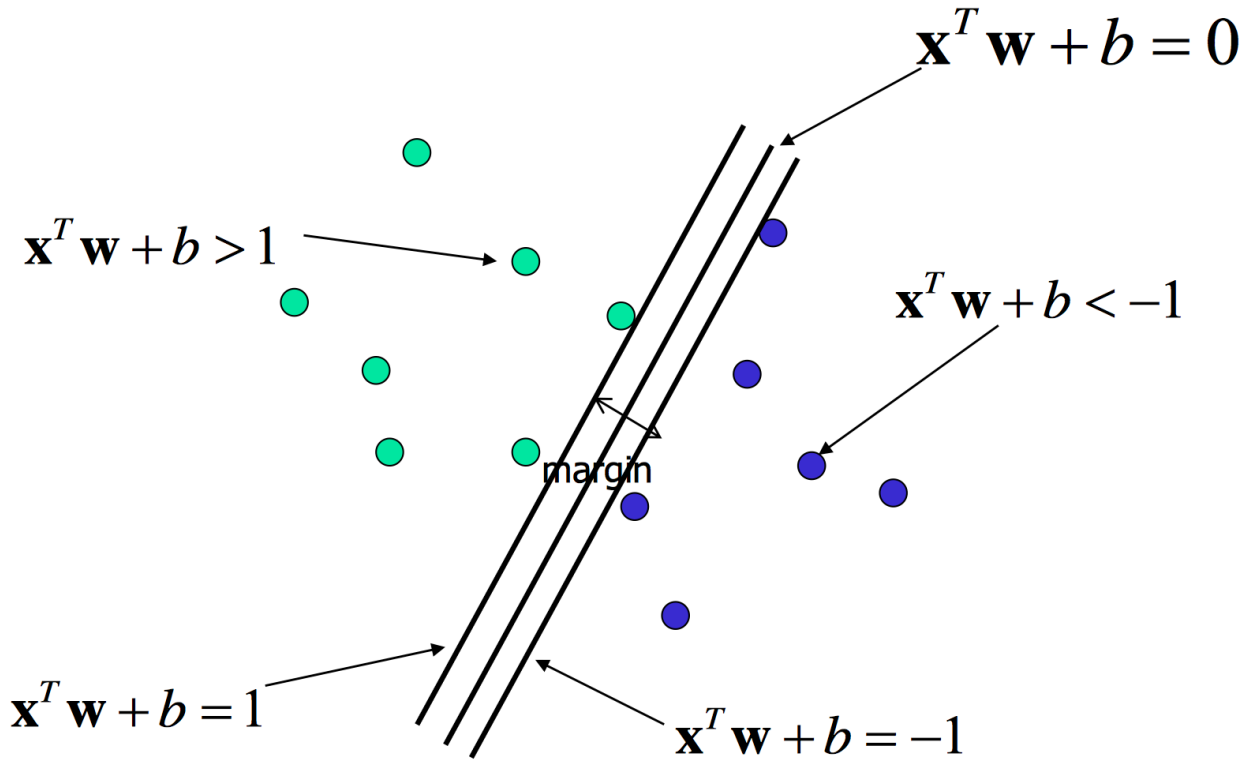
SVM

The **original** idea of SVM is to find a hyperplane ($X^T w + b = 0$) which could classify two classes as best as possible.



Best means that the hyperplane could maxmize the magin. The magin could be calculated according to the computation two parallels' distance.

$$margin = \frac{2}{||w||}$$

Therefore, we have the optimization problem:

$$max_{w,b} \frac{2}{||w||}$$
$$s.t.\, y_i(w^T x_i + b) \geq 1$$

This problem is equivalent to (for calculate convienence)

$$min_{w,b} \frac{1}{2}||w||^2$$
$$s.t.\, y_i(w^T x_i + b) \geq 1$$

1. How to solve the problem?

We could solve the problem by using method of lagrange multiplier. Firstly, we could get the lagrange function

$$L = \frac{1}{2}||w||^2 - \Sigma_{i=1}^{N} a_i(y_i(w^T x_i + b) - 1)$$
$$where\ a_i \geq 0$$

When the samples do not satisify the condition $y_i(w^T x_i + b) \geq 1$, then $max_a L(w, b, a) = +\infty$ while if the condition is satisified, $max_a L(w, b, a) = \frac{1}{2}||w||^2$. Therefore the original problem is equivalent to

$$min_{w,b} max_a L(w, b, a)$$

Here, in order to solve this problem convienently, we could transform this prime problem into its dual problem

$$max_a min_{w,b} L(w, b, a)$$

- Calculate $min_{w,b} L(w, b, a)$

As usual, making the partial deriative of $L(w, b, a)$ over w and b to be 0 could get the minimal value of $L(w, b, a)$. The results are

$$w = \Sigma_{i=1}^{N} a_i y_i x_i$$
$$\Sigma_{i=1}^{N} a_i y_i = 0$$

Then the minimal value of $L(w, b, a)$ is

$$\Sigma_{i=1}^{N} a_i - \frac{1}{2} \Sigma_{i=1}^{N} \Sigma_{j=1}^{N} a_i a_j y_i y_j x_i^T x_j$$

- Calculate $max_a min_{w,b} L(w, b, a)$

Now the optimization problem is

$$min_a\ \frac{1}{2} \Sigma_{i=1}^{N} \Sigma_{j=1}^{N} a_i a_j y_i y_j x_i^T x_j - \Sigma_{i=1}^{N} a_i$$
$$s.t.\ \Sigma_{i=1}^{N} a_i y_i = 0, a_i \geq 0$$

If we could get the solution of $\mathbf{a}$, we could get $\hat{w} = \Sigma_{i=1}^{N} \hat{a}_i y_i x_i$. Then we could get the model

$$f(\mathbf{x}) = \mathbf{x}^T w + b = \Sigma_{i=1}^{N} \hat{a}_i y_i \mathbf{x}_i^T \mathbf{x}_i + b$$

There is at least one j makes $a_j > 0$ and so we could solve $\hat{b} = \frac{1}{y_j} - \Sigma_{i=1}^{N} \hat{a}_i y_i \mathbf{x}_i^T \mathbf{x}_i$. Of course, we could choose arbitrary support vector(we'll cover it in next part) to calculate $\hat{b}$. However, in practise, for robustness sake, we would calculate the average value for $\hat{b}$ which is $\frac{1}{|S|} \Sigma_{s \in S} (\frac{1}{y_s} - \Sigma_{i \in S} a_i y_i x_i^T x_s)$ where S is the set of all support vectors.

- What is support vectors?

We actually have KKT condition for the prime problem

$$\begin{cases} a_i \geq 0 \\ y_i(w^T x_i + b) \geq 1 \\ a_i(y_i(w^T x_i + b) - 1) = 0 \end{cases}$$

Therefore, for (x_i, y_i), we always have $a_i = 0$ or $y_i(w^T x_i + b) - 1 = 0$. If $a_i = 0$, the sample won't affect $f(\mathbf{x})$ while $y_i(w^T x_i + b) - 1 = 0$ means that the sample is actually on the margin boundary which actually affects the performance of $f(\mathbf{x})$. These samples are called support vectors.

- Of course, how to get the solution of $\mathbf{a}$?

This is a problem of quadratic programming. We could solve it by using SMO(sequential minimal optimization).

2. Soft Margin

The first part is based on an assumption that our training dataset could be seperated linearly. However, if our data is approximate linear seperable, we could also use a trick called soft margin to solve it. Hard margin requires that $y_i(x_i^T w + b) \geq 1$ while soft margin allows some samples do not satisfy the condition. However, of course we hope the break-rule samples are as less as possible. Therefore, we could add regularization to make the objective function to be

$$min_{w,b} \frac{1}{2}||w||^2 + C\Sigma_{i=1}^N max(0, 1 - y_i(w^T x_i + b))$$

The larger C, the more penalty on misclassification. When C is close to infinity, the soft margin svm turns out to be hard margin svm. Here, if we introduce slack variables $\xi_i \geq 0$, then we could rewrite the objective function as

$$\begin{aligned} &min_{w,b,\xi} \ \frac{1}{2}||w||^2 + C\Sigma_{i=1}^N \xi_i \\ &s.t. \ y_i(x_i^T w + b) \geq 1 - \xi_i, \xi_i \geq 0 \end{aligned}$$

- Note:
  - Soft margin svm still keeps the property that only support vectors matters.
  - The loss used in soft margin svm is **Hinge loss** $max(0, 1 - z)$

3. Kernel Trick

What if the training dataset cannot be linear seperated completely? Now we need a trick called kernel. The idea is that we could map samples into a higher dimension in which the samples could be seperated linearly. Therefore, our goal turns out to find

$$f(\mathbf{x}) = \phi(\mathbf{x})^T w + b$$

Then the objective function could be

$$\begin{aligned} &min_a \ \frac{1}{2}\Sigma_{i=1}^N \Sigma_{j=1}^N a_i a_j y_i y_j \phi(x_i)^T \phi(x_j) - \Sigma_{i=1}^N a_i \\ &s.t. \ \Sigma_{i=1}^N a_i y_i = 0, a_i \geq 0 \end{aligned}$$

In higher dimension, it's hard to calculate the $\phi(x_i)^T \phi(x_j)$ directly. A natural idea is to define a kernel function which equals $\phi(x_i)^T \phi(x_j)$. Then the objective function could be written like

$$\begin{aligned} &min_a \ \frac{1}{2}\Sigma_{i=1}^N \Sigma_{j=1}^N a_i a_j y_i y_j \mathbf{k}(x_i, x_j) - \Sigma_{i=1}^N a_i \\ &s.t. \ \Sigma_{i=1}^N a_i y_i = 0, a_i \geq 0 \end{aligned}$$

According to the deriation in part 1, we could easily get the solution

$$f(\mathbf{x}) = \Sigma_{i=1}^{N} a_i y_i \mathbf{k}(x, x_i) + b$$

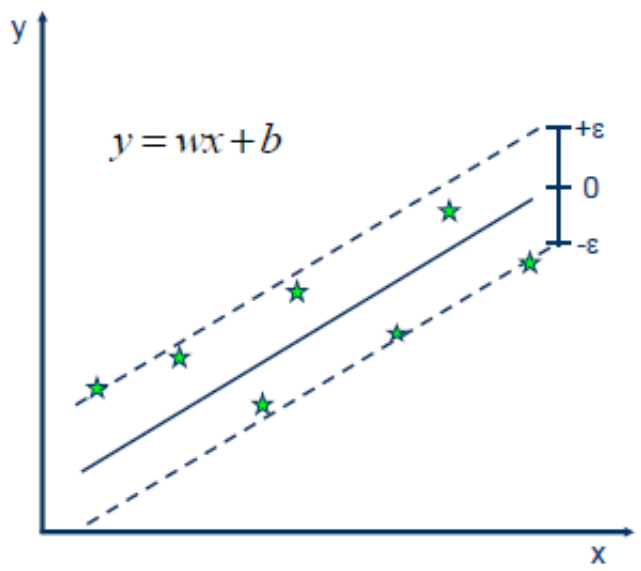
There are some kernel functions that are usually used:

- Linear kernel : $x_i^T x_j$
- Gaussian kernel: $exp(-\frac{||x_i - x_j||^2}{2\sigma^2})$
- Laplace kernel: $exp(-\frac{||x_i - x_j||^2}{\sigma})$
- Sigmoid kernel: $tanh(\beta x_i^T x_j + \theta)$

Note: we could also construct a new kernel function by linear combination of existing kernel functions.

4. SVM in Regression

In last three parts, we introduce SVM based on classification problem. However, svm could be applied in regression problem also.



- Solution:

$$\min \frac{1}{2}\|w\|^2$$

- Constraints:

$$y_i - wx_i - b \le \varepsilon$$
$$wx_i + b - y_i \le \varepsilon$$

5. Additionals

- Could SVM be used in multi-classification?

  Yes. How? We could use 1-versus-the-rest approach or one-versus-one approach to build SVMs for multi-classification.

- What's the differences between SVM and logistic regression?
  - Loss function: logistic is logorithm loss function while SVM with soft margin is hinge loss.
  - Samples: logistic considers all samples while svm cares only about the support vectors.
  - **Kernel**: for svm if the problem is not linear seperable, we usually use the kernel trick while in logistic we won't. That's because for svm, we only have to calculate kernel functions for support vectors rather than the whole set of samples. However, if we use the kernel trick in logistic regression, we have to calculate the kernel function between each pair of samples.
  - Normalization: distance computation plays an important role on SVM. Therefore, we

have to normalize our dataset first while logistic regression does not matter.
    - Regularization: SVM is the algorithm whose structure risk (balance between model complexity and training error) is minimal because its objective function has the L2 regularization($||w||^2$) naturally.

- Does it have linear or non-linear boundary?

  Without kernel, SVM has linear boundary. With kernel, it has non-linear boundary.