Outline

1 Administration

2 Review of last lecture

3 Geometric Understanding of SVM
Quiz 1

Tuesday Oct 21 6-7:20pm, THH 301

Please arrive on time

Make-up Oct 21 12-1:20pm, TBA

Those who have requested will be notified
Lecture schedule

**Oct 15 or Oct 16**

TA will lead on *Pragmatics*

**Oct 20 or Oct 21**

No lecture in the scheduled time — prepare for the quiz
Outline

1. Administration

2. Review of last lecture
   - Support vector machines

3. Geometric Understanding of SVM
Support vector machines

**Hinge loss** Assuming the label $y \in \{-1, 1\}$ and the decision rule is $h(x) = \text{SIGN}(f(x))$ with $f(x) = w^T \phi(x) + b$,

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

or $\ell_{\text{HINGE}}(f(x), y) = \max(0, 1 - yf(x))$

**Intuition**: penalize more if incorrectly classified (the left branch to the kink point)
Primal formulation of support vector machines (SVM)

Minimizing the total hinge loss on all the training data

$$\min_{w, b} \sum_n \max(0, 1 - y_n[w^T \phi(x_n) + b]) + \frac{\lambda}{2} \|w\|^2_2$$

equivalently,

$$\min_{w, b, \{\xi_n\}} C \sum_n \xi_n + \frac{1}{2} \|w\|^2_2$$

s.t. \[1 - y_n[w^T \phi(x_n) + b] \leq \xi_n, \quad \forall n\]
\[\xi_n \geq 0, \quad \forall n\]
Dual formulation and kernelized SVM

Dual is also a convex quadratic programming

\[
\max_{\alpha} \sum_n \alpha_n - \frac{1}{2} \sum_{m,n} y_m y_n \alpha_m \alpha_n \phi(x_m)^T \phi(x_n)
\]

s.t. \( 0 \leq \alpha_n \leq C, \ \forall \ n \)
\[
\sum_n \alpha_n y_n = 0
\]

We replace the inner products \( \phi(x_m)^T \phi(x_n) \) with a kernel function

\[
\max_{\alpha} \sum_n \alpha_n - \frac{1}{2} \sum_{m,n} y_m y_n \alpha_m \alpha_n k(x_m, x_n)
\]

s.t. \( 0 \leq \alpha_n \leq C, \ \forall \ n \)
\[
\sum_n \alpha_n y_n = 0
\]
Review of last lecture

Support vector machines

Recovering solution to the primal formulation

**Weights**

\[ \mathbf{w} = \sum_n y_n \alpha_n \phi(x_n) \leftarrow \text{Linear combination of the input features!} \]

**b**

\[ b = [y_n - \mathbf{w}^T \phi(x_n)] = [y_n - \sum_m y_m \alpha_m k(x_m, x_n)], \quad \text{for any } C > \alpha_n > 0 \]

**Making prediction on a test point \( x \)**

\[ h(x) = \text{SIGN}(\mathbf{w}^T \phi(x) + b) = \text{SIGN}(\sum_n y_n \alpha_n k(x_n, x) + b) \]

*Again, to make prediction, it suffices to know the kernel function.*
Things you need to know about deriving the dual

Make sure you can follow the recipe

- Formulate a Lagrangian function that incorporates the constraints, thru introducing dual variables
- Minimize the Lagrangian function to solve the primal variables
- Put the primal variables into the Lagrangian and express in terms of dual variables
- Maximize the Lagrangian with respect to dual variables
- Recover the solution (for the primal variables) from the dual variables
Outline

1. Administration
2. Review of last lecture
3. Geometric Understanding of SVM
Intuition: where to put the decision boundary?

Consider the binary classification in the following figure. We have assumed, for convenience, that the training dataset is separable — there is a decision boundary that separates the two classes perfectly.

There are infinite many ways of putting the decision boundary $\mathcal{H} : \mathbf{w}^T \phi(x) + b = 0$! Our intuition is, however, to put the decision boundary to be in the middle of the two classes as much as possible. In other words, we want the decision boundary is to be far to every point as much as possible as long as the decision boundary classifies every point correctly.
Distances

The distance from a point $\phi(x)$ to the decision boundary is

$$d_H(\phi(x)) = \frac{|w^T \phi(x) + b|}{\|w\|_2}$$

(We have derived the above in the recitation/quiz0. Please re-verify it as a take-home exercise). We can remove the absolute $|\cdot|$ by exploiting the fact that the decision boundary classifies every point in the training dataset correctly. Namely, $(w^T \phi(x) + b)$ and $x$’s label $y$ are of the same sign. The distance is now,

$$d_H(\phi(x)) = \frac{y[w^T \phi(x) + b]}{\|w\|_2}$$
Maximizing margin

**Margin** The margin is defined as the smallest distance from all the training points

\[
\text{MARGIN} = \min_n \frac{y_n [\mathbf{w}^T \phi(x_n) + b]}{\|\mathbf{w}\|_2}
\]
Maximizing margin

**Margin** The margin is defined as the smallest distance from all the training points

\[
\text{MARGIN} = \min_n \frac{y_n[w^T \phi(x_n) + b]}{\|w\|_2}
\]

Since we are interested in finding a \( w \) to put all points *as distant as possible* from the decision boundary, we maximize the margin

\[
\max_w \min_n \frac{y_n[w^T \phi(x_n) + b]}{\|w\|} = \max_w \frac{1}{\|w\|_2} \min_n y_n[w^T \phi(x_n) + b]
\]

\( \mathcal{H} : w^T \phi(x) + b = 0 \)
Rescaled margin

Since the margin does not change if we scale \((w, b)\) by a constant factor \(c\) (as \(w^T \phi(x) + b = 0\) and \((cw)^T \phi(x) + (cb) = 0\) are the same decision boundary), we fix the scale by forcing

\[
\min_n y_n[w^T \phi(x_n) + b] = 1
\]
Rescaled margin

Since the margin does not change if we scale \((w, b)\) by a constant factor \(c\) (as \(w^T \phi(x) + b = 0\) and \((cw)^T \phi(x) + (cb) = 0\) are the same decision boundary), we fix the scale by forcing

\[
\min_n y_n[w^T \phi(x_n) + b] = 1
\]

In this case, our margin becomes

\[
\text{MARGIN} = \frac{1}{\|w\|_2}
\]

precisely, the closest point to the decision boundary has a distance of that.
Combining everything we have, for a separable training dataset, we aim to

\[
\max_w \frac{1}{\|w\|_2^2} \quad \text{such that} \quad y_n[w^T \phi(x_n) + b] \geq 1, \quad \forall \ n
\]

This is equivalent to

\[
\min_w \frac{1}{2} \|w\|_2^2 \quad \text{s.t.} \quad y_n[w^T \phi(x_n) + b] \geq 1, \quad \forall \ n
\]

This starts to look like our first formulation for SVMs. For this geometric intuition, SVM is called \textit{max margin} (or large margin) classifier. The constraints are called \textit{large margin constraints}.
SVM for non-separable data

Suppose there are training data points that cannot be classified correctly no matter how we choose \( w \). For those data points,

\[
y_n[w^T \phi(x_n) + b] \leq 0
\]

for any \( w \). Thus, the previous constraint

\[
y_n[w^T \phi(x_n) + b] \geq 1, \quad \forall \ n
\]

is no longer feasible.
SVM for non-separable data

Suppose there are training data points that cannot be classified correctly no matter how we choose $w$. For those data points,

$$y_n[w^T \phi(x_n) + b] \leq 0$$

for any $w$. Thus, the previous constraint

$$y_n[w^T \phi(x_n) + b] \geq 1, \quad \forall \ n$$

is no longer feasible. To deal with this issue, we introduce slack variables $\xi_n$ to help

$$y_n[w^T \phi(x_n) + b] \geq 1 - \xi_n, \quad \forall \ n$$

where we also require $\xi_n \geq 0$. Note that, even for “hard” points that cannot be classified correctly, the slack variable will be able to make them satisfy the above constraint (we can keep increasing $\xi_n$ until the above inequality is met.)
SVM Primal formulation with slack variables

We obviously do not want $\xi_n$ goes to infinity, so we balance their sizes by penalizing them toward zero as much as possible

$$\min_w \frac{1}{2} \|w\|^2 + C \sum_n \xi_n$$

s.t. $y_n [w^T \phi(x_n) + b] \geq 1 - \xi_n$, $\forall n$

$\xi_n \geq 0$, $\forall n$

where $C$ is our tradeoff (hyper)parameter. This is precisely the primal formulation we first got for SVM.
Meaning of “support vectors” in SVMs

**Complementary slackness** At optimum, we have to have

\[
\alpha_n \{1 - \xi_n - y_n [w^T \phi(x_n) + b]\} = 0, \quad \forall \ n
\]

That means, for some \( n \), \( \alpha_n = 0 \). Additionally, our optimal solution is given by

\[
w = \sum_n \alpha_n y_n \phi(x_n) = \sum_{n: \alpha_n > 0} \alpha_n y_n \phi(x_n)
\]

In words, our solution is only determined by those training samples whose corresponding \( \alpha_n \) is strictly positive. Those samples are called *support vectors*.

Non-support vectors whose \( \alpha_n = 0 \) can be removed by the training dataset — this removal will not affect the optimal solution (i.e., after the removal, if we construct another SVM classifier on the reduced dataset, the optimal solution is the same as the one on the original dataset.)
Who are support vectors?

Case analysis Since, we have

\[ 1 - \xi_n - y_n [w^T \phi(x_n) + b] = 0 \]

We have

- \( \xi_n = 0 \). This implies \( y_n [w^T \phi(x_n) + b] = 1 \). They are on points that are \( 1/\|w\|_2 \) away from the decision boundary.
- \( \xi_n < 1 \). These are points that can be classified correctly but do not satisfy the large margin constraint – they have smaller distances to the decision boundary.
- \( \xi_n > 1 \). These are points that are misclassified.
Visualization of how training data points are categorized

Support vectors are those being circled with the orange line.