SVM

Sujay Sanghavi

Maximum Margin Classification

Binary classification: assume (now) that labels are $+1$ or -1.
Linear classifier:

$$\begin{aligned} \widehat{y}_\beta(x) &= +1 \quad \text{if } w^\top x - b > 0 \\ \widehat{y}_\beta(x) &= -1 \quad \text{if } w^\top x - b < 0 \end{aligned}$$

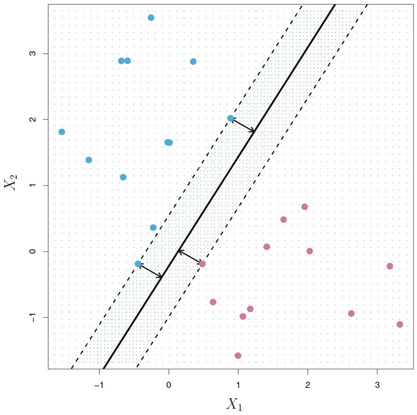
Every such linear classifier (w, b) has a corresponding **separating hyperplane**:
the set of all x's satisfying

$$w^\top x - b = 0$$

Data is **linearly separable** if there is some (w, b) which separates the data

Maximum Margin Classification

For linearly separable data, there may be many (w, b)'s that separate.
How to choose the "best" one ?



Margin of a linear classifier is the minimum distance between any point and the corresponding separating hyperplane.

Maximum margin classifier: the one with the largest margin.

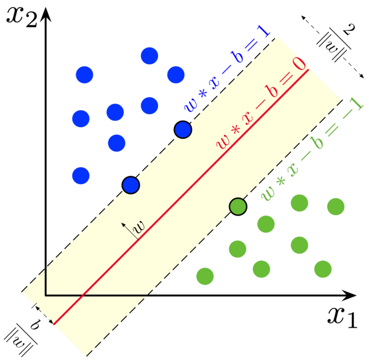
Finding Maximum Margin Classifier for Separable Data

(w, b) separates the data if for every sample $(x^{(i)}, y^{(i)})$ we have

$$y^{(i)} \left(w^\top x^{(i)} - b \right) \; > \; 0$$

We can rescale (w, b) appropriately so that for every sample $(x^{(i)}, y^{(i)})$,

$$y^{(i)} \left(w^\top x^{(i)} - b \right) \; \geq \; 1$$



The **margin**, i.e. the distance between the two hyperplanes

$$\begin{aligned} w^\top x - b &= +1 \\ w^\top x - b &= -1 \end{aligned}$$

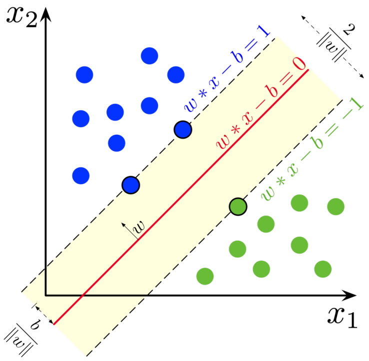
is equal to $\frac{2}{\|w\|}$

So we should minimize $\|w\|$ ...

Finding Maximum Margin Classifier for Separable Data

We can write max-margin classification of separable data as a convex optimization problem:

$$\min_{w,b} \quad \|w\|^2$$
$$s.t. \quad y^{(i)} \left(w^\top x^{(i)} - b \right) \ \geq \ 1 \quad \text{for all samples } i$$



Once the best $(\widehat{w}, \widehat{b})$ found by solving this problem, the classification rule is:

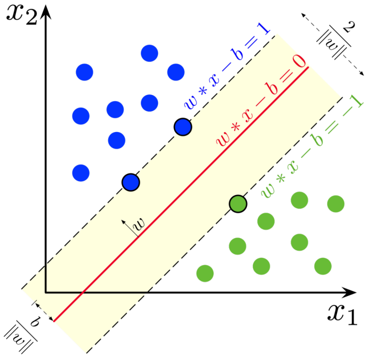
For a new feature vector x, estimate its label by

$$\widehat{y} \ = \ \text{sign}\left(\widehat{w}^\top x - \widehat{b} \right)$$

Finding Maximum Margin Classifier for Separable Data

We can write max-margin classification of separable data as a convex optimization problem:

$$\begin{aligned} \min_{w,b} \quad & \|w\|^2 \\ s.t. \quad & y^{(i)}\left(w^\top x^{(i)} - b\right) \geq 1 \quad \text{for all samples } i \end{aligned}$$



Best $(\widehat{w}, \widehat{b})$ determined by the data points i at the boundaries, i.e. for which

$$y^{(i)}\left(\widehat{w}^\top x^{(i)} - \widehat{b}\right) = 1$$

These samples i are called **support vectors**

Soft Margin

Q: But what if training data is not separable ?

A: Add a penalty for being wrong. This is often called a "soft margin" ...

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{n}\sum_i \xi^{(i)} \; + \; \lambda \|w\|^2 \\ s.t. \quad & y^{(i)}\left(w^\top x^{(i)} - b\right) \; \geq \; 1 - \xi^{(i)} \quad \text{for all samples } i \\ & \xi^{(i)} \; \geq \; 0 \quad \text{for all samples } i \end{aligned}$$

Again, the best $(\widehat{w}, \widehat{b})$ are determined by the **support vectors**, which have $y^{(i)}\left(\widehat{w}^\top x^{(i)} - \widehat{b}\right) \; = \; 1 - \xi^{(i)}$.

Dual of SVM

Every convex optimization problem has a **dual problem**. The dual of the soft margin one is

$$\begin{aligned} \max_c \quad & \sum_i c^{(i)} - \frac{1}{2}\sum_i \sum_j y^{(i)} c^{(i)} \langle x^{(i)}, x^{(j)} \rangle y^{(j)} c^{(j)} \\ s.t. \quad & \sum_i c^{(i)} y^{(i)} = 0 \\ & 0 \le c^{(i)} \le \frac{1}{2n\lambda} \quad \text{for all samples } i \end{aligned}$$

Suppose $\widehat{c}$ is the optimum of this dual problem.
The optimal $(\widehat{w}, \widehat{b})$ of the original problem is then given by

$$\widehat{w} = \sum_i \widehat{c}^{(i)} y^{(i)} x^{(i)} \quad \text{and} \quad \widehat{b} = w^\top x^{(i)} - y^{(i)} \text{ for any supporting } i$$

$\star$ Turns out $\widehat{c}^{(i)} \neq 0$ only if sample i a supporting vector.

Kernel Trick

Replace all $\langle x^{(i)}, x^{(j)} \rangle$ by $k(x^{(i)}, x^{(j)})$, where $k(\cdot,\cdot)$ is the **kernel function**

$$\begin{aligned} \max_{c} \quad & \sum_i c^{(i)} \;-\; \frac{1}{2} \sum_i \sum_j y^{(i)} c^{(i)}\, k(x^{(i)}, x^{(j)})\, y^{(j)} c^{(j)} \\ s.t. \quad & \sum_i c^{(i)} y^{(i)} \;=\; 0 \\ & 0 \le c^{(i)} \le \frac{1}{2n\lambda} \quad \text{for all samples } i \end{aligned}$$

Suppose $\widehat{c}$ is the optimum of this dual problem.
The optimal $(\widehat{w}, \widehat{b})$ now not easy to write down, but:

$$k(\widehat{w}, x) \;=\; \sum_i \widehat{c}^{(i)} y^{(i)} k(x^{(i)}, x)$$

$\star$ Turns out $\widehat{c}^{(i)} \neq 0$ only if sample i a supporting vector.

Kernel SVM

Solve:

$$\begin{aligned}
\max_{c} \quad & \sum_i c^{(i)} \;-\; \frac{1}{2} \sum_i \sum_j y^{(i)} c^{(i)}\, k(x^{(i)}, x^{(j)})\, y^{(j)} c^{(j)} \\
s.t. \quad & \sum_i c^{(i)} y^{(i)} \;=\; 0 \\
& 0 \le c^{(i)} \le \frac{1}{2n\lambda} \quad \text{for all samples } i
\end{aligned}$$

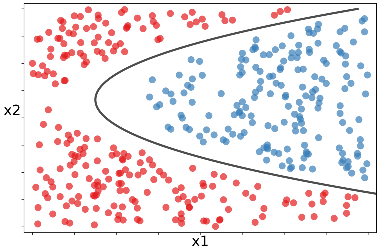
and then decision rule:

$$\widehat{y}(x) \;=\; \text{sign}\left(\sum_i \widehat{c}^{(i)} y^{(i)} k(x^{(i)}, x) \;-\; \widehat{b} \right)$$

So decision requires evaluating $k(x^{(i)}, x)$ for the supporting vectors i.
Hence the name **Support Vector Machine**.

Kernel SVM

Different choices of the kernel $k(\cdot,\cdot)$ give different classifiers. These are **non-linear** classifiers.
Now: two popular kernels.



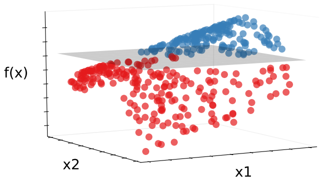
Quadratic Kernel:

$$k(a,b) \;=\; \left(a^\top b\right)^2$$

This is **equivalent to a linear classifier over a larger set of features.**

Quadratic Kernel SVM

Quadratic Kernel: $k(a,b) \; = \; \left(a^\top b\right)^2$

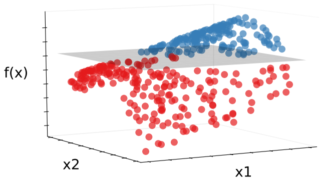


In $d = 2$ dimensions, quadratic kernel can be thought of as linear classification using more features:

$$(x_1^2,\, x_1x_2,\, x_2^2,\, x_1,\, x_2,\, 1)$$

Quadratic Kernel SVM

Quadratic Kernel: $k(a,b) \; = \; \left(a^\top b\right)^2$



In $d = 2$ dimensions, quadratic kernel can be thought of as linear classification using more features:

$$(x_1^2,\, x_1x_2,\, x_2^2,\, x_1,\, x_2,\, 1)$$

$$\widehat{y}(x) \; = \; \beta_{11}x_1^2 + \beta_{12}x_1x_2 + \beta_{22}x_2^2 + \beta_1x_1 + \beta_2x_2 + \beta_0$$

vs

$$\widehat{y}(x) \; = \; \beta_1x_1 + \beta_2x_2 + \beta_0$$

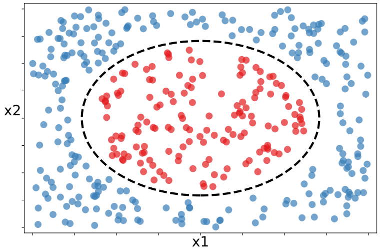
(this is NOT how they are actually made, this is just for understanding)

RBF Kernel SVM

Radial Basis Function (RBF) Kernel: (aka "Gaussian Kernel") is

$$k(a, b) \;=\; \exp\left(-\frac{\|a-b\|^2}{2\sigma^2}\right)$$

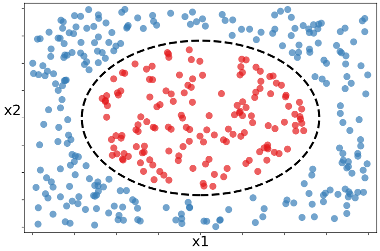
σ is a parameter.

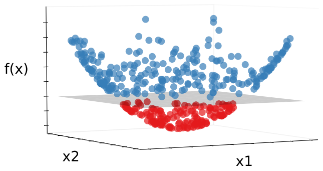


This is equivalent to using ...

... an infinite number of non-linear features, corresponding to the taylor expansion of $k(\cdot,\cdot)$ ...

RBF Kernel SVM





Properties of Kernels

A kernel function $k(\cdot, \cdot)$ has to be

- **Symmetric:** $k(a, b) \; = \; k(b, a)$ for all points a and b
- **Positive Definite:** for any n points $x^{(1)}, \dots, x^{(n)}$ and real numbers $c^{(1)}, \dots, c^{(n)}$, it has to be that

$$\sum_{i=1}^{n} \sum_{j=1}^{n} k(x^{(i)}, x^{(j)}) \, c^{(i)} \, c^{(j)} \; \geq \; 0$$

Why use Kernels ?

Main advantage: Kernels can have model complexity and (hopefully) model **accuracy grow with dataset size.**

Linear Classifier:

$$\widehat{y}(x) = \text{sign}(\widehat{w}^\top x - \widehat{b})$$

This changes with number of training samples only if $(\widehat{w}, \widehat{b})$ change.

Why use Kernels ?

Main advantage: Kernels can have model complexity and (hopefully) model **accuracy grow with dataset size.**

Linear Classifier:

$$\widehat{y}(x) \;=\; \text{sign}(\widehat{w}^\top x - \widehat{b})$$

This changes with number of training samples only if $(\widehat{w}, \widehat{b})$ change.

Kernel SVM:

$$\widehat{y}(x) \;=\; \text{sign}\left(\sum_i \widehat{c}^{(i)} y^{(i)} k(x^{(i)}, x) \;-\; \widehat{b}\right)$$

This sums up over all the support vectors, which (for kernels like RBF) are often all the data points.

Why use Kernels ?

Main advantage: Kernels can have model complexity and (hopefully) model **accuracy grow with dataset size.**

Linear Classifier:

$$\widehat{y}(x) \;=\; \text{sign}(\widehat{w}^\top x - \widehat{b})$$

This changes with number of training samples only if $(\widehat{w}, \widehat{b})$ change.

Kernel SVM:

$$\widehat{y}(x) \;=\; \text{sign}\left(\sum_i \widehat{c}^{(i)} y^{(i)} k(x^{(i)}, x) \;-\; \widehat{b}\right)$$

This sums up over all the support vectors, which (for kernels like RBF) are often all the data points.

Main disadvantage: Finding out the label for a new x involves computing time that grows with training set size.

Upshot: Kernel SVMs can (and often do) get more accurate when given bigger training set, but the cost of inference also grows...

Popular SVM Kernels

- Polynomial: $k(x_i, x_j) = (x_i^T x_j + c)^d$, c and d are constants.
- RBF : $k(x_i, x_j) = exp(-\gamma\|x_i - x_j\|^2), \gamma > 0$
- Hyperbolic Tangent: $k(x_i, x_j) = tanh(\kappa x_i^T x_j + c)$, κ and c are constants.
- Laplacian: $k(x_i, x_j) = exp(-\gamma\|x_i - x_j\|), \gamma > 0$, a constant
- MultiQuadric: $k(x_i, x_j) = \sqrt{\|x_i - x_j\|^2 + c^2}$, c is a constant
- Log: $k(x_i, x_j) = -\log(\|x_i - x_j\|^d + 1)$, d is a constant