}^\top \boldsymbol{x}_n + b$.
- Much easier to compute than in logistic regression, where we need to compute the sigmoid function $\sigma(\boldsymbol{w}^\top \boldsymbol{x}_n + b)$ in each iteration.

Recovering our previous SVM formulation

Rewrite the geometric formulation as the hinge loss formulation:

$$\min_{\boldsymbol{w},b} \sum_n \max(0, 1 - y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b]) + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2$$

Here's the geometric formulation again:

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n \;\text{ s.t. } y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n,\; \xi_n \geq 0,\; \forall\; n$$

Now since $y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n \iff \xi_n \geq 1 - y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b]$:

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} C\sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 \text{ s.t. } \xi_n \geq \max(0, 1 - y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b]),\; \forall\; n$$

Now since the ξ_n should always be as small as possible, we obtain:

$$\min_{\boldsymbol{w},b} C\sum_n \max(0, 1 - y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b]) + \frac{1}{2}\|\boldsymbol{w}\|_2^2$$

Advantages of SVM

We've seen that the geometric formulation of SVM is equivalent to minimizing the empirical hinge loss. This explains why SVM:

1. Is less sensitive to outliers.
2. Maximizes distance of training data from the boundary.
3. Generalizes well to many nonlinear models.
4. Only requires a subset of the training points.
5. Scales better with high-dimensional data.

We will need to use duality to show the next three properties.

Outline

1. Review of SVM Max Margin Formulation
2. SVM: Hinge Loss Formulation
3. A Dual View of SVMs (the short version)
4. Some Practice Exercises

Please attend tomorrow's recitation to understand Lagrange duality better!

A Dual View of SVMs (the short version)

What is duality?

Duality is a way of transforming a constrained optimization problem.

It tells us sometimes-useful information about the problem structure, and can sometimes make the problem easier to solve.

- Dual problem is always convex–easy to solve.
- Primal and dual problems are not always equivalent.
- Dual variables tell us "how bad" constraints are.

The main point you should understand is that we will solve the dual SVM problem in lieu of the max margin (primal) formulation

Derivation of the dual

Here is a skeleton of how to derive the dual problem.

Recipe

1. Formulate the generalized Lagrangian function that incorporates the constraints and introduces dual variables
2. Minimize the Lagrangian function over the primal variables
3. Substitute the primal variables for dual variables in the Lagrangian
4. Maximize the Lagrangian with respect to dual variables
5. Recover the solution (for the primal variables) from the dual variables

Deriving the dual for SVM

Primal SVM

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n \\ \text{s.t.} \quad & y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n, \quad \forall \quad n \\ & \xi_n \geq 0, \quad \forall \; n \end{aligned}$$

The constraints are equivalent to the following canonical forms:

$$-\xi_n \leq 0 \quad \text{and} \quad 1 - y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] - \xi_n \leq 0$$

Lagrangian

$$\begin{aligned} L(\boldsymbol{w}, b, \{\xi_n\}, \{\alpha_n\}, \{\lambda_n\}) = C\sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 - \sum_n \lambda_n \xi_n \\ + \sum_n \alpha_n\{1 - y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] - \xi_n\} \end{aligned}$$

under the constraints that $\alpha_n \geq 0$ and $\lambda_n \geq 0$.

Deriving the dual of SVM

Lagrangian

$$L(\boldsymbol{w}, b, \{\xi_n\}, \{\alpha_n\}, \{\lambda_n\}) = C\sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 - \sum_n \lambda_n \xi_n + \sum_n \alpha_n\{1 - y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] - \xi_n\}$$

under the constraints that $\alpha_n \geq 0$ and $\lambda_n \geq 0$.

- Primal variables: $\boldsymbol{w}$, $\{\xi_n\}$, b; dual variables $\{\lambda_n\}$, $\{\alpha_n\}$
- Minimize the Lagrangian function over the primal variables by setting $\frac{\partial L}{\partial \boldsymbol{w}} = 0$, $\frac{\partial L}{\partial b} = 0$, and $\frac{\partial L}{\partial \xi_n} = 0$.
- Substitute the solutions to primal variables for dual variables in the Lagrangian
- Maximize the Lagrangian with respect to dual variables
- After some further maths and simplifications, we have...

Dual formulation of SVM

Dual is also a convex quadratic program

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \sum_n \alpha_n - \frac{1}{2} \sum_{m,n} y_m y_n \alpha_m \alpha_n \boldsymbol{x}_m^\top \boldsymbol{x}_n \\ \text{s.t.} \quad & 0 \leq \alpha_n \leq C, \quad \forall\ n \\ & \sum_n \alpha_n y_n = 0 \end{aligned}$$

- There are N dual variables α_n, one for each data point
- Independent of the size d of $\mathbf{x}$: SVM scales better for high-dimensional feature.
- May seem like a lot of optimization variables when N is large, but many of the α_n's become zero. α_n is non-zero only if the n^{th} point is a support vector

Once we solve for α_n's, how to get w and b?

Recovering **w**

$$\frac{\partial L}{\partial \boldsymbol{w}} = 0 \rightarrow \boldsymbol{w} = \sum_n \alpha_n y_n \boldsymbol{x}_n$$

Only depends on support vectors, i.e., points with $\alpha_n > 0$!

Recovering b

If $0 < \alpha_n < C$ and $y_n \in \{-1, 1\}$:

$$y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] = 1$$

$$b = y_n - \boldsymbol{w}^\top \boldsymbol{x}_n$$

$$b = y_n - \sum_m \alpha_m y_m \boldsymbol{x}_m^\top \boldsymbol{x}_n$$

Why do many α_n's become zero?

$$\max_{\boldsymbol{\alpha}} \quad \sum_n \alpha_n - \frac{1}{2}\sum_{m,n} y_m y_n \alpha_m \alpha_n \boldsymbol{x}_m^\top \boldsymbol{x}_n$$

$$\text{s.t.} \quad 0 \leq \alpha_n \leq C, \quad \forall\ n$$

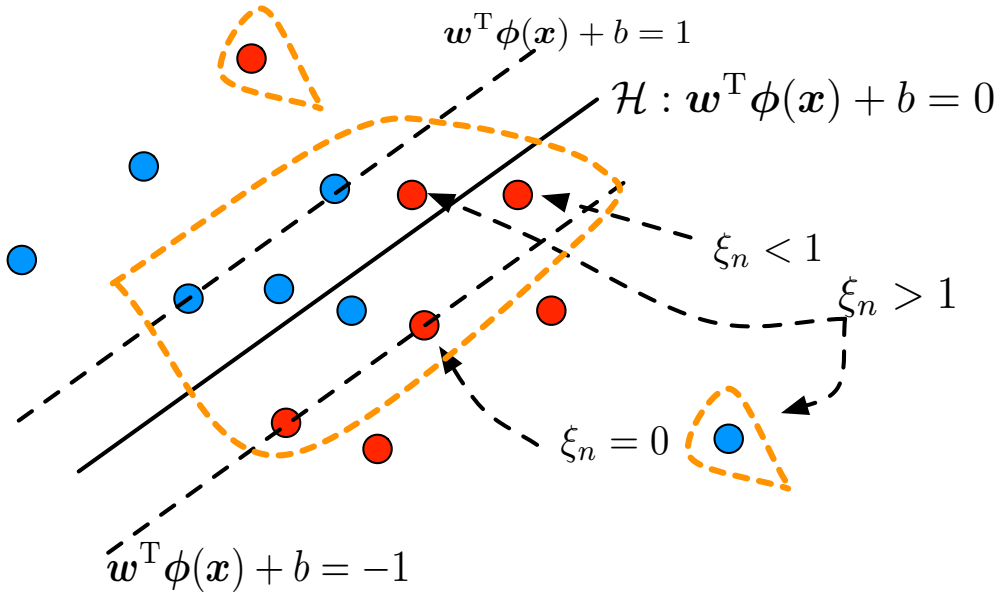
$$\sum_n \alpha_n y_n = 0$$

- By strong duality and KKT complementary slackness conditions, it tells us:

$$\alpha_n\{1 - \xi_n - y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b]\} = 0 \quad \forall n$$

- This tells us that $\alpha_n > 0$ iff $1 - \xi_n = y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b]$
  - If $\xi_n = 0$, then support vector is on the margin
  - Otherwise, $\xi_n > 0$ means that the point is an outlier

Visualizing the support vectors



Support vectors ($\alpha_n > 0$) are highlighted by the dotted orange lines.

- $\xi_n = 0$ and $0 < \alpha_n < C$ when $y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] = 1$.
- $\xi_n > 0$ and $\alpha_n = C$ if $y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] < 1$.

Some Practice Exercises

Exercise 1

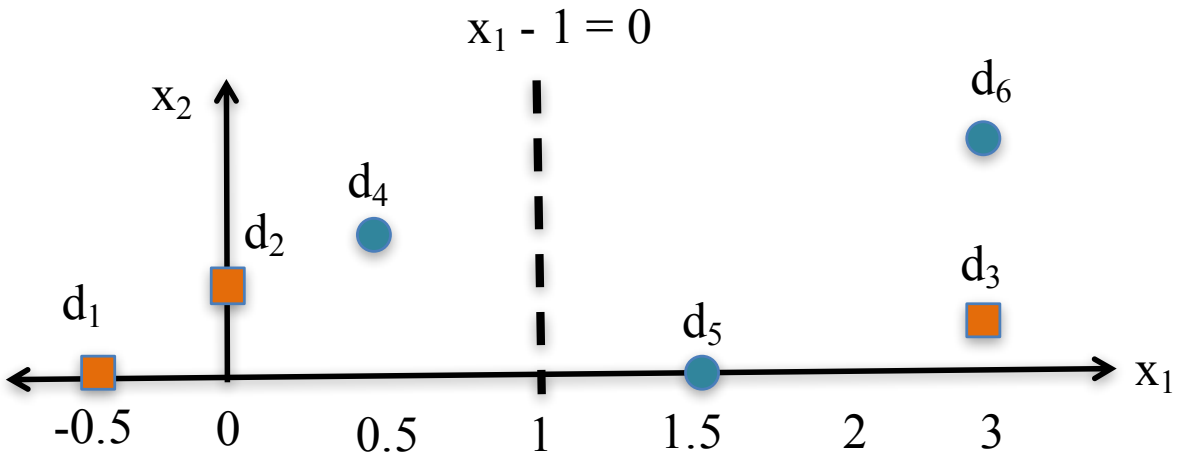


Figure 1

Consider a dataset of 6 data points $d_i = (\mathbf{x}_i, y_i)$ for $i = 1, 2, \ldots 6$, where $\mathbf{x} = [x_1, x_2]$. Points d_1, d_2, and d_3 have the label $y = -1$, and d_4, d_5 and d_6 have the label $y = 1$. We train a linear SVM classifier on this data and get the decision boundary $x_1 - 1 = 0$. Which points are the support vectors corresponding to this decision boundary? d_2, d_3, d_4, d_5

Exercise 2

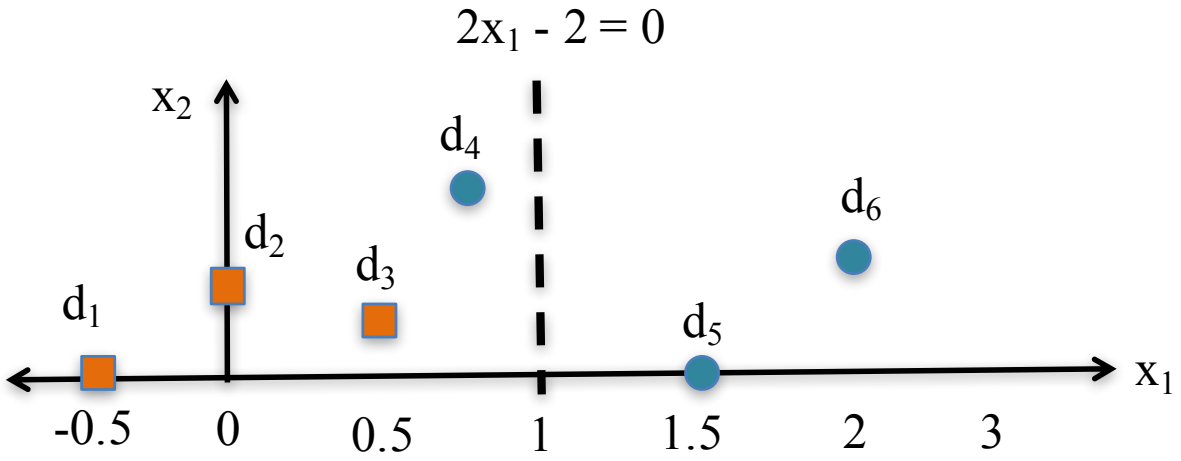


Figure 2

Consider a dataset of 6 data points $d_i = (\mathbf{x}^{(i)}, y^{(i)})$ for $i = 1, 2, \ldots, 6$, where $\mathbf{x} = [x_1, x_2]$. Points d_1, d_2, and d_3 have the label $y = -1$, and d_4, d_5 and d_6 have the label $y = 1$. We train a linear SVM classifier and obtain the decision boundary $[w_1, w_2] = [2, 0]$ and $b = -2$ as shown in Figure 2. For which points are the dual variables $\alpha_i = 0$?

d_1, d_2, d_6

Outline

1. Review of SVM Max Margin Formulation
2. SVM: Hinge Loss Formulation
3. A Dual View of SVMs (the short version)
4. Some Practice Exercises