Lecture 5

Recap

$$w^* = \Phi^T \alpha = \sum_{i=1}^{n} \alpha_i \phi(x_i)$$

Calculate $\alpha = (K + \lambda I)^{-1} y$ where $K = \Phi\Phi^T \in \mathbb{R}^{n \times n}$ is the kernel matrix.

kernel Trick

$$w^{*T}\phi(x) = \sum_{i=1}^{n} \alpha_i \phi(x_i)^T \phi(x)$$

Therefore, only inner products in the new feature space matter!

Kernel methods are exactly about computing inner products without explicitly computing. The exact form of ! is inessential ϕ; all we need to do is know the inner products $\phi(x)^T\phi(x')$.

Support Vector Machines (SVMs)

- One of the most commonly used classification algorithms
- Allows us to explore the concept of *margins* in classification
- Works well with the kernel trick
- Strong theoretical guarantees

The function class for SVMs is a linear function on a feature map ϕ applied to the datapoints: $sign(w^T\phi(x)+b)$. Note, the bias term b is taken separately for SVMs.

Margins: separable case, geometric intuition

When data is **linearly separable**, there are infinitely many hyperplanes with zero training errpor:

The further away the separating hyperplane is from the datapoints, the better.

Margin for linearly separable data: Distance from the hyperplane to the point closest the the hyperplane.

Distance from a point to a hyperplane $w^Tx+b=0$?

Assume the projection is $x'=x-\beta\frac{w}{||w||_2}$, then

$$0=w^T(x-\beta\frac{w}{||w||_2})+b=w^Tx-\beta||w||+b\Rightarrow\beta=\frac{w^Tx+b}{||w||_2}$$

Therefore the distance is $||x-x'||_2=|\beta|=\frac{|w^Tx+b|}{||w||_2}$

For a hyperplane that correctly classifies (x,y) , the distance becomes $\frac{y(w^Tx+b)}{||w||_2}$

Motivation:

$$Pr[y|x;w]=\sigma(y(w^Tx+b))=\frac{1}{1+\exp(-y(w^Tx+b))}$$

$$\begin{cases}if\ y=1, want\ w^Tx+b\gg 0\\ if\ y=-1, want\ w^Tx+b\ll 0\end{cases}\quad\therefore\ want\ y(w^Tx+b)\gg 0$$

Margin: the smallest distance from all training points to the hyperplane

$$MARGIN\ of\ (w,b)=\min_i\frac{y_i(w^T\phi(x_i)+b)}{||w||_2}$$

The intuition "the further away the better" translates to solving:

$$\max_{w,b}\min_i\frac{y_i(w^T\phi(x_i)+b)}{||w||_2}=\max_{w,b}\frac{1}{||w||_2}\min_i y_i(w^T\phi(x_i)+b)$$

Maximizing margin, rescaling. rescaling (w,b) by multiplying both by some scalar does not change the hyperplane.

Decision Boundary:

$$w^T\phi(x) + \sigma = 0 \Leftrightarrow (10^6 w)^T\phi(x) + 10^6\sigma = 0$$

We can thus always scale (w, b) s.t. $\min_i y_i(w^T\phi(x_i) + b) = 1$ (Multiplying original (w, b) by $\frac{1}{\min_i(y_i(w^T\phi(x_i)+b))}$)

The margin then becomes

$$MARGIN\ OF\ (w, b) = \frac{1}{||w||_2} \min_i y_i(w^T\phi(x_i) + b) = \frac{1}{||w||_2}$$

SVM for separable data: "Primal" formulation

For a separable training set, we aim to solve

$$\max_{w.b} \frac{1}{||w||_2} \quad s.t.\ y_i(w^T\phi(x_i) + b) = 1$$

This is non-convex!

This is equivalent to:

$$\min_{w,b} \frac{1}{2}||w||_2^2$$
$$s.t.\ y_i(w^T\phi(x_i) + b) \geq 1, \forall i \in [n]$$

This is convex! Minimizing a convex function with convex constraint is convex.

SVM is thus also called max-margin classifier. The constraints above are called hard- margin constraints.

General non-separable case

If data is not linearly separable, the previous constraint

$$y_i(w^T\phi(x_i) + b) \geq 1,\ \forall i \in [n]$$

is obviously not feasible.

Can't even match $sign(w^T\phi(x_i) + b)\ \forall i \in [n]$, if not linearly separable.

Even if data is linearly separable, should we always separate it?

Forcing the classifier to classify all datapoints correctly might not be good.

If data is not linearly separable, the previous constraint $y_i(w^T\phi(x_i) + b) \geq 1, \forall i \in [n]$ is not feasible. And more generally, forcing classifier to always classify all datapoints correctly may not be the best idea.

To deal with this issue, we relax the constraints to l_1 norm soft-margin constraints:

$$y_i(w^T\phi(x_i) + b) \geq 1 - \xi_i,\ \forall i \in [n]$$
$$\Leftrightarrow 1 - y_i(w^T\phi(x_i) + b) \leq \xi_i,\ \forall i \in [n]$$

where we introduce slack variables $\xi \geq 0$.

Recall Hinge Loss:

$$l_{hinge}(z) = \max\{0, 1 - z\}$$

In our case, $z = y(w^T\phi(x) + b)$.

Why l_1 Penalization?

Hinge loss: $l(x) = \max(0, 1 - z)$

Squared hinge loss: $l(x) = \max(0, 1 - z)^2$

Difference: x^2 grows much faster than x , squared hinge loss would really penalize getting some predictions wrong.

Because of this absolute value loss can be more robust to outlines in data compared to squared loss.

a $1 - 0$ regression example: mean vs. median

If I have $x_1, x_2, \cdots, x_n$

What is $w_{l_2}^* = \arg\min_w \sum_i (x_i - w)^2$? $w_{l_2}^* = \frac{\sum x_i}{n}$

What is $w_{l_1}^* = \arg\min_w \sum_i |x_i - w|$? $w_{l_1} = median(x_1, \cdots, x_n)$

Median is more robust to outlines than mean.

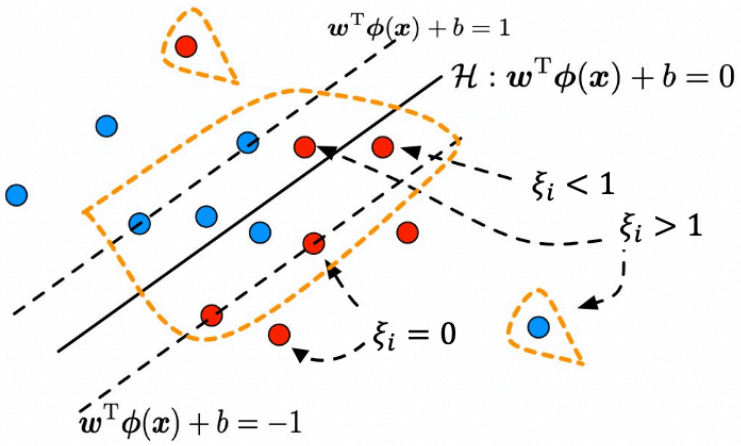
For $1 - 0$ regression:

$$\begin{aligned} y &= 10x + noise \\ w_{l_2}^* &= \arg\min_w \sum_i (y_i - wx_i)^2 \\ w_{l_1}^* &= \arg\min_w \sum_i |y_i - wx_i| \end{aligned}$$

We want ξ_i to be as small as possible. The objective becomes

$$\begin{aligned} &\min_{w,b,\{\xi_i\}} \frac{1}{2}||w||_2^2 + C\sum_i \xi_i \\ &s.t.\ y_i(w^T\phi(x_i) + b) \geq 1 - \xi_i,\ \forall i \in [n] \\ &\quad \xi_i \geq 0, \forall i \in [n] \end{aligned}$$

where C is a hyper-parameter to balance the two goals.



- when $\xi_i = 0$, point is classified correctly and satisfies large margin constraint.
- when $\xi_i < 1$, point is classified correctly but does not satisfy large margin constraint.
- when $\xi_i > 1$, point is misclassified.

Another view: In one sentence: linear model with l_2 regularized hinge loss. (l_1 is the penalization ξ , l_2 is the optimization problem $\min \frac{1}{2}||w||_2^2$)

For a linear model (w, b) ,this means

$$\min_{w,b} \sum_i \max\{0, 1 - y_i(w^T\phi(x_i) + b\} + \frac{\lambda}{2}||w||_2^2$$

Equivalent forms

The formulation

$$\begin{aligned} &\min_{w,b,\{\xi_i\}} C\sum_i \xi_i + \frac{1}{2}||w||_2^2 \\ &s.t.\ \ y_i(w^T\phi(x_i) + b) \geq 1 - \xi_i,\ \forall i \in [n] \\ &\quad \xi_i \geq 0, \forall i \in [n] \end{aligned}$$

In order to $\min \xi_i$, we should set ξ_i to be as small as possible, which is equivalent to:

$$\begin{aligned} &\min_{w,b,\{\xi_i\}} C\sum_i \xi_i + \frac{1}{2}||w||_2^2 \\ &s.t.\ \ \max\{0, 1 - y_i(w^T\phi(x_i) + b\} = \xi_i,\ \forall i \in [n] \end{aligned}$$

This is also equivalent to:

$$\min_{w,b} C\sum_i \max\{0, 1 - y_i(w^T\phi(x_i) + b\} + \frac{1}{2}||w||_2^2$$

And

$$\min_{w,b} \sum_i \max\{0, 1 - y_i(w^T\phi(x_i) + b\} + \frac{\lambda}{2}||w||_2^2$$

With $\lambda = 1/C$. This is exactly minimizing l_2 regularized hinge loss!

Optimization

$$\begin{aligned} \min_{w,b,\{\xi_i\}} & C\sum_i \xi_i + \frac{1}{2}||w||_2^2 \\ s.t.\ & y_i(w^T\phi(x_i) + b) \geq 1 - \xi_i,\ \forall i \in [n] \\ & \xi_i \geq 0, \forall i \in [n] \end{aligned}$$

It is a convex (in fact, a quadratic) problem.

Thus can apply any convex optimization algorithms, e.g. SGD.

There are more specialized and efficient algorithms, but usually we apply kernel trick, which requires solving the dual problem.

SVMs: Dual formulation & Kernel trick

Recall SVM formulation for separable cases:

$$\begin{aligned} \min_{w,b} & \frac{1}{2}||w||_2^2 \\ s.t.\ & y_i(w^T\phi(x_i) + b) \geq 1, \forall i \in [n] \end{aligned}$$

Can we use the kernel trick(dual) ?

Can we show that w^* is a linear combination of feature vectors $\phi(x_i)$?

Recall: by setting the gradient of $F(w) = ||\Phi w - y||_2^2 + \lambda||w||_2^2$ to be 0 :

$$w^* = \frac{1}{\lambda}\Phi^T(y - \Phi w^*) = \Phi^T\alpha = \sum_{i=1}^n \alpha_i\phi(x_i)$$

Thus the least square solution is a linear combination of features of the datapoints!

Kernelizing SVM

Claim: for the SVM problem, $w^* = \sum_i \alpha_i y_i \phi(x_i)$

Informal proof:

Formulation as a linear model with l_2 regularized hinge loss.

$$F(w) = \min_{w,\epsilon} \sum_i \max\{0, 1 - y_i(w^T\phi(x_i) + b)\} + \frac{\lambda}{2}||w||_2^2$$

This is a convex problem. $\therefore$ GD will find a minimizer with any initialization (for some appropriate step size).

Recall $l(z) = \max(0, 1 - z)$

$$\frac{\partial F(w)}{\partial w} = \sum_{i=1}^{n}\Big(\frac{\partial l(z)}{\partial z}|_{z=y_i(w^T\phi(x)+b)}(-y_i\phi(x_i))\Big) + \lambda w^{(t)}$$

$$w^{(0)} \leftarrow 0$$

$$w^{(t+1)} \leftarrow w^{(t)} - \eta\left(\sum_{i=1}^{n}\Big(\frac{\partial l(z)}{\partial z}|_{z=y_i(w^T\phi(x)+b)}(-y_i\phi(x_i))\Big) + \lambda w^{(t)}\right)$$

$\therefore$, $w^{(t)}$ always lie in span of $\phi(x_i)$ (data), which means $w^{(t)} = \sum \alpha_i^{(t)} y_i \phi(x_i)$, $\forall t$, for some $\alpha_i^{(t)}$

$$\therefore\ w^* = \sum \alpha_i^* y_i \phi(x_i) \quad for\ some\ a_i^*$$

If $w = \sum_i \alpha_i y_i \phi(x_i)$, how can we use this?

$$\begin{aligned} &\min_{w,b} \frac{1}{2}||w||_2^2 \\ &s.t.\ y_i(w^T\phi(x_i)+b) \geq 1, \forall i \in [n] \end{aligned} \quad \rightarrow \quad \begin{aligned} &\min_{\alpha,b} \frac{1}{2}||\sum_i \alpha_i y_i \phi(x_i)|| \\ &s.t.\ y_j(\sum_i \alpha_i y_i \phi(x_i)^T\phi(x_j)+b) \geqslant 1 \end{aligned}$$

This is equivalent to

$$\begin{aligned} &\min_{\alpha,b} \frac{1}{2}\sum_i\sum_j \alpha_i\alpha_j y_i y_j \phi(x_i)^T\phi(x_j) \\ &s.t.\ y_j(\sum_i \alpha_i y_i \phi(x_i)^T\phi(x_j)+b) \geq 1 \quad \forall i \in [n] \end{aligned}$$

Dual form for for separable cases

For the primal for the separable case, with some optimization theory (Lagrange duality, not covered in this class), we can show this is equivalent to,

$$\begin{aligned} &\max_{\alpha_i} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j} y_i y_j \alpha_i \alpha_j \phi(x_i)^T\phi(x_j) \\ &s.t.\ \sum_{i=1}^{n} \alpha_i y_i = 0\ and\ \alpha_i \geq 0,\ \forall i \in [n] \end{aligned}$$

Using the kernel function k for the mapping ϕ", we can kernelize this!

$$\begin{aligned} &\max_{\alpha_i} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j} y_i y_j \alpha_i \alpha_j k(x_i, x_j) \\ &s.t.\ \sum_{i=1}^{n} \alpha_i y_i = 0\ and\ \alpha_i \geq 0,\ \forall i \in [n] \end{aligned}$$

No need to compute $\varphi(x)$, only focus on inner product. This is also a quadratic program and many efficient optimization algorithms exist.

For the primal for the general (non-separable) case:

$$\min_{w,b,\{\xi_i\}} C\sum_i \xi_i + \frac{1}{2}||w||_2^2$$
$$s.t.\ y_i(w^T\phi(x_i)+b) \geq 1-\xi_i,\ \forall i \in [n]$$
$$\xi_i \geq 0, \forall i \in [n]$$

The dual is very similar,

$$\max_{\alpha_i} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j} y_i y_j \alpha_i \alpha_j k(x_i, x_j)$$
$$s.t.\ \sum_{i=1}^{n} \alpha_i y_i = 0\ and\ 0 \leq \alpha_i \leq C,\ \forall i \in [n]$$

How do we predict given the solution α_i^i to the dual optimization problem?

Remember that,

$$w^* = \sum_i \alpha_i^* y_i \phi(x_i) = w = \sum_{i:\alpha_i^*>0} \alpha_i^* y_i \phi(x_i)$$

A point with $\alpha_i^* > 0$ is called a "support vector". Hence the name SVM.

To make a prediction on any datapoint x,

$$sign(w^{*T}\phi(x)+b^*) = sign\Big(\sum_{i:\alpha_i^*>0} \alpha_i y_i \phi(x_i)\phi(x) + b^*\Big)$$
$$= sign\Big(\sum_{i:\alpha_i^*>0} \alpha_i y_i k(x_i, x) + b^*\Big)$$

All we need now is to identify b^* .

Bias term b^*

First, let's consider the separable case. It can be shown (we will not cover in class), that in the separable case the support vectors lie on the margin.

$y_i(w^{*T}\phi(x_i)+b^*) = 1$:

$$\Rightarrow w^{*T}\phi(x_i) + b = y_i$$
$$\Rightarrow b^* = y_i - w^{*T}\phi(x_i),\ for\ any\ i\ s.t.\ \alpha_i^* > 0$$

For general (non-separable case), For any support vector $\phi(x_i)$ with $0 < \alpha_i^* < C$, it can be shown that $1 = y_i(w^{*T}\phi(x_i)+b^*)$ (i.e. that support vector lies on the margin). Therefore, as before,

$$b^* = y_i - w^{*T}\phi(x_i) = y_i - \sum_{j=1}^{n} \alpha_j^* y_j k(x_j, x_i)$$

In practice, often average over all i with $0 < \alpha_i^* < C$ to stabilize computation.

With α^* and b^* in hand, we can make a prediction on any datapoint x ,

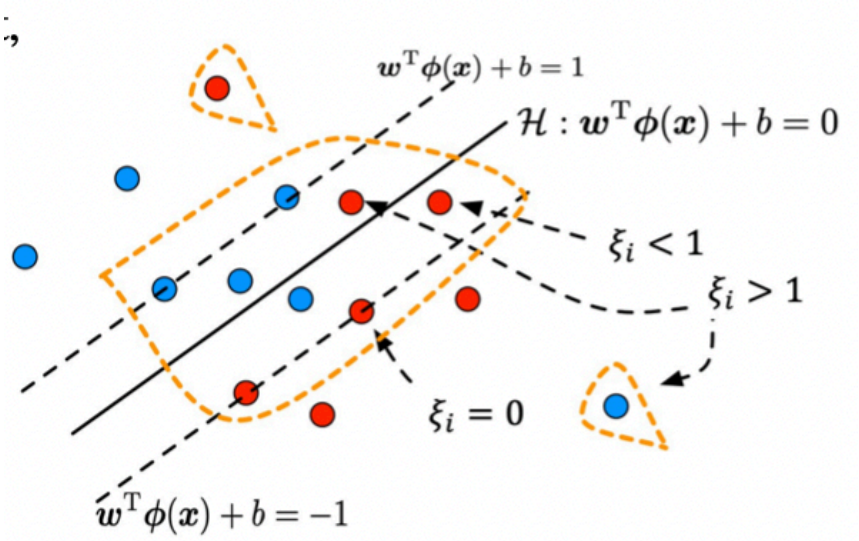
$$sign(w^{*T}\phi(x) + b^*) = sign\Big(\sum_{i:\alpha_i^*>0} \alpha_i^* y_i k(x_i, x) + b^*\Big)$$

SVMs: Understanding further

Support vectors are $\phi(x_i)$ such that $\alpha_i^* > 0$.

They are the set of points which satisfy one of the following:

1. they are tight with respect to the large margin constraint,
2. they do not satisfy the large margin constraint,
3. they are misclassified.



Support vectors (circled with the orange line) are the only points that matter!

when $\xi_i^* = 0, y_i(w^{*T}\phi(x_i) + b^*) = 1$, and thus the point is $1/||w^*||_2$ away from the hyperplane.

when $\xi_i^* < 1$, the point is classified correctly but does not satisfy the large margin constraint.

when $\xi_i^* > 1$, the point is misclassified.

One potential drawback of kernel methods: non-parametric, need to potentially keep all the training points.

$$sign(w^{*T}\phi(x) + b^*) = sign\Big(\sum_{i:\alpha_i^*>0} \alpha_i^* y_i k(x_i, x) + b^*\Big)$$

For SVM though, very often $SV = |\{i : \alpha_i^* > 0\}| \ll n$.

Linear Kernel and Polynomial Kernel

Linear kernel does nothing.

Data may become linearly separable when lifted to the high-dimensional feature space! That's what polynomial kernel does.

Gaussian Kernel

$$k(x, x') = \exp\Big(-\frac{||x - x'||_2^2}{2\sigma^2}\Big)$$

For some $\sigma > 0$ .This is also parameterized as

$$k(x, x') = \exp(-\gamma||x - x'||_2^2)$$

For some $\gamma > 0$. What does the decision boundary look like? What is the effect of γ ?

If γ is larger, the boundary will be much local (focus on local information). If γ is smaller, it will be more global.

Note that the prediction is of the form:

$$sign(w^{*T}\phi(x) - b^*) = sign\Big(\sum_{i:\alpha_i^*>0} \alpha_i^* y_i k(x_i, x) - b^*\Big)$$

Summary

SVM: max-margin linear classifier

Primal (equivalent to minimizing l_2 regularized hinge loss):

$$\begin{aligned}
&\min_{w,b,\{\xi_i\}} C\sum_i \xi_i + \frac{1}{2}||w||_2^2 \\
&s.t.\ y_i(w^T\phi(x_i) + b) \geq 1 - \xi_i,\ \forall i \in [n] \\
&\quad \xi_i \geq 0, \forall i \in [n]
\end{aligned}$$

Dual (kernelizable, reveals what training points are support vectors):

$$\begin{aligned}
&\max_{\alpha_i} \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i,j} y_i y_j \alpha_i \alpha_j \phi(x_i)^T\phi(x_j) \\
&s.t.\ \sum_{i=1}^n \alpha_i y_i = 0\ and\ 0 \leq \alpha_i \leq C,\ \forall i \in [n]
\end{aligned}$$