Support vector machines

COMS 4721

Recap: linear classifiers (with $Y = \{-1, +1\}$)

Setting: linearly separable data
Assume there is a linear classifier that perfectly classifies the training data $S$: for some $w_*, t_* \in \mathbb{R}$,

$$y((w_*, x) - t_*) > 0 \text{ for all } (x, y) \in S.$$ 

Linear programming
Solve linear feasibility problem: find $w \in \mathbb{R}^d$ and $t \in \mathbb{R}$ such that

$$y((w, x) - t) > 0 \text{ for all } (x, y) \in S.$$ 

Can find some linear separator in polynomial time.

Perceptron algorithm
Finds some linear separator quickly if there is a large margin.

Support vector machines (SVMs)

Motivation
▶ Ambiguity and potential instability in what LP and Perceptron returns.
▶ Unclear what to do (or if algorithms work) when $S$ is not linearly separable, but you still hope to find a good linear separator with small classification error rate.

Support vector machines (Vapnik and Chervonenkis, 1963)
▶ Characterize a stable solution for linearly separable problems—the maximum margin solution.
▶ SVM specified as solution to a convex optimization problem that can be solved in polynomial time.
▶ Kernelizable via convex duality. (SVM gets its name from its dual form.)
▶ Slight alteration to optimization problem gives natural way to handle non-separable cases via convex surrogate losses.

The plan
1. Characterize the maximum margin solution as the solution to a optimization problem.
2. Study the dual of this optimization problem to reveal properties of the solution, and also show how to “kernelize” it.
3. Show how to generalize the optimization problem to the case where data are not linearly separable.
Maximum margin solution

**Why use the “maximum margin” solution?**

(i) Uniquely determined by $S$ (except in degenerate cases), unlike LP’s/Perceptron’s.

(ii) It is a particular “learning bias”—i.e., an assumption about the problem—that seems to be commonly useful.

**Our goal:** Precisely characterize the maximum margin solution as the solution to a mathematical optimization problem. (Don’t worry for now about how to actually solve the problem.)

Distance to a set

The distance between a point $x$ and a set $A$ is the Euclidean distance between $x$ and the closest point in $A$:

$$\text{dist}(x, A) := \min_{z \in A} \|x - z\|_2.$$ 

Distance to the decision boundary

Consider linear classifier $f_{w,t}$ (where $w \in \mathbb{R}^d \setminus \{0\}$ and $t \in \mathbb{R}$).

- Correct classification on $(x, y)$:
  
  $$f_{w,t}(x) = y \iff y(\langle w, x \rangle - t) > 0.$$ 

- Proj. of $x$ onto $\text{span}\{w\}$: $\frac{\langle w, x \rangle}{\|w\|_2} \cdot \frac{w}{\|w\|_2}$.

- Distance to affine hyperplane $H$ is
  
  $$\text{dist}(x, H) = \frac{\|w\|_2}{\|w\|_2}.$$ 

- If $f_{w,t}(x) = y$, then
  
  $$\text{dist}(x, H) = y(\langle w, x \rangle - t) \|w\|_2.$$
**Margin of a Linear Separator**

If \( f_w, t(x) = y \) for all \((x, y) \in S\), then the **margin** (i.e., smallest distance to decision boundary \( H \)) is

\[
\min_{(x, y) \in S} \text{dist}(x, H) = \frac{\min_{(x, y) \in S} y((w, x) - t)}{\|w\|_2}.
\]

To maximize the margin:

- Force numerator to be 1 via **linear constraints**:
  \[ y((w, x) - t) \geq 1 \text{ for all } (x, y) \in S. \]

- Then minimize the denominator \( \|w\|_2 \) subject to these constraints.

**Maximum Margin Linear Separator**

The solution \((\hat{w}, \hat{t})\) to the following mathematical optimization problem:

\[
\min_{w \in \mathbb{R}^d, t \in \mathbb{R}} \frac{1}{2} \|w\|^2
\]

s.t. \[ y((w, x) - t) \geq 1 \text{ for all } (x, y) \in S \]

gives the **linear classifier with the maximum margin on** \( S \).

The linear classifier obtained by solving this optimization problem is called a **support vector machine**.

The optimization problem is a **convex optimization problem** that can be solved in polynomial time. (Actual algorithm to come later.)

If there is a solution (i.e., the problem is separable), then the solution is **unique**. (Compare to LP’s and Perceptron’s lack of determinism from \( S \).)

**Duality**

- The SVM (maximum margin) solution is **entirely determined by certain data points called the “support vectors”**.
  
  In other words: can throw away all data except the support vectors, re-solve SVM optimization problem, and get the same solution.

- This is revealed via **convex duality**.

- As a bonus, we also get Kernel SVMs.
Observation
Where a separating hyperplane may be placed depends on the “outer” points on the sets. Points in the interior do not matter.

In geometric terms
For each class, replace all points from the class with the smallest convex set which contains all these points:

A set $C$ is convex if it contains the line segments between all pairs of points in $C$.

Support vectors
Margin is determined by the extreme points of the convex hulls closest to the hyperplane.

Implications
- SVM optimization problem focuses attention to the area closest to the decision surface.
- SVM classifier entirely determined by these closest extreme points—called the support vectors.

Maximum margin linear classifiers, again
Equivalent definition of maximum margin affine hyperplane:

1. Find shortest line segment connecting the convex hulls of negative examples $S_{\ominus} := \{ x : (x, -1) \in S \}$ and positive examples $S_{\oplus} := \{ x : (x, +1) \in S \}$.

2. Place affine hyperplane orthogonal to line segment at midpoint.

$$\min_{u \in \text{conv}(S_{\oplus}), \ v \in \text{conv}(S_{\ominus})} \| u - v \|_2$$

How do we optimize w.r.t. $u \in \text{conv}(S_{\oplus})$ and $v \in \text{conv}(S_{\ominus})$?

“Corner points” of the convex set are called extreme points.

Key idea
A hyperplane separates two classes if and only if it separates their convex hulls.
**Convex combinations**

For any set of points \( P := \{p_1, p_2, \ldots, p_m\} \subset \mathbb{R}^d \), every \( z \in \text{conv}(P) \) can be written as a **convex combination** of points in \( P \):

\[
  z = \sum_{i=1}^{m} \alpha_i p_i ,
\]

where \( \alpha_i \geq 0 \) for all \( i = 1, 2, \ldots, m \), and \( \sum_{i=1}^{m} \alpha_i = 1 \).

(\text{In fact, the convex combination only needs to involve the extreme points.})

**SVM dual optimization problem**

Suppose \( S_\oplus = \{x_1, \ldots, x_m\} \) and \( S_\ominus = \{x_{m+1}, \ldots, x_n\} \).

Write \( u \in \text{conv}(S_\oplus) \) and \( v \in \text{conv}(S_\ominus) \) as

\[
  u = \sum_{i=1}^{m} \alpha_i x_i , \quad v = \sum_{i=m+1}^{n} \alpha_i x_i .
\]

Optimization problem

\[
  \min_{u \in \text{conv}(S_\oplus), \atop v \in \text{conv}(S_\ominus)} \|u - v\|_2^2
\]

becomes

\[
  \min_{\alpha_1, \ldots, \alpha_n \geq 0} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle
\]

\[
  \text{s.t.} \quad \sum_{i=1}^{m} \alpha_i = \sum_{i=m+1}^{n} \alpha_i = 1 .
\]

(where \( \{(x_i, y_i)\}_{i=1}^{n} \) are the training examples).

**Using the dual solution**

Getting a linear classifier from the dual solution

Let \( \hat{\alpha}_1, \ldots, \hat{\alpha}_n \) be the solution to the dual problem. Then

\[
  \hat{w} := \hat{u} - \hat{v} = \sum_{i=1}^{n} \hat{\alpha}_i y_i x_i ,
\]

\[
  \hat{t} := \min_{x \in S_\oplus} \langle \hat{w}, x \rangle + \max_{x \in S_\ominus} \langle \hat{w}, x \rangle / 2 .
\]

\( \hat{w}, \hat{t} \) only depends on the support vector examples.

**Kernelizing SVMS** (Boser, Guyon, and Vapnik, 1992)

**SVM dual problem**

\[
  \min_{\alpha_1, \ldots, \alpha_n \geq 0} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j)
\]

\[
  \text{s.t.} \quad \sum_{i=1}^{m} \alpha_i = \sum_{i=m+1}^{n} \alpha_i = 1 .
\]

**SVM dual solution**

\[
  \langle \hat{w}, \cdot \rangle := \sum_{i=1}^{n} \hat{\alpha}_i y_i K(x_i, \cdot);
\]

\[
  \hat{t} := \frac{1}{2} \left( \max_{x \in S_\oplus} \sum_{i=1}^{n} \hat{\alpha}_i y_i K(x_i, x) + \min_{x \in S_\ominus} \sum_{i=1}^{n} \hat{\alpha}_i y_i K(x_i, x) \right) .
\]

Just need to keep around the support vectors (i.e., examples where \( \hat{\alpha}_i > 0 \)).
**SOFT-MARGIN SVMs** (Cortes and Vapnik, 1995)

When $S = \{(x_i, y_i)\}_{i=1}^n$ is not linearly separable, the (primal) SVM optimization problem

$$\min_{w \in \mathbb{R}^d, t \in \mathbb{R}} \frac{1}{2}\|w\|^2$$

subject to $y_i(\langle w, x_i \rangle - t) \geq 1$ for all $i = 1, 2, \ldots, n$

has no solution.

Introduce slack variables $\xi_1, \xi_2, \ldots, \xi_n \geq 0$, and a trade-off parameter $C > 0$:

$$\min_{w \in \mathbb{R}^d, t \in \mathbb{R}, \xi \in \mathbb{R}^n} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^n \xi_i$$

subject to $y_i(\langle w, x_i \rangle - t) \geq 1 - \xi_i$ for all $i = 1, 2, \ldots, n,$

$\xi_i \geq 0$ for all $i = 1, 2, \ldots, n$.

which is always feasible—"soft margin" SVM.

(Slack variables are auxiliary variables; not needed to form the linear classifier.)

---

**INTERPRETATION OF SLACK VARIABLES**

For given $(w, t)$, $\xi_i/\|w\|_2$ is distance that $x_i$ would have to move so that $y_i(\langle w, x_i \rangle - t) \geq 1$.

---

**ANOTHER INTERPRETATION OF SLACK VARIABLES**

Constraints with non-negative slack variables: (using $\lambda := 1/(nC)$)

$$\min_{w \in \mathbb{R}^d, t \in \mathbb{R}, \xi \in \mathbb{R}^n} \frac{\lambda}{2}\|w\|^2 + \frac{1}{n}\sum_{i=1}^n \xi_i$$

subject to $y_i(\langle w, x_i \rangle - t) \geq 1 - \xi_i$ for all $i = 1, 2, \ldots, n,$

$\xi_i \geq 0$ for all $i = 1, 2, \ldots, n$.

Equivalent unconstrained form:

$$\min_{w \in \mathbb{R}^d, t \in \mathbb{R}} \frac{\lambda}{2}\|w\|^2 + \frac{1}{n}\sum_{i=1}^n \ell_{\text{hinge}}(w, t; x_i, y_i).$$

Notation: $[a]_+ := \max\{0, a\}$.

The **hinge loss** of a linear classifier $f_{w, t}$ on an example $(x, y)$ is defined to be

$$\ell_{\text{hinge}}(w, t; x, y) := \left[1 - y(\langle w, x \rangle - t)\right]_+.$$
Zero-one loss vs. hinge loss

\[ \ell_{\text{hinge}}(w, t; x, y) \]

Hinge loss: an upper-bound on zero-one loss.

\[ 1 \{ y(\langle w, x \rangle - t) \leq 0 \} \leq 1 - y(\langle w, x \rangle - t) = \ell_{\text{hinge}}(w, t; x, y). \]

Soft-margin SVM minimizes an upper-bound on the training error rate, plus a term that encourages large margins.

This is computationally tractable (unlike minimizing training error rate) because the hinge loss is a convex function of \((w, t)\), and so is \(\frac{\lambda}{2} \|w\|^2\).

Recap

- Formulate learning as a mathematical optimization problem defined by training data \(\{(x_i, y_i)\}_{i=1}^n\):

\[
\min_{\mathbf{w} \in \mathbb{R}^d, t \in \mathbb{R}} \frac{\lambda}{2} \|\mathbf{w}\|^2 + \frac{1}{n} \sum_{i=1}^n \ell_{\text{hinge}}(\mathbf{w}, t; x_i, y_i).
\]

- Duality reveals properties of solution, delivers “kernelization”.