DDA3020 Machine Learning
Lecture 07 Support Vector Machine

JIA, Kui
School of Data Science, CUHK-SZ

October 10/12, 2023

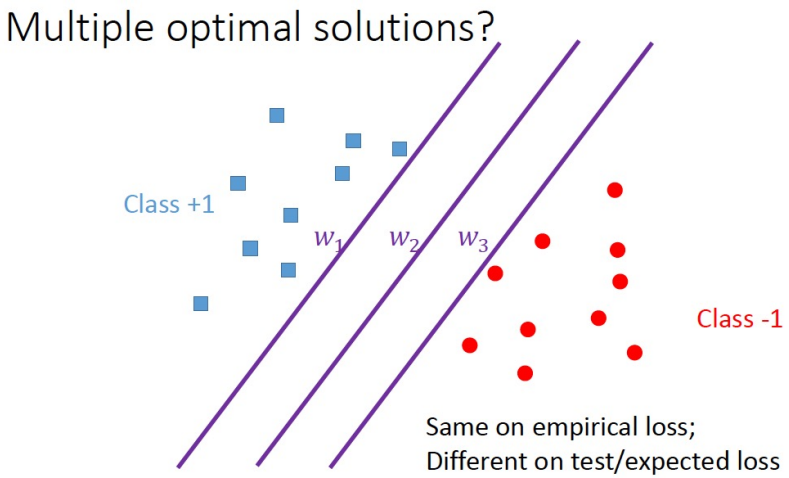
Outline

Classification

Binary classification:

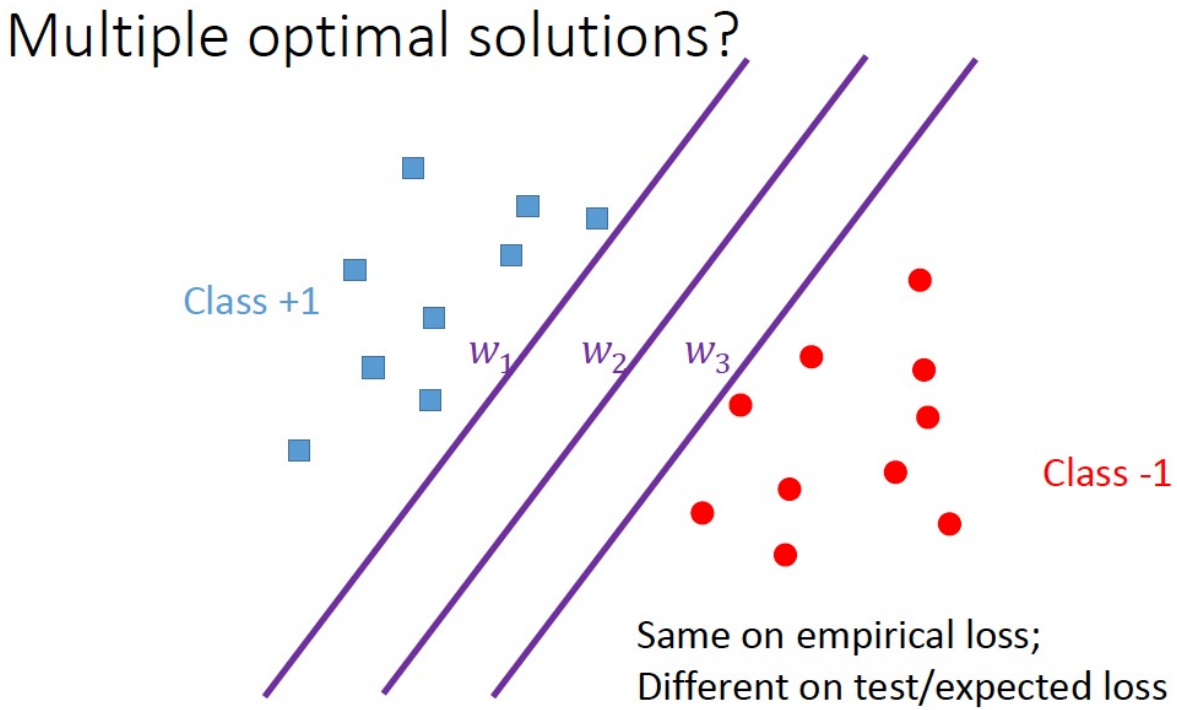
- Given training data set $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$, and $\mathbf{x}_i \in \mathbb{R}^n, y_i \in \{-1, +1\}$
- We adopt the sign hypothesis function $y = \text{Sgn}(f_{\mathbf{w}}(\mathbf{x})) = \text{Sgn}(\mathbf{w}^\top \mathbf{x})$
- Then, we require that
  - If $y_i = +1$, then $\mathbf{w}^\top \mathbf{x}_i > 0$
  - If $y_i = -1$, then $\mathbf{w}^\top \mathbf{x}_i < 0$

Classification



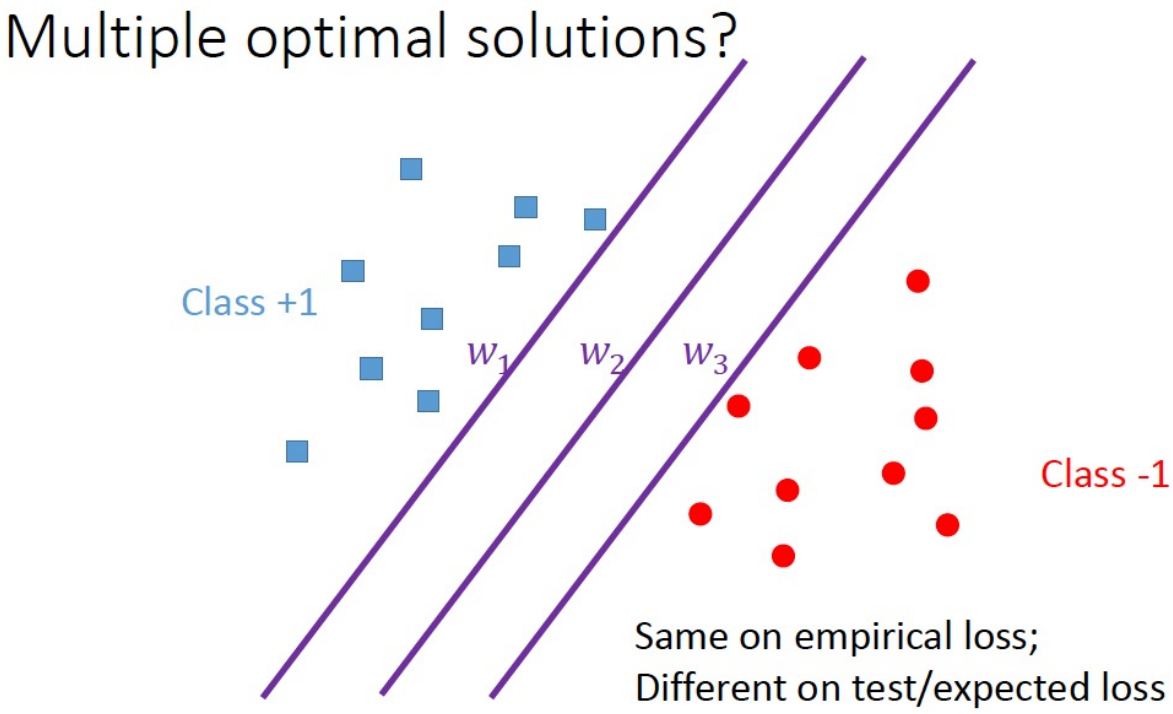
- There could be multiple decision boundaries to perfectly separate the above data. Why?
- For standard logistic regression, the objective function (*i.e.*, cross entropy loss) is convex, rather than strongly/strictly convex. Consequently, there are multiple values of parameters that can perfectly fit the training data.
- For regularized logistic regression, the objective function (*i.e.*, cross entropy loss + $\lambda \cdot \ell_2$ regularization) is strictly convex, which has the unique optimal solution. However, it depends on the trade-off hyper-parameter λ. For sure you can use cross-validation to use a suitable λ, but is there any more elegant approach?

Classification



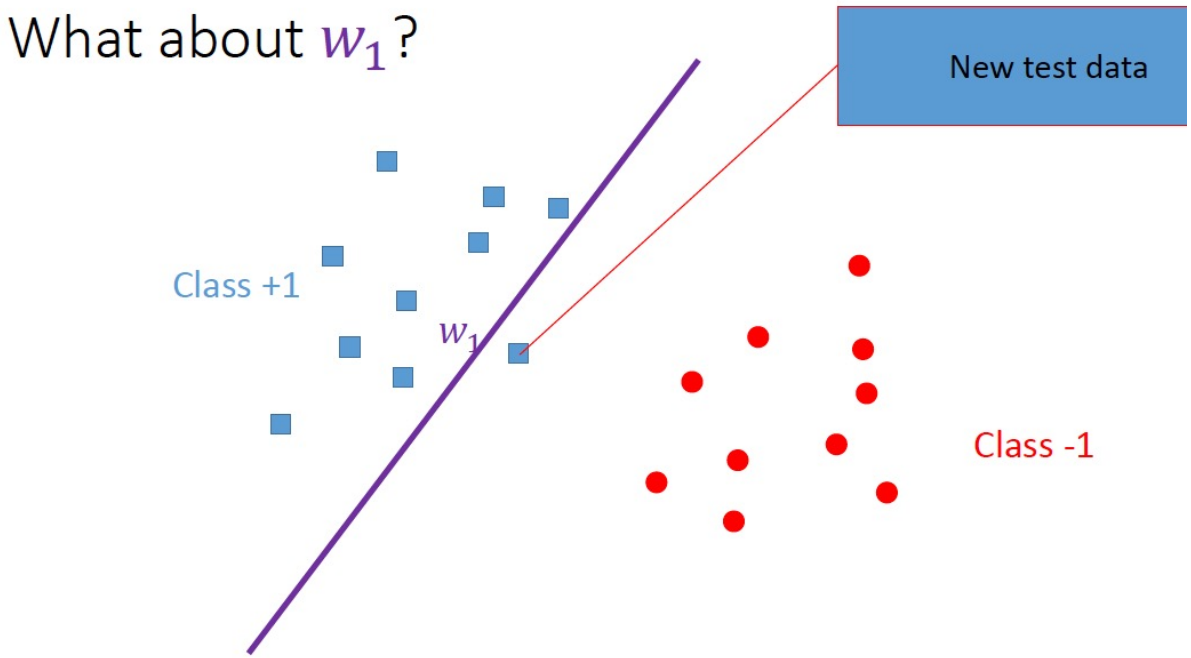
- Just following your intuition, which decision boundary do you prefer?

Classification



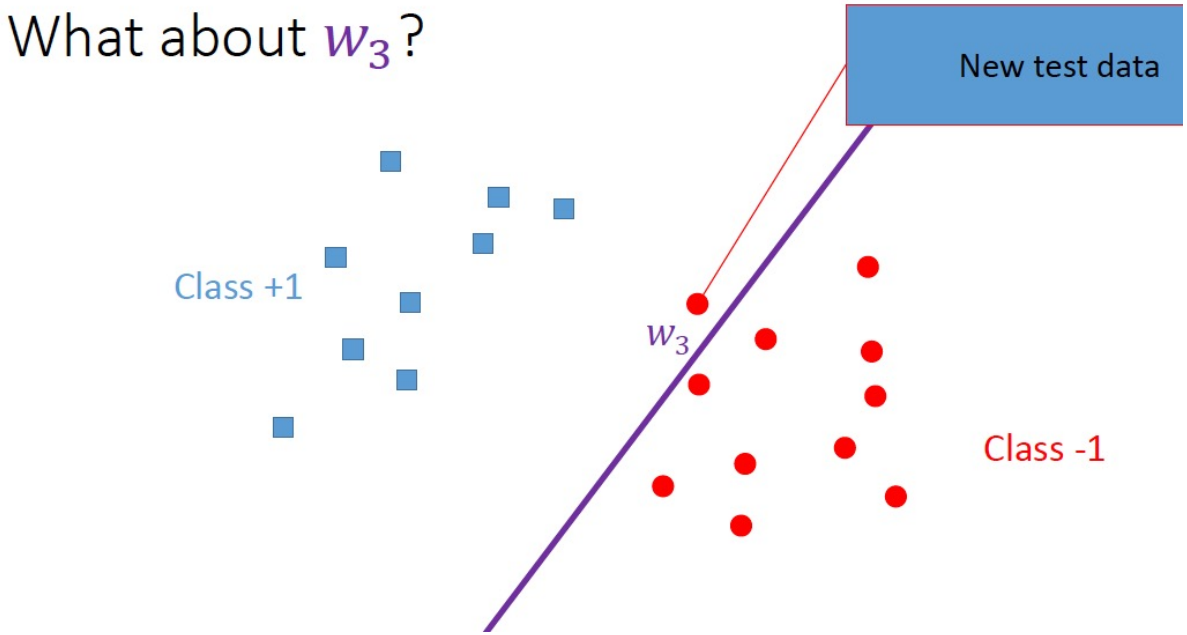
- Just following your intuition, which decision boundary do you prefer?
- The middle one (*i.e.*, $\mathbf{w}_2^\top \mathbf{x} = 0$) seems better, as it is far from data of both positive and negative classes.
- How to model such intuition?

Classification



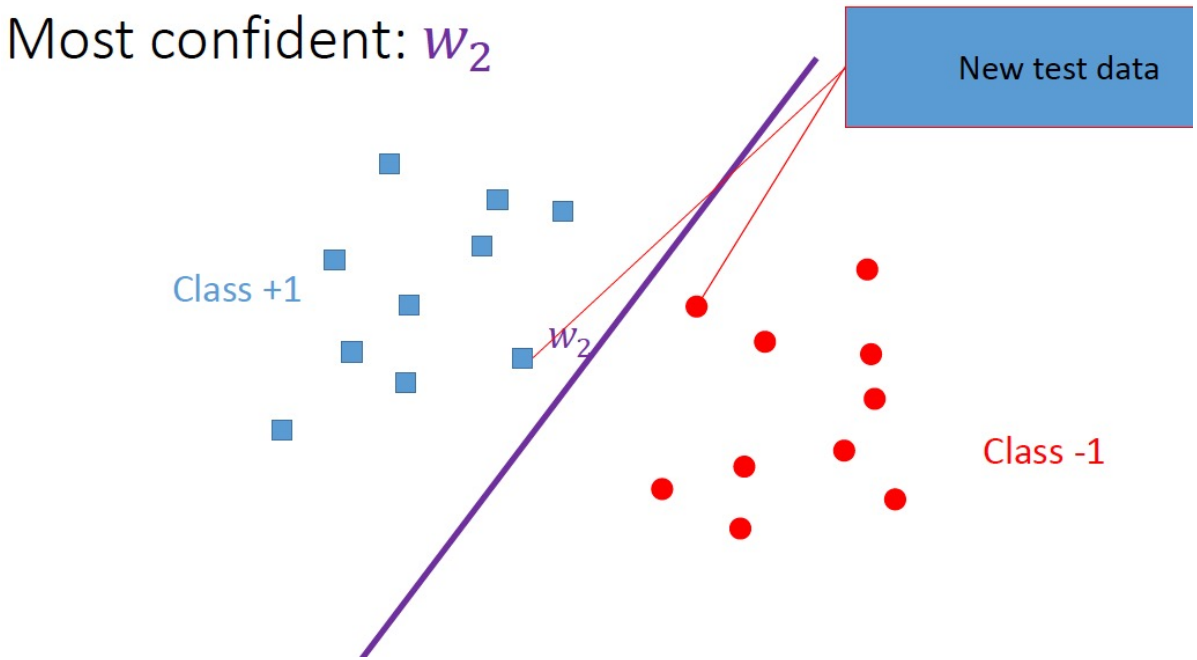
- Just following your intuition, which decision boundary do you prefer?
- The middle one (*i.e.*, $\mathbf{w}_2^\top \mathbf{x} = 0$) seems better, as it is far from data of both positive and negative classes.
- How to model such intuition?

Classification



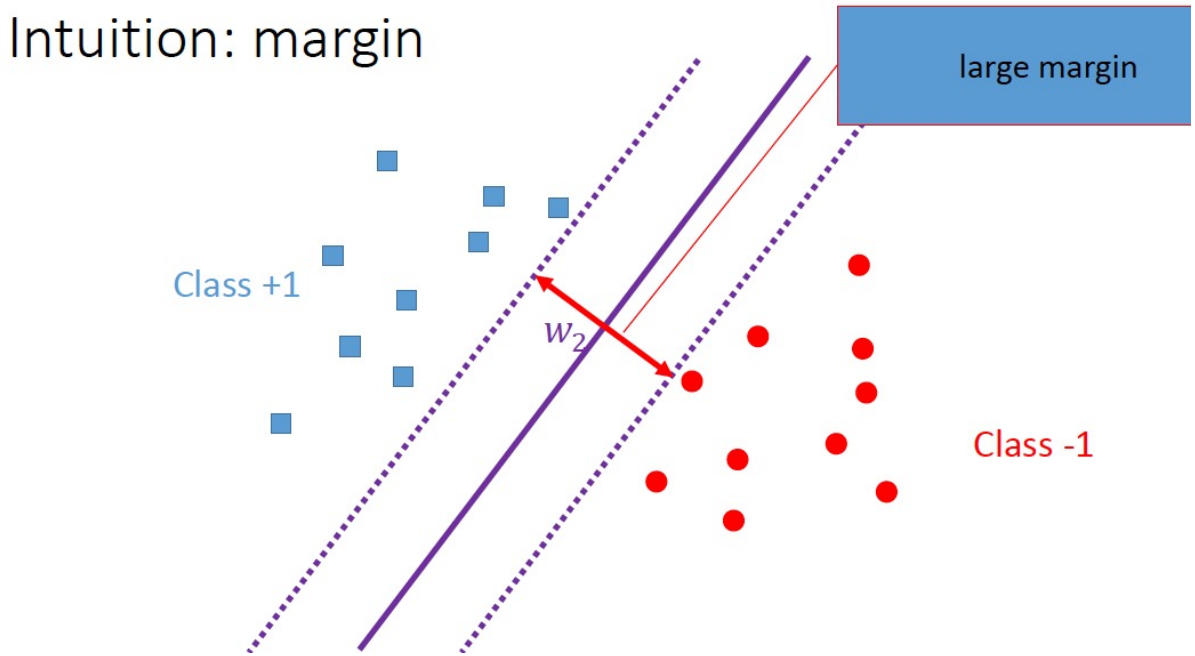
- Just following your intuition, which decision boundary do you prefer?
- The middle one (*i.e.*, $\mathbf{w}_2^\top \mathbf{x} = 0$) seems better, as it is far from data of both positive and negative classes.
- How to model such intuition?

Classification



- Just following your intuition, which decision boundary do you prefer?
- The middle one (*i.e.*, $\mathbf{w}_2^\top \mathbf{x} = 0$) seems better, as it is far from data of both positive and negative classes.
- How to model such intuition?

Large margin intuition



- We introduce the concept margin: the distance from the closest point of positive and negative classes to the decision boundary
- The intuition is to choose the decision boundary with large margin, which is called large margin classifier, also called support vector machine (SVM)

Mathematics behind large margin classification

Inner vector product:

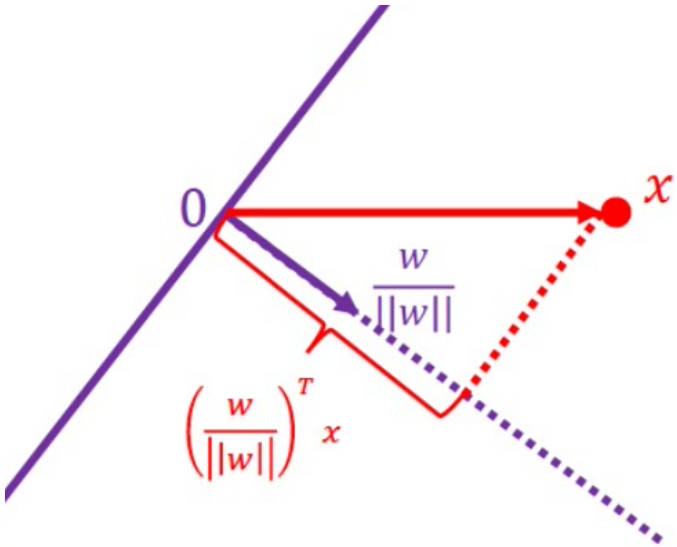
- $\boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \boldsymbol{\nu} = \begin{bmatrix} \nu_1 \\ \nu_2 \end{bmatrix}$
- $\|\boldsymbol{\mu}\| = \sqrt{\mu_1^2 + \mu_2^2}$, the length of $\boldsymbol{\mu}$
- $\boldsymbol{\mu}^\top \boldsymbol{\nu} = \mu_1\nu_1 + \mu_2\nu_2$. How to represent it in the above plot? Note: $\boldsymbol{\mu}^\top \boldsymbol{\nu} = \|\boldsymbol{\mu}\|\|\boldsymbol{\nu}\| \cos\theta$
- $\boldsymbol{\mu}^\top \boldsymbol{\nu} = p \cdot \|\boldsymbol{\mu}\|$, where p is the length of projection of $\boldsymbol{\nu}$ on $\boldsymbol{\mu}$
- Note that if the angle between $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ is larger than $90°$, then $p < 0$

Mathematics behind large margin classification

Lemma 1: $\mathbf{x}$ has distance $\frac{|f_{\mathbf{w}}(\mathbf{x})|}{\|\mathbf{w}\|}$ to the the hyperplane $f_{\mathbf{w}}(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} = 0$

Proof:

1. $\mathbf{w}$ is orthogonal to the hyperplane, as $\mathbf{w}^\top(\mathbf{x}_1 - \mathbf{x}_2) = 0$ for any two points $\mathbf{x}_1, \mathbf{x}_2$ at the hyperplane
2. The unit direction is $\frac{\mathbf{w}}{\|\mathbf{w}\|}$
3. The projection of $\mathbf{x}$ is $\left(\frac{\mathbf{w}}{||\mathbf{w}||}\right)^\top \mathbf{x} = \frac{f_{\mathbf{w}}(\mathbf{x})}{\|\mathbf{w}\|}$



Mathematics behind large margin classification

Claim 1: $\mathbf{w}$ is orthogonal to the hyperplane $f_{\mathbf{w},b}(x) = \mathbf{w}^\top \mathbf{x} + b = 0$

Proof:

1. pick any $\mathbf{x}_1$ and $\mathbf{x}_2$ on the hyperplane
2. $\mathbf{w}^\top \mathbf{x}_1 + b = 0$
3. $\mathbf{w}^\top \mathbf{x}_2 + b = 0$
4. So $\mathbf{w}^\top (\mathbf{x}_1 - \mathbf{x}_2) = 0$

Mathematics behind large margin classification

Claim 2: $\mathbf{0}$ has distance $\frac{-b}{\|\mathbf{w}\|}$ to the hyperplane $\mathbf{w}^\top \mathbf{x} + b = 0$

Proof:

1. pick any $\mathbf{x}_1$ on the hyperplane
2. Project $\mathbf{x}_1$ to the unit direction $\frac{\mathbf{w}}{\|\mathbf{w}\|}$ to get the distance
3. $\left(\frac{\mathbf{w}}{\|\mathbf{w}\|}\right)^\top \mathbf{x}_1 = \frac{-b}{\|\mathbf{w}\|}$ since $\mathbf{w}^\top \mathbf{x}_1 + b = 0$
4. The projection length of $\mathbf{x}_1$ to $\frac{\mathbf{w}}{\|\mathbf{w}\|}$ is equivalent to the distance from $\mathbf{0}$ to the hyperplane, *i.e.*, $\frac{-b}{\|\mathbf{w}\|}$

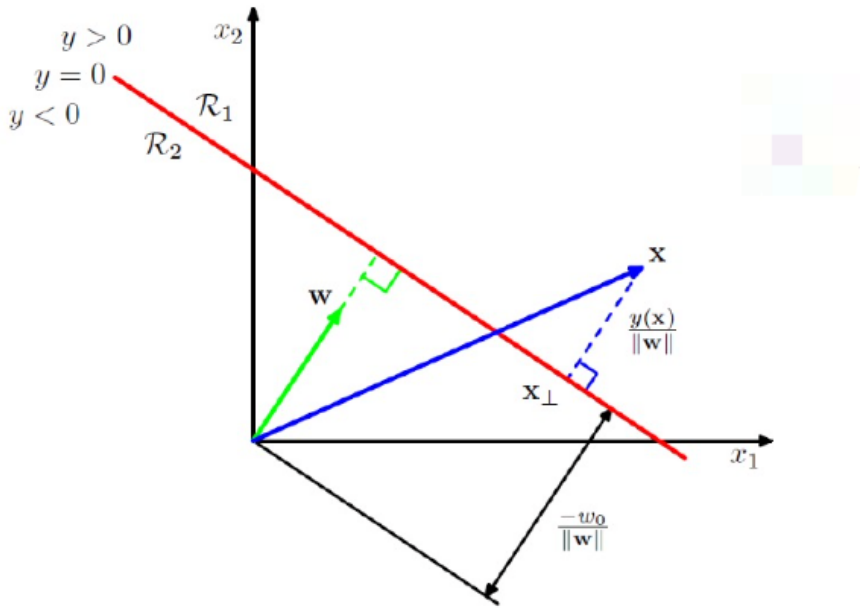
Mathematics behind large margin classification

Lemma 2: $\mathbf{x}$ has distance $\frac{|f_{\mathbf{w},b}(\mathbf{x})|}{\|\mathbf{w}\|}$ to the hyperplane $f_{\mathbf{w},b}(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b = 0$

Proof:

1. Let $\mathbf{x} = \mathbf{x}_\perp + r\frac{\mathbf{w}}{\|\mathbf{w}\|}$, then $|r|$ is the distance
2. Multiply both sides by $\mathbf{w}^\top$ and add b
3. Left hand side: $\mathbf{w}^\top \mathbf{x} + b = f_{\mathbf{w},b}(\mathbf{x})$
4. Right hand side: $\mathbf{w}^\top \mathbf{x}_\perp + r\frac{\mathbf{w}^\top \mathbf{w}}{\|\mathbf{w}\|} + b = 0 + r\|\mathbf{w}\|$
5. Thus, $f_{\mathbf{w},b}(\mathbf{x}) = r\|\mathbf{w}\|$. We obtain $|r| = \frac{|f_{\mathbf{w},b}(\mathbf{x})|}{\|\mathbf{w}\|}$.

The notation here is: $y(\mathbf{x}) = f_{\mathbf{w},b}(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + w_0, b = w_0$.



Mathematics behind large margin classification

Margin over all training data points:

$$\gamma = \min_i \frac{|f_{\mathbf{w},b}(\mathbf{x}_i)|}{\|\mathbf{w}\|}$$

Since only want correct $f_{\mathbf{w},b}$, and recall $y_i \in \{+1, -1\}$, we have

$$\gamma = \min_i \frac{y_i f_{\mathbf{w},b}(\mathbf{x}_i)}{\|\mathbf{w}\|}$$

If $f_{\mathbf{w},b}$ incorrect on some $\mathbf{x}_i$, the margin is negative

Mathematics behind large margin classification

- Maximize margin over all training data points:

$$\max_{\mathbf{w},b} \gamma = \max_{\mathbf{w},b} \min_{i} \frac{y_i f_{\mathbf{w},b}\left(\mathbf{x}_i\right)}{\|\mathbf{w}\|} = \max_{\mathbf{w},b} \min_{i} \frac{y_i \left(\mathbf{w}^\top \mathbf{x}_i + b\right)}{\|\mathbf{w}\|}$$

- A bit complicated ...

Mathematics behind large margin classification

- Observation: when $(\mathbf{w}, b)$ scaled by a factor c, the margin unchanged

$$\frac{y_i \left(c\mathbf{w}^\top \mathbf{x}_i + cb\right)}{\|c\mathbf{w}\|} = \frac{y_i \left(\mathbf{w}^\top \mathbf{x}_i + b\right)}{\|\mathbf{w}\|}$$

- Let's consider a fixed scale such that

$$y_{i^*} \left(\mathbf{w}^\top \mathbf{x}_{i^*} + b\right) = 1$$

where $\mathbf{x}_{i^*}$ is the point closest to the hyperplane

Mathematics behind large margin classification

- Let's consider a fixed scale such that

$$y_{i^*}\left(\mathbf{w}^\top \mathbf{x}_{i^*} + b\right) = 1$$

where $\mathbf{x}_{i^*}$ is the point closet to the hyperplane - Now we have for all data

$$y_i\left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geq 1$$

and at least for one i the equality holds - Then the margin is $\frac{1}{\|\mathbf{w}\|}$

Mathematics behind large margin classification

- Maximize margin over all training data points:

$$\max_{\mathbf{w},b} \gamma = \max_{\mathbf{w},b} \min_{i} \frac{y_i f_{\mathbf{w},b}(\mathbf{x}_i)}{\|\mathbf{w}\|} = \max_{\mathbf{w},b} \min_{i} \frac{y_i \left(\mathbf{w}^\top \mathbf{x}_i + b\right)}{\|\mathbf{w}\|}$$

- Utilizing $y_{i^*}\left(\mathbf{w}^\top \mathbf{x}_{i^*} + b\right) = 1$, the above optimization is simplified to

$$\begin{array}{rl} \min_{\mathbf{w},b} & \frac{1}{2}||\mathbf{w}||^2 \\ \text{subject to} & y_i \left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geq 1, \forall i \end{array}$$

- **Training/learning**: solving the above optimization problem is called training or learning of the large margin classifier, and we obtain the solution $\mathbf{w}^*, b^*$
- **Prediction**: given the solution $\mathbf{w}^*, b^*$, for a new test data $\boldsymbol{x}_t$, we predict it as $+\mathbf{1}$ if $(\mathbf{w}^*)^\top \mathbf{x}_t + b^* > 0$, otherwise $-\mathbf{1}$.

Alternative view of logistic regression

- Hypothesis function:

$$f_{\mathbf{w},b}(\mathbf{x}) = \frac{1}{1+\exp(-\mathbf{w}^\top \mathbf{x})} = g(z)$$

  where $z = \mathbf{w}^\top \mathbf{x}$
- If $y = 1$, we want $f_{\mathbf{w},b}(\mathbf{x}) \approx 1$, *i.e.*, $\mathbf{w}^\top \mathbf{x} \gg 0$
- If $y = -1$, we want $f_{\mathbf{w},b}(\mathbf{x}) \approx 0$, *i.e.*, $\mathbf{w}^\top \mathbf{x} \ll 0$
- Objective function of logistic regression

$$J(\mathbf{w}) = -\delta_{y=1} \log(f_{\mathbf{w},b}(\mathbf{x})) - \delta_{y=-1} \log(1 - f_{\mathbf{w},b}(\mathbf{x})), \tag{1}$$

  where $\delta_a = 1$ if a is true, otherwise 0.

Objective of SVM

- Objective function of regularized logistic regression

$$\frac{1}{m}\sum_{i}^{m}\left[\delta_{y_i=1}\big(-\log(f_{\mathbf{w},b}(\mathbf{x}_i))\big)+\delta_{y_i=-1}\big(-\log(1-f_{\mathbf{w},b}(\mathbf{x}_i))\big)\right]+\frac{\lambda}{2m}\sum_{j=1}^{n}w_j^2$$

- Objective function of support vector machine

$$\frac{1}{m}\sum_{i}^{m}\left[\delta_{y_i=1}\text{cost}_1(\mathbf{w}^\top\mathbf{x}_i+b)+\delta_{y_i=-1}\text{cost}_{-1}(\mathbf{w}^\top\mathbf{x}_i+b)\right]+\frac{\lambda}{2m}\sum_{j=1}^{n}w_j^2$$

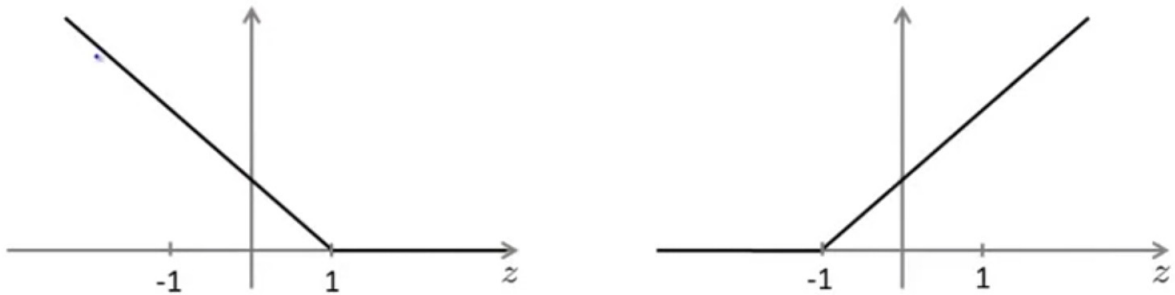
$$\equiv C\sum_{i}^{m}\left[\delta_{y_i=1}\text{cost}_1(\mathbf{w}^\top\mathbf{x}_i+b)+\delta_{y_i=-1}\text{cost}_{-1}(\mathbf{w}^\top\mathbf{x}_i+b)\right]+\frac{1}{2}\sum_{j=1}^{n}w_j^2,$$

where $C=\frac{1}{\lambda}$.

Objective of SVM

- Objective function of support vector machine

$$C\sum_{i}^{m}\left[\delta_{y_i=1}\text{cost}_1(\mathbf{w}^\top\mathbf{x}_i+b)+\delta_{y_i=-1}\text{cost}_{-1}(\mathbf{w}^\top\mathbf{x}_i+b)\right]+\frac{1}{2}\sum_{j=1}^{n}w_j^2$$



- If $y_i=+1$, we require that $\mathbf{w}^\top\mathbf{x}_i+b\geq 1$. In other words, $\text{cost}_1(\mathbf{w}^\top\mathbf{x}_i+b)=0$ if $\mathbf{w}^\top\mathbf{x}_i+b\geq 1$
- If $y_i=-1$, we require that $\mathbf{w}^\top\mathbf{x}_i+b\leq -1$. In other words, $\text{cost}_{-1}(\mathbf{w}^\top\mathbf{x}_i+b)=0$ if $\mathbf{w}^\top\mathbf{x}_i+b\leq -1$
- **Hinge loss**:

$$\max\left(0,1-y_i(\mathbf{w}^\top\mathbf{x}_i+b)\right) \tag{2}$$

Mathematics behind large margin classification

- However, hinge loss is non-smooth. We transform the objective function of support vector machine to the following

$$\min_{\mathbf{w},b} \frac{1}{2}\sum_{j=1}^{n} w_j^2 \tag{3}$$
$$s.t.\ \mathbf{w}^\top \mathbf{x}_i + b \geq 1, \text{if } y_i = 1;\ \mathbf{w}^\top \mathbf{x}_i + b < -1, \text{if } y_i = -1.$$

- It can be simplified as follows

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \tag{4}$$
$$s.t.\ y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1, \forall i$$

Mathematics behind large margin classification

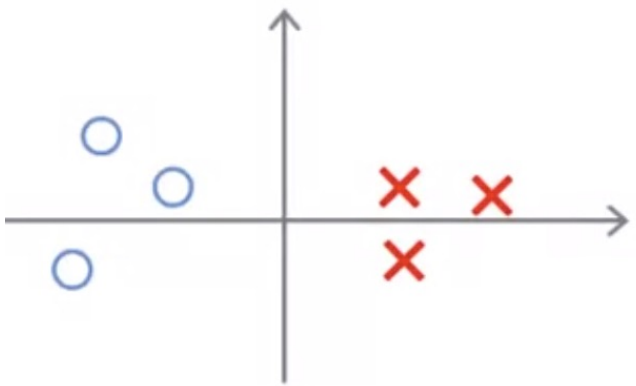
- Utilizing $p_i = \frac{\mathbf{w}^\top \mathbf{x}_i + b}{\|\mathbf{w}\|}$, which denotes the projection length of $\mathbf{x}_i$ on $\mathbf{w}$ or the distance from $\mathbf{x}_i$ to the decision boundary $\mathbf{w}^\top \mathbf{x} + b = 0$, we have

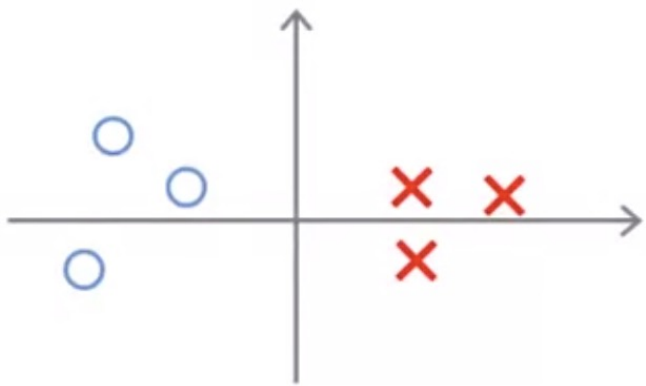
$$\mathbf{w}^\top \mathbf{x}_i + b = p_i \cdot \|\mathbf{w}\| \tag{5}$$

- The objective function of support vector machine is transformed to

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \tag{6}$$
$$s.t.\ y_i \cdot p_i \cdot \|\mathbf{w}\| \geq 1, \forall i$$

- Let's see the following two decision boundaries (plot below)
- If the projection length p_i is larger, then $\|\mathbf{w}\|$ could be smaller, leading to better solution. Thus, **we prefer large margin**.





Lagrange duality

- Given a general minimization problem

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & h_i(\mathbf{x}) \leq 0, \quad i = 1, \ldots, m \\ & \ell_j(\mathbf{x}) = 0, \quad j = 1, \ldots, r \end{aligned}$$

  Note that here $\mathbf{x}$ denotes the argument we aim to optimize, rather than a data point.
- The **Lagrangian function**:

$$L(\mathbf{x}, \mathbf{u}, \mathbf{v}) = f(\mathbf{x}) + \sum_{i=1}^{m} u_i h_i(\mathbf{x}) + \sum_{j=1}^{r} v_j \ell_j(\mathbf{x})$$

- The **Lagrange dual function**:

$$g(\mathbf{u}, \mathbf{v}) = \min_{\mathbf{x}\in\mathbb{R}^n} L(\mathbf{x}, \mathbf{u}, \mathbf{v})$$

- The **dual problem**:

$$\begin{aligned} & \max_{\mathbf{u}\in\mathbb{R}^m, \mathbf{v}\in\mathbb{R}^r} g(\mathbf{u}, \mathbf{v}) \\ & \text{subject to } \mathbf{u} \geq 0 \end{aligned}$$

KKT conditions

- Given general problem

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & h_i(\mathbf{x}) \leq 0, \quad i = 1, \ldots, m \\ & \ell_j(\mathbf{x}) = 0, \quad j = 1, \ldots, r \end{aligned}$$

- The **Karush-Kuhn-Tucker conditions** or **KKT conditions** are:
  - $0 \in \partial f(\mathbf{x}) + \sum_{i=1}^{m} u_i \partial h_i(\mathbf{x}) + \sum_{j=1}^{r} v_j \partial \ell_j(\mathbf{x})$ (stationarity)
  - $u_i \cdot h_i(\mathbf{x}) = 0$ for all i (complementary slackness)
  - $h_i(\mathbf{x}) \leq 0, \ell_j(\mathbf{x}) = 0$ for all i, j (primal feasibility)
  - $u_i \geq 0$ for all i (dual feasibility)

Reference: S. Boyd and L. Vandenberghe (2004), Convex Optimization, Cambridge University Press, Chapter 5.

Optimization of SVM

- The objective function of support vector machine is

$$\begin{aligned} &\min_{\mathbf{w},b} \ \frac{1}{2}\|\mathbf{w}\|^2 \\ &s.t. \ y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1, \forall i \end{aligned} \tag{7}$$

- It can be transformed to

$$\begin{aligned} &\min_{\mathbf{w},b} \ \frac{1}{2}\|\mathbf{w}\|^2 \\ &s.t. \ 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b) \leq 0, \forall i \end{aligned} \tag{8}$$

- Its Lagrange function is

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i}^{m} \alpha_i \big(1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b)\big),$$

Optimization of SVM

- Lagrange function:

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 + \sum_i^m \alpha_i \big(1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b)\big),$$

- The primal and dual optimal solutions should satisfy KKT conditions:
  - Stationarity:

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \;\Rightarrow\; \mathbf{w} = \sum_i^m \alpha_i y_i \mathbf{x}_i \tag{9}$$

$$\frac{\partial L}{\partial b} = 0 \;\Rightarrow\; \sum_i^m \alpha_i y_i = 0 \tag{10}$$

  - Feasibility:

$$\alpha_i \geq 0,\; 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b) \leq 0,\; \forall i \tag{11}$$

  - Complementary slackness:

$$\alpha_i \big(1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b)\big) = 0,\; \forall i \tag{12}$$

Optimization of SVM

- Lagrange function:

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 + \sum_i^m \alpha_i \big(1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b)\big),$$

- Replacing the stationary condition into Lagrange function, we have

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) \tag{13}$$

$$= \frac{1}{2}\|\mathbf{w}\|^2 + \sum_i^m \alpha_i - \sum_i^m \alpha_i y_i \Big(\sum_j^m \alpha_j y_j \mathbf{x}_j\Big)^\top \mathbf{x}_i - \sum_i^m \alpha_i y_i b \tag{14}$$

$$= \frac{1}{2}\|\mathbf{w}\|^2 + \sum_i^m \alpha_i - \sum_{i,j}^m \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j - b\sum_i^m \alpha_i y_i \tag{15}$$

$$= \sum_i^m \alpha_i - \frac{1}{2}\|\mathbf{w}\|^2 \tag{16}$$

$$= \sum_i^m \alpha_i - \frac{1}{2}\sum_{i,j}^m \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j \tag{17}$$

Optimization of SVM

- Then, we obtain the following dual problem:

$$\max_{\boldsymbol{\alpha}} \sum_{i}^{m} \alpha_i - \frac{1}{2} \sum_{i,j}^{m} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j, \tag{18}$$

$$s.t. \sum_{i}^{m} \alpha_i y_i = 0, \ \alpha_i \geq 0, \ \forall i \tag{19}$$

It can be solved by any off-the-shelf optimization solver.

- Then, we replace the solved $\boldsymbol{\alpha}$ back into the stationary condition, thus we obtain the primal solution $\mathbf{w}$,

$$\mathbf{w} = \sum_{i}^{m} \alpha_i y_i \mathbf{x}_i \tag{20}$$

Optimization of SVM

Solution interpretation:

- The primal solution $\mathbf{w}$ and the dual solution $\boldsymbol{\alpha}$ should also satisfy other KKT conditions
  - Feasibility: $\alpha_i \geq 0,\ 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b) \leq 0,\ \forall i$
  - Complementary slackness: $\alpha_i\big(1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b)\big) = 0,\ \forall i$
- When comparing above conditions together, we have that for $\mathbf{x}_i,\ \forall i$,
  - If it satisfies $1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b) < 0$, then $\alpha_i = 0$;
  - If it satisfies $1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b) = 0$, then $\alpha_i \geq 0$.
- If $\alpha_i = 0$, then it means that $\mathbf{x}_i$ doesn't contribute to $\mathbf{w}$, *i.e.*, the SVM classifier
- The data points with $\alpha_i > 0$ construct the classifier, and they are called support vectors, which locate at the hyperplanes $y_i(\mathbf{w}^\top \mathbf{x}_i + b) = 1$. And, we define the support set as $\mathcal{S} = \{i | \alpha_i > 0\}$ This is why we call it support vector machine.

Optimization of SVM

The remaining issue is **how to determine the bias parameter b?**

- For any support vector $\mathbf{x}_j, j \in \mathcal{S}$, we have

$$y_j(\mathbf{w}^\top \mathbf{x}_j + b) = 1, \ \forall j \in \mathcal{S} \tag{21}$$

$$\Rightarrow y_j(\sum_i^m \alpha_i y_i \mathbf{x}_i^\top \mathbf{x}_j + b) = 1, \ \forall j \in \mathcal{S} \tag{22}$$

- Product y_j for both sides of the above equation, and utilizing $y_j \cdot y_j = 1$, we have

$$\sum_i^m \alpha_i y_i \mathbf{x}_i^\top \mathbf{x}_j + b = y_j, \ \forall j \in \mathcal{S} \tag{23}$$

$$\Rightarrow b = \frac{1}{|\mathcal{S}|} \sum_{j \in \mathcal{S}} \left(y_j - \sum_i^m \alpha_i y_i \mathbf{x}_i^\top \mathbf{x}_j\right) \tag{24}$$

Prediction using SVM

Prediction:

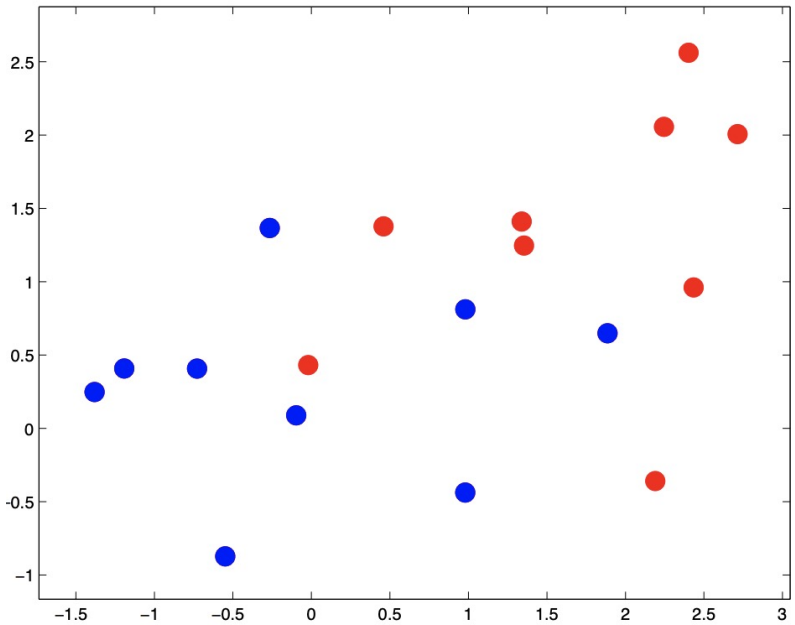
- Given the optimized parameters $\{\boldsymbol{\alpha}, \mathbf{w}, b\}$, given a new data $\mathbf{x}$, its prediction is

$$\mathbf{w}^\top \mathbf{x} + b = \sum_i^m \alpha_i y_i \mathbf{x}_i^\top \mathbf{x} + \frac{1}{|\mathcal{S}|} \sum_{j \in \mathcal{S}} \left(y_j - \sum_i^m \alpha_i y_i \mathbf{x}_i^\top \mathbf{x}_j \right) \tag{25}$$

- If $\mathbf{w}^\top \mathbf{x} + b > 0$, then the predicted class of $\mathbf{x}$ is $+1$, otherwise -1
- If and only if $y(\mathbf{w}^\top \mathbf{x} + b) > 0$, then your prediction is correct
- Note that the prediction of new data depends on inner product with existing training data, which is important to derive **kernel SVM** later

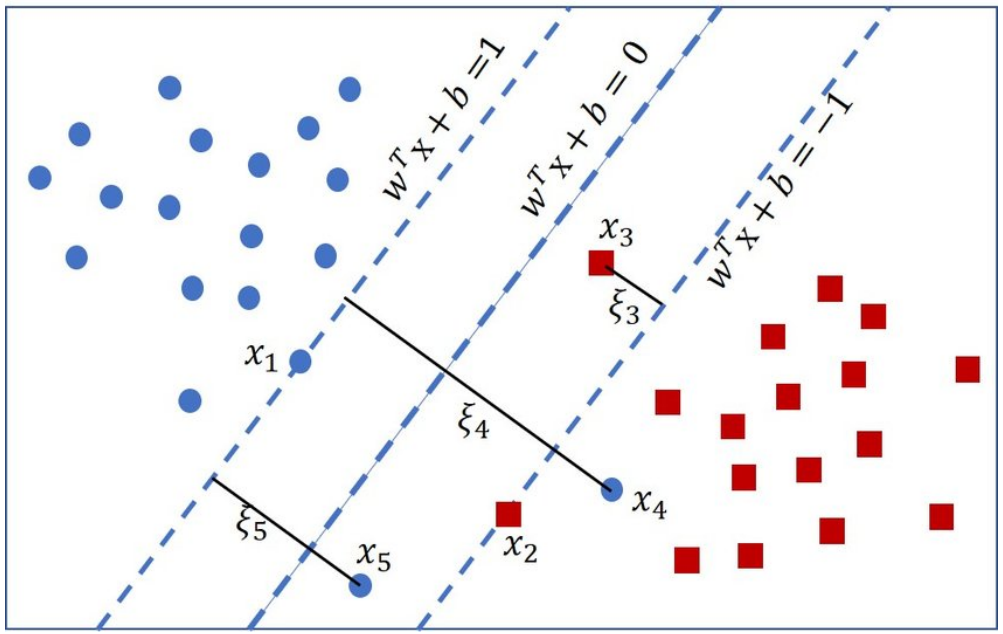
SVM with slack variables

- In above derivation, we assume that all primal constraints $y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1, \forall i$ can be satisfied, implying that the training data is separable.
- However, sometimes samples of different classes are overlapped (*i.e.*, non-separable), as shown below.
- Consequently, some constraints will be violated, and we can not obtain the feasible solution.



SVM with slack variables

- To handle such data, we introduce slack variable $\xi_i \geq 0$
- We allow some errors for training data, *i.e.*, $y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i, \forall i$, rather than $y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1, \forall i$
- But we hope that such errors $\xi_i, \forall i$ are small
- Please plot the corresponding hinge loss with slack variables



SVM with slack variables

- In this case, the SVM is formulated as follows

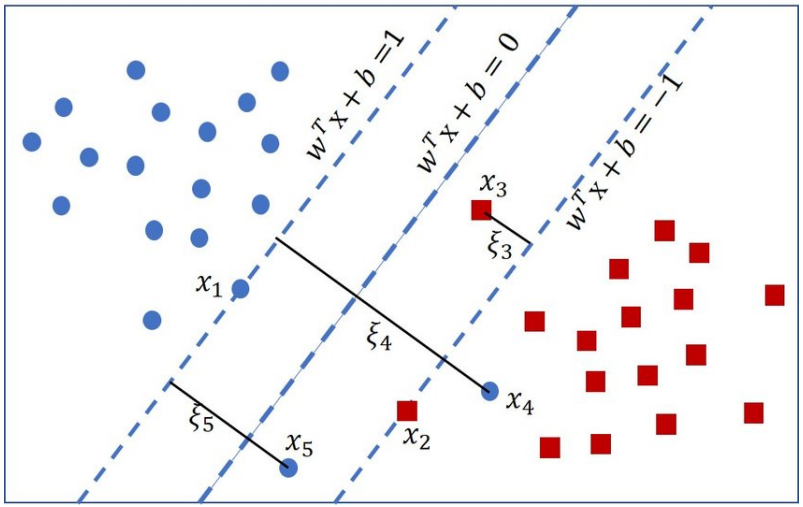
$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i^m \xi_i \tag{26}$$

$$s.t.\ 1-\xi_i - y_i(\mathbf{w}^\top \mathbf{x}_i + b) \leq 0, -\xi_i \leq 0,\ \forall i$$

- Its Lagrange function is

$$\mathcal{L}(\mathbf{w}, b, , \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i^m \xi_i + \sum_i^m \left[\alpha_i\left(1-\xi_i - y_i(\mathbf{w}^\top \mathbf{x}_i + b)\right) + \mu_i(-\xi_i)\right],$$

and $\alpha_i, \mu_i \geq 0, \forall i$.



SVM with slack variables

- Lagrange function:

$$\mathcal{L}(\mathbf{w}, b, , \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i^m \xi_i + \sum_i^m \left[\alpha_i\big(1-\xi_i - y_i(\mathbf{w}^\top \mathbf{x}_i + b)\big) + \mu_i(-\xi_i)\right],$$

- The primal and dual optimal solutions should satisfy KKT conditions:
  - **Stationarity**:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = 0 \;\Rightarrow\; \mathbf{w} = \sum_i^m \alpha_i y_i \mathbf{x}_i \tag{27}$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \;\Rightarrow\; \sum_i^m \alpha_i y_i = 0 \tag{28}$$

$$\frac{\partial \mathcal{L}}{\partial \xi_i} = 0 \;\Rightarrow\; \alpha_i = C - \mu_i,\ \forall i \tag{29}$$

  - **Feasibility**:

$$\alpha_i \geq 0,\ 1-\xi_i - y_i(\mathbf{w}^\top \mathbf{x}_i + b) \leq 0,\ \xi_i \geq 0, \mu_i \geq 0,\ \forall i \tag{30}$$

  - **Complementary slackness**:

$$\alpha_i\big(1-\xi_i - y_i(\mathbf{w}^\top \mathbf{x}_i + b)\big) = 0,\ \mu_i\xi_i = 0,\ \forall i \tag{31}$$

SVM with slack variables

- Lagrange function:

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i}^{m}\xi_i + \sum_{i}^{m}\left[\alpha_i\left(1-\xi_i - y_i(\mathbf{w}^\top \mathbf{x}_i + b)\right) + \mu_i(-\xi_i)\right].$$

- Replacing all stationary conditions into Lagrange function to eliminate primal variables, we have

$$\begin{aligned}\mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\mu}) &= \frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i}^{m}\left[\alpha_i\left(1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b)\right)\right] + \sum_{i}^{m}(C - \alpha_i - \mu_i)\xi_i \qquad (32)\\ &= \sum_{i}^{m}\alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j.\end{aligned}$$

SVM with slack variables

- Then, we obtain the following dual problem:

$$\max_{\boldsymbol{\alpha},\boldsymbol{\mu}} \sum_{i}^{m} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j, \tag{33}$$

$$s.t. \sum_{i}^{m} \alpha_i y_i = 0, \ 0 \leq \alpha_i \leq C, \mu_i \geq 0, \alpha_i = C - \mu_i, \ \forall i \tag{34}$$

- Utilizing $\alpha_i = C - \mu_i$, we obtain a simpler dual problem:

$$\max_{\boldsymbol{\alpha}} \sum_{i}^{m} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j, \tag{35}$$

$$s.t. \sum_{i}^{m} \alpha_i y_i = 0, \ 0 \leq \alpha_i \leq C, \ \forall i \tag{36}$$

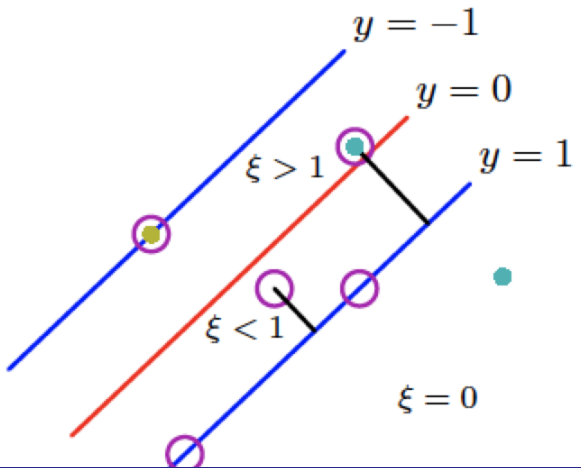
- Note that the only change in dual problem is the constraint $0 \leq \alpha_i \leq C$, which is $\alpha_i \geq 0$ in the dual problem of standard SVM.

Reference: `https://nianlonggu.com/2019/06/07/tutorial-on-SVM/`

SVM with slack variables

Solution interpretation

- The solution α_i has three cases: $\alpha_i = 0, 0 < \alpha_i < C, \alpha_i = C$
- $\alpha_i = 0$: the corresponding data are correctly classified and doesn't contribute to the classifier, locating outside of the margin
- $0 < \alpha_i < C$: in this case, $\mu_i > 0$ due to $\alpha_i = C - \mu_i$; Since $\mu_i \xi_i = 0$, then we have $\xi_i = 0$. The corresponding data are correctly classified and contributes to the classifier, locating on the margin
- $\alpha_i = C$: in this case, $\mu_i = 0$; then we have $\xi_i > 0$. The corresponding data contributes to the classifier, locating inside the margin
  - If $\xi_i \leq 1$, then the data is still correctly classified, not crossing decision boundary
  - If $\xi_i > 1$, then the data is incorrectly classified, crossing decision boundary



SVM with slack variables

How to determine the bias parameter b?

- We define $\mathcal{M} = \{i | 0 < \alpha_i < C\}$
- Since $0 < \alpha_i < C$, we have $\xi_i = 0$
- Then, for any support vector $\mathbf{x}_j, j \in \mathcal{M}$, we have

$$y_j(\mathbf{w}^\top \mathbf{x}_j + b) = 1, \ \forall j \in \mathcal{M} \tag{37}$$

$$\Rightarrow y_j(\sum_i^m \alpha_i y_i \mathbf{x}_i^\top \mathbf{x}_j + b) = 1, \ \forall j \in \mathcal{M} \tag{38}$$

- Utilizing $y_j \cdot y_j = 1$, we have

$$\sum_i^m \alpha_i y_i \mathbf{x}_i^\top \mathbf{x}_j + b = y_j, \ \forall j \in \mathcal{M} \tag{39}$$

$$\Rightarrow b = \frac{1}{|\mathcal{M}|} \sum_{j \in \mathcal{M}} (y_j - \sum_i^m \alpha_i y_i \mathbf{x}_i^\top \mathbf{x}_j) \tag{40}$$

- Note that using the average of all support vectors, rather than one single support vector, could make the solution of b more numerically stable.

Optimization of SVM

Why do we prefer to optimize the dual problem, rather than directly optimizing the primal problem? There are two main advantages:

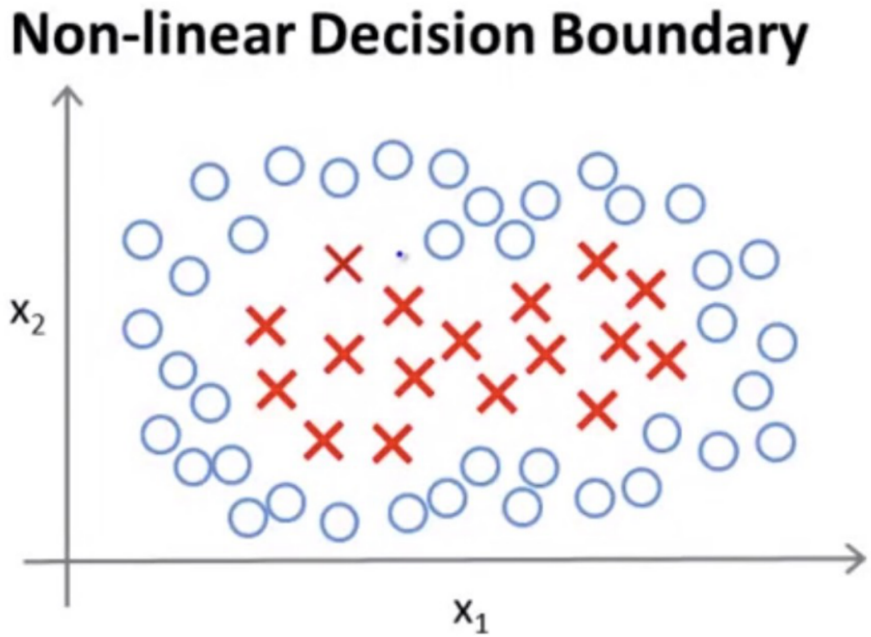
- By examining the dual form of the optimization problem, we gained significant insight into the structure of the problem
- The entire algorithm can be written in terms of only inner products between input feature vectors. In the following, we will exploit this property to apply the kernels to classification problem. The resulting algorithm, support vector machines, will be able to efficiently learn in very high dimensional spaces.

Reference:

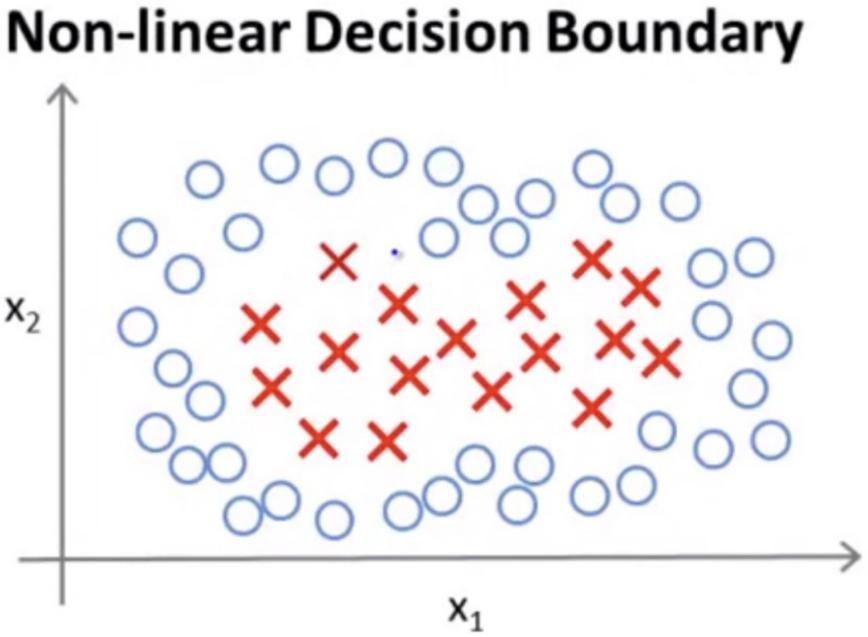
`https://stats.stackexchange.com/questions/19181/why-bother-with-the-dual-problem-wh`

Kernels

- In above derivation, SVM can only handle linearly separable data.
- For non-linearly separable data (*e.g.*, XOR data, and the following data), how to use SVM?
- Recall that the polynomial regression can handle non-linearly separable data



SVM with polynomial hypothesis function



Predict $y = 1$ if

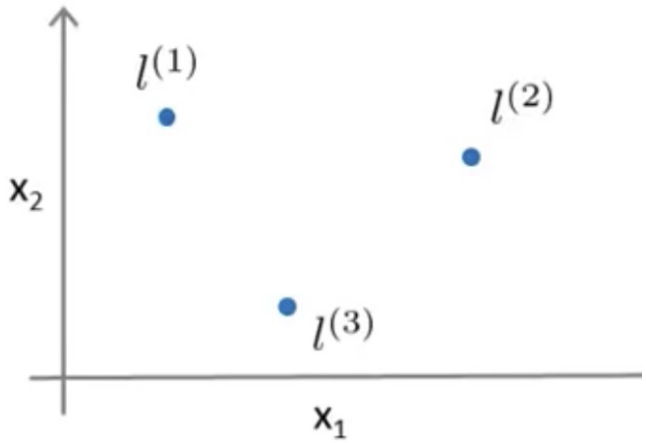
$$w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1 x_2 + w_4 x_1^2 + w_5 x_2^2 + \cdots \geq 0$$

- As introduced before, one can choose high-order polynomial hypothesis function to handle non-linear separable data,

$$f_{\mathbf{w},b}(\mathbf{x}) = \mathbf{w}^\top [1; x_1; x_2; x_1 x_2; x_1^2; x_2^2; \cdots] \tag{41}$$

- However, in many real problems, such as image classification, the dimensionality of original features $|\mathbf{x}|$ is already very high. Consequently, the dimensionality of high-order polynomial function will be too high, causing high computational cost or overfitting
- To tackle this difficulty, we will introduce kernel.

Kernels



- Given a new data $\mathbf{x}$, compute its new features based on proximity o landmarks $l^{(1)}, l^{(2)}, l^{(3)}$ (plot above), and here we use the Gaussian kernel, as follows

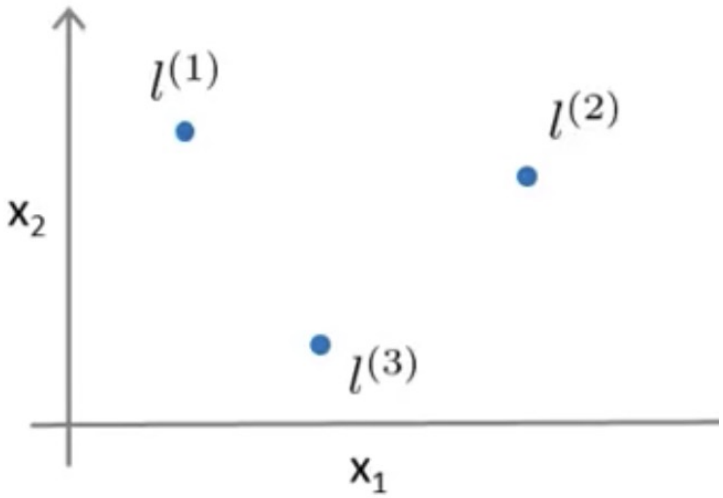
$$f_1 = \text{similarity}(\mathbf{x}, l^{(1)}) = \exp(-\frac{\|\mathbf{x} - l^{(1)}\|^2}{2\sigma^2}) \tag{42}$$

$$f_2 = \text{similarity}(\mathbf{x}, l^{(2)}) = \exp(-\frac{\|\mathbf{x} - l^{(2)}\|^2}{2\sigma^2}) \tag{43}$$

$$f_3 = \text{similarity}(\mathbf{x}, l^{(3)}) = \exp(-\frac{\|\mathbf{x} - l^{(3)}\|^2}{2\sigma^2}) \tag{44}$$

- Then, we have a new representation $[f_1; f_2; f_3]$ for the data $\mathbf{x}$

Kernels



- Kernel and similarity

$$f_1 = \text{similarity}(\mathbf{x}, l^{(1)}) = \exp(-\frac{\|\mathbf{x} - l^{(1)}\|^2}{2\sigma^2}) \tag{45}$$

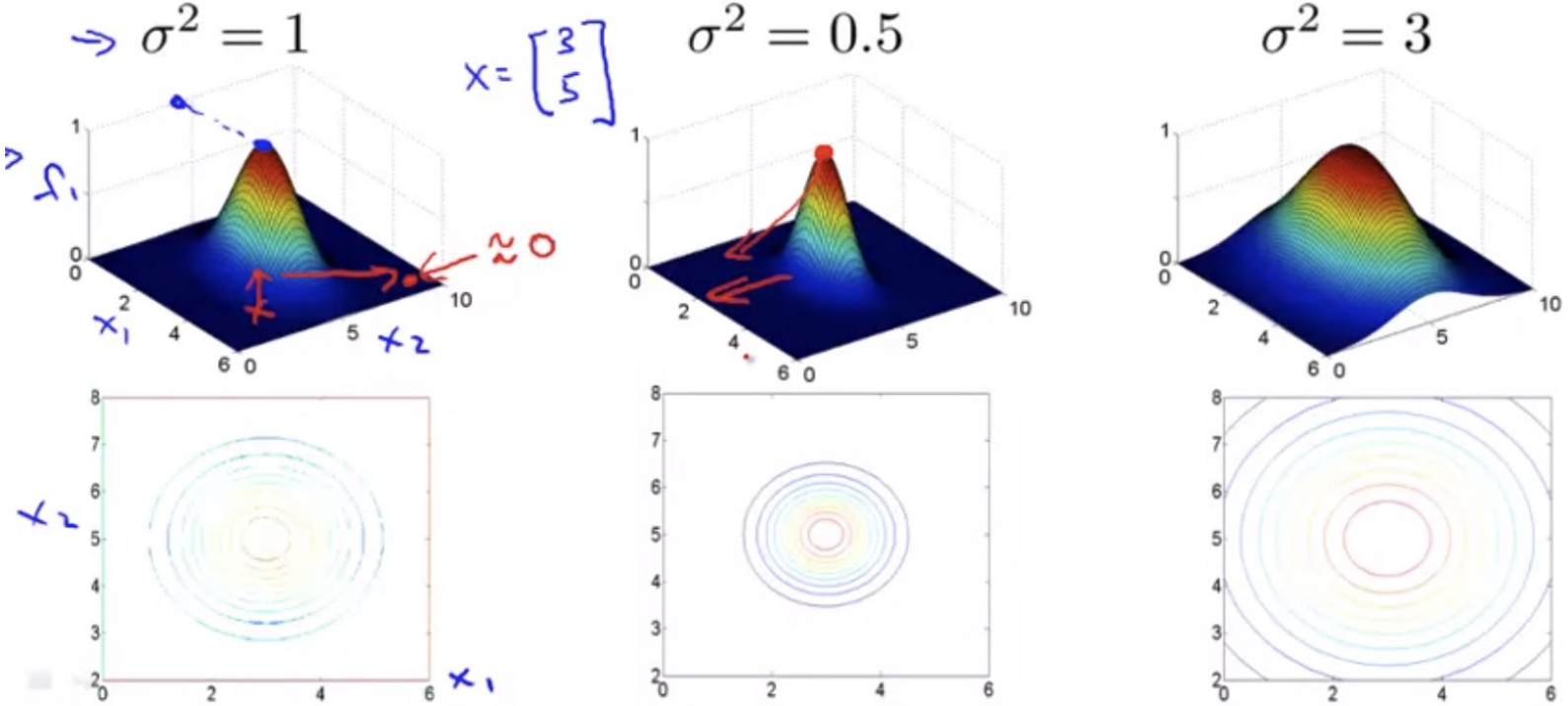
- If $\mathbf{x} \approx l^{(1)}$, then $f_1 \approx 1$
- If $\mathbf{x}$ is far from $l^{(1)}$, then $f_1 \approx 0$

Kernels

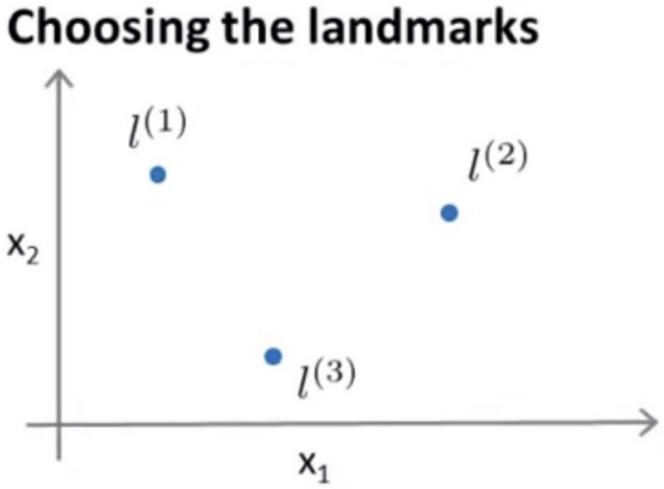
Example:

$$l^{(1)} = \begin{bmatrix} 3 \\ 5 \end{bmatrix}, \quad f_1 = \exp\left(-\frac{\|x - l^{(1)}\|^2}{2\sigma^2}\right)$$

$\sigma^2 = 1$ $\quad$ $\sigma^2 = 0.5$ $\quad$ $\sigma^2 = 3$



Kernels



Given $\mathbf{x}$:

$$f_i = \text{similarity}\left(\mathbf{x}, l^{(i)}\right)$$
$$= \exp\left(-\frac{\left\|\mathbf{x} - l^{(i)}\right\|^2}{2\sigma^2}\right)$$

How to obtain the landmark points?
We can set all training data points as landmarks.

SVM with Kernels

- We firstly define the following kernel

$$k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j) \tag{46}$$

- Utilizing this kernel to replacing $\mathbf{x}_i^\top \mathbf{x}_j$, we have the following dual problem

$$\max_{\boldsymbol{\alpha}} \sum_i^m \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j), \tag{47}$$

$$s.t. \sum_i^m \alpha_i y_i = 0, \ \alpha_i \geq 0, \ \forall i \tag{48}$$

- The solution of b becomes

$$b = \frac{1}{|\mathcal{S}|} \sum_{j \in \mathcal{S}} \left(y_j - \sum_i^m \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}_j) \right) \tag{49}$$

- The prediction of new data $\mathbf{x}$ becomes

$$\mathbf{w}^\top \mathbf{x} + b = \sum_i^m \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + b \tag{50}$$

- Since $\boldsymbol{\alpha}$ is sparse, the above classifier is also called sparse kernel classifier.

SVM with Kernels

Widely used kernels:

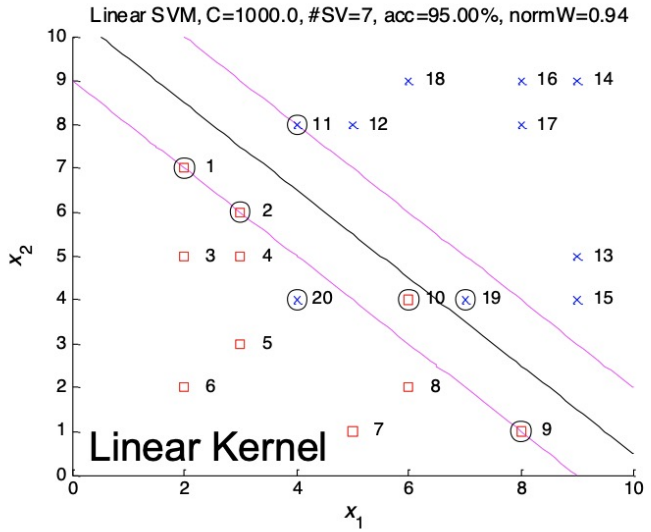
$$\text{Polynomial kernel: } k(\mathbf{x}, \mathbf{x}_i) = \left(1 + \frac{\mathbf{x}^\top \mathbf{x}_i}{\sigma^2}\right)^p, \ p > 0 \tag{51}$$

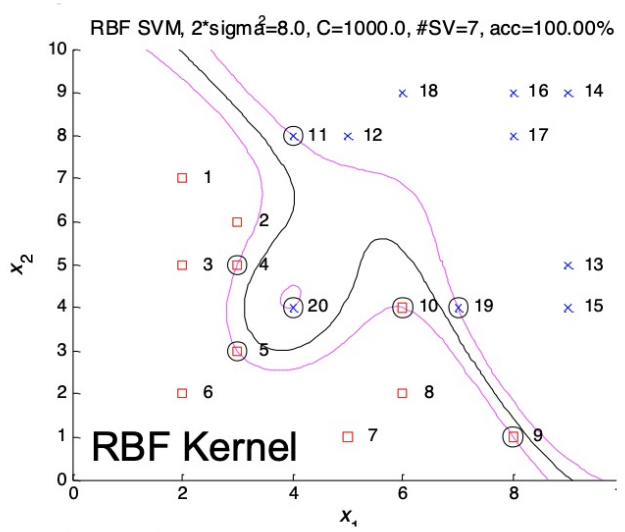
$$\text{Radial Basis Function (RBF) kernel: } k(\mathbf{x}, \mathbf{x}_i) = \exp\left\{-\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{2\sigma^2}\right\} \tag{52}$$

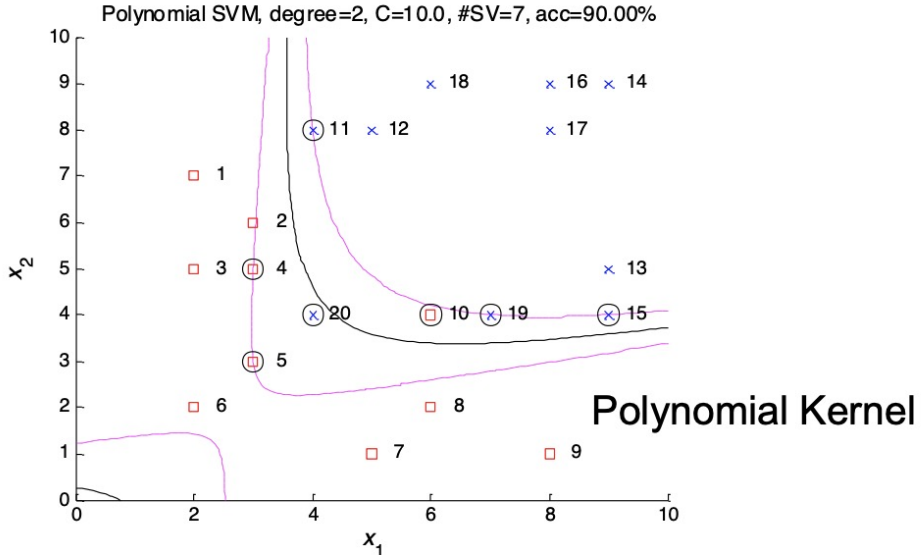
$$\text{Sigmoidal kernel: } k(\mathbf{x}, \mathbf{x}_i) = \frac{1}{1 + \exp^{-\frac{\mathbf{x}^\top \mathbf{x}_i + b}{\sigma^2}}} \tag{53}$$

SVM with Kernels

Comparing kernels:







SVM with Kernels

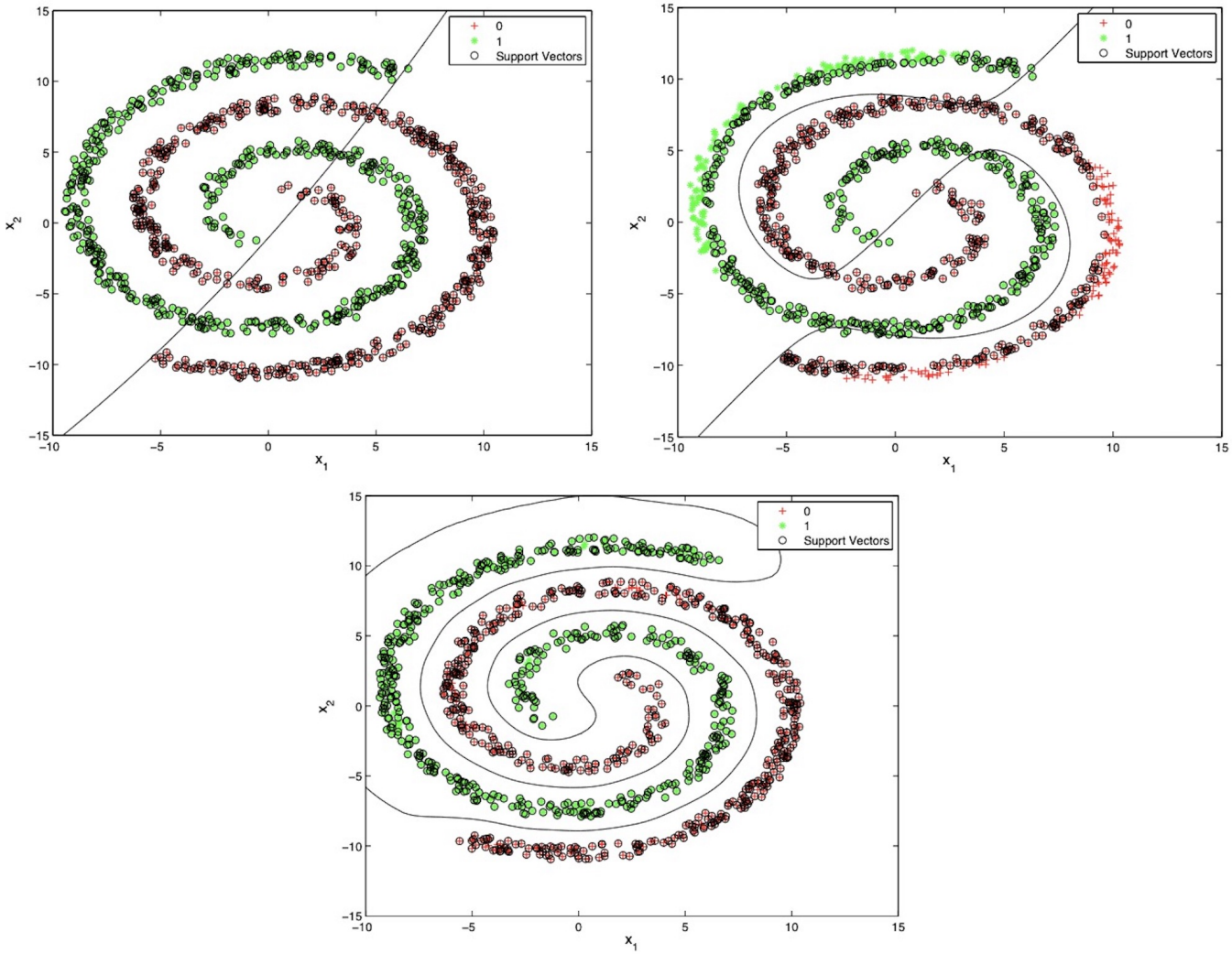
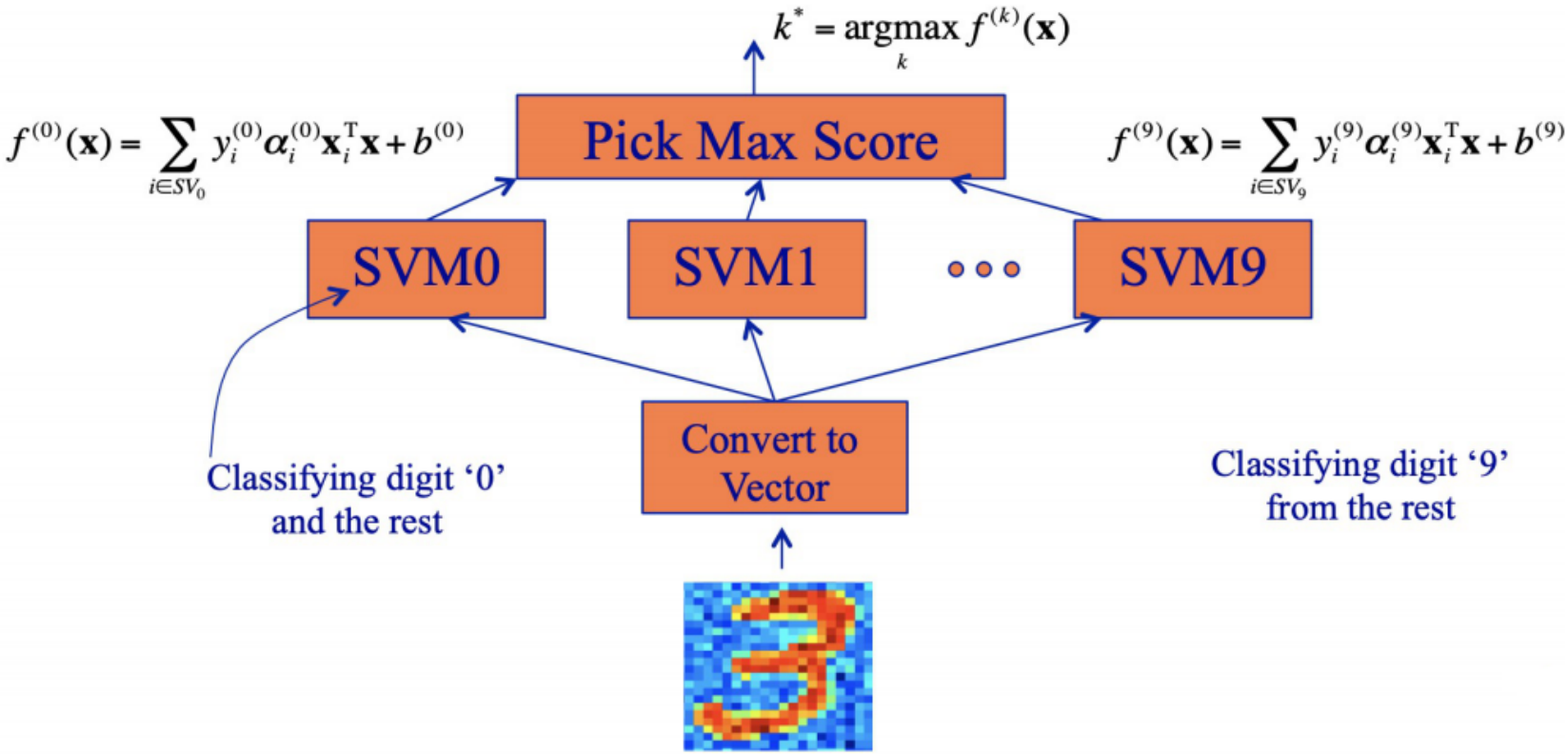


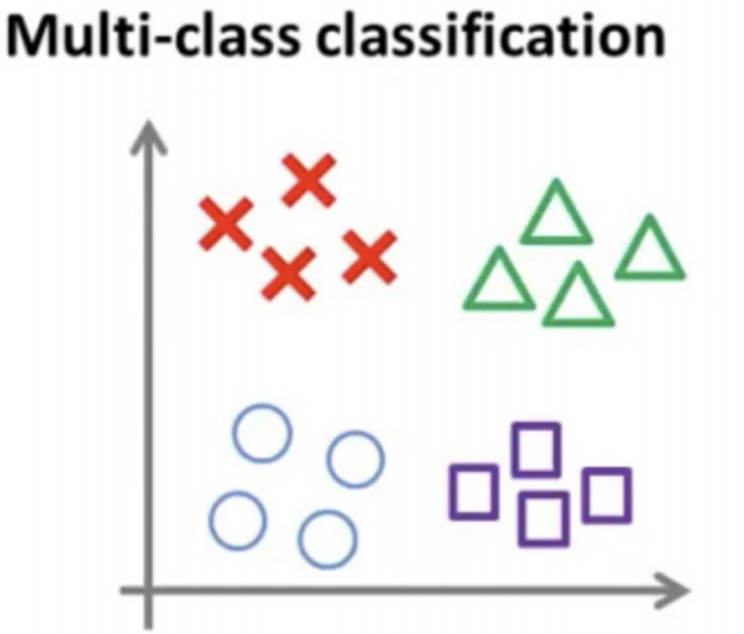
Figure: Decision boundaries produced by SVM with a 2nd-order polynomial kernel (top-left), a 3rd-order polynomial kernel (top-right), and a RBF kernel (bottom).

Multi-class SVM

- SVM is good for binary classification:
  $f(\mathbf{x}) > 0 \Rightarrow \mathbf{x} \in$ Class 1; $\quad f(\mathbf{x}) \leq 0 \Rightarrow \mathbf{x} \in$ Class 2
- To classify multiple classes, we use the one-vs-rest approach to convert K binary classifications to a K-class classification:



Multi-class SVM



$$y \in \{1, 2, 3, \ldots, K\}$$

- Many SVM packages already have built-in multi-class classification functionality.
- Otherwise, use one-vs.-all method. (Train K SVMs, one to distinguish $y = k$ from the rest, for $k = 1, 2, \ldots, K$), get $(\mathbf{w}^{(1)}, b^{(1)}), \ldots, (\mathbf{w}^{(K)}, b^{(K)})$.
- Predict the label of $\mathbf{x}$ as

$$\arg\max_{k \in \{1,2,\ldots,K\}} \left(\mathbf{w}^{(k)}\right)^\top \mathbf{x} + b^{(k)}$$

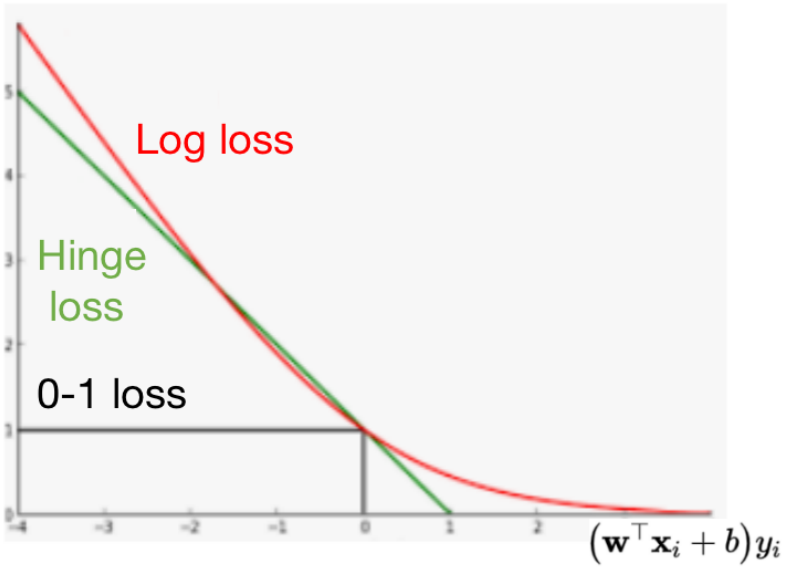
SVM vs. Logistic regression

SVM : Hinge loss

$$\text{loss}\left(f\left(\mathbf{x}_i\right), y_i\right) = \left(1 - \left(\mathbf{w}^\top \mathbf{x}_i + b\right) y_i\right)_+$$

Logistic Regression : Log loss (- log conditional likelihood)

$$\text{loss}\left(f\left(\mathbf{x}_i\right), y_i\right) = -\log P\left(y_i \mid \mathbf{x}_i, \mathbf{w}, b\right) = \log\left(1 + e^{-\left(\mathbf{w}^\top \mathbf{x}_i + b\right) y_i}\right)$$



SVM vs. Logistic regression

Logistic regression (LR) vs. SVM

- $n = |\mathbf{x}|$ indicates the number of features, and $m = |\mathcal{D}_{train}|$ is the number of training data
- If n is large (relative to m), then the data is linearly separable, one can use LR or SVM without kernel
- If n is small, and m is intermediate, then the data may be non-linearly separable, one use SVM with Gaussian kernel
- If n is small, and m is large, then create/add more features to make the data more separable, and one can use LR or SVM without kernel. Why not choose SVM with kernel in this case?

Software of SVM

- **Matlab**: `fitcsvm` trains an SVM for two-class classification.
- **Python**: `svm` from the `sklearn` package provides a set of supervised learning methods used for classification, regression and outliers detection.
- **C/C++**: `LibSVM` is a library for SVM. It also has Java, Perl, Python, Cuda, and Matlab interface.
- **Java**: `SVM-JAVA` implements sequential minimal optimization for training SVM in Java.
- **Javascript**: http://cs.stanford.edu/people/karpathy/svmjs/demo/

Using SVM in practice

- You are not required to implement SVM by yourself, and there are many well implemented sortwares, such as libsvm.
- When you choose a software to learn a SVM model, you need to specify:
  - Choice of parameter C (*i.e.*, the tradeoff hyper-parameter of the slack variables)
  - Choice of kernel
    - Linear kernel
    - Gaussian kernel, but you should set the kernel size (*i.e.*, variance of Gaussian distribution). Note that do perform feature scaling before using the Gaussian kernel.

Reading material

References:

- Andrew Ng's note on SVM:
  `https://see.stanford.edu/materials/aimlcs229/cs229-notes3.pdf`
- Chapter 7.1 of Bishop's book
- KKT conditions:
  `https://www.stat.cmu.edu/~ryantibs/convexopt-S15/scribes/12-kkt-pdf`

More variants of SVM:

- Semi-supervised SVM
- Structured SVM
- SVM with latent variables
- SVM for regression

Summary of SVM

What you need to know:

- Lagrange duality and KKT conditions
- Support vector machine:
  - Derivation of large margin
  - Derivation of hingle loss
  - Optimization using dual problem and KKT conditions
  - SVM with slack variables
  - SVM with kernels
  - Relationship between SVM and logistic regression