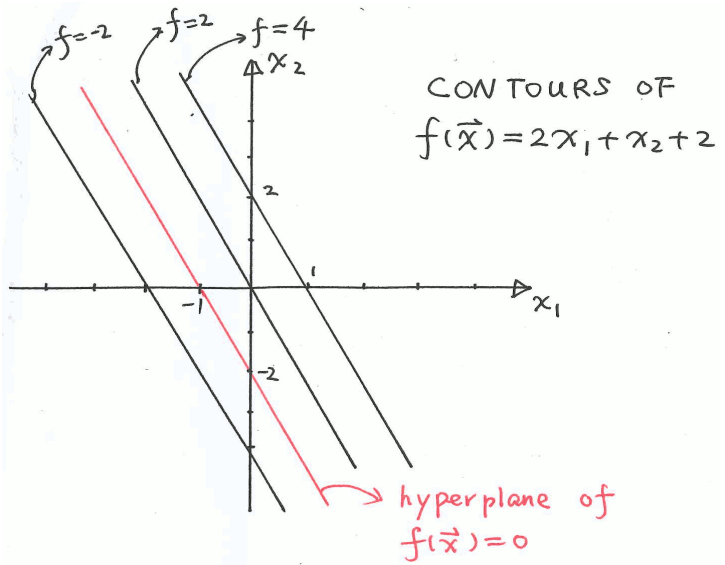
Support Vector Machines (SVM)

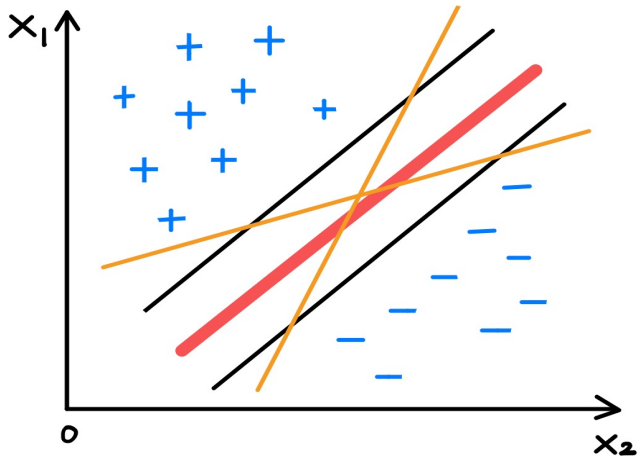
Contour lines and hyperplanes of linear functions

- Linear functions: $f(x) = w^\top x + b$. E.g. on $\mathbb{R}^2$, let $x = (x_1, x_2)^\top$, and $w = (w_1, w_2)^\top$, then $f(x) = w^\top x + b = w_1 x_1 + w_2 x_2 + b$.
- Contour of f: for each $c \in \mathbb{R}$, $\{x : f(x) = c\}$ is a contour of f at level c.
- The contours of linear functions are a class of lines (for $\mathbb{R}^2$), or a class of planes (for $\mathbb{R}^3$), or a class of hyperplanes (for general $\mathbb{R}^n$). We simply call them uniformly as hyperplanes.



• Any hyperplane of a linear function can be written as $w^\top x + b = 0$, denoted by (w, b).

Which one is the best?



- Goal: separate two classes by a hyperplane.
- Black and yellow lines: too close to training samples.
  - sensitive to noise.
- Red line: in the "middle" of training samples
  - noise has smallest influence.
- **Support vector machine:** SVM learns a decision boundary leading to the largest margin from both classes.

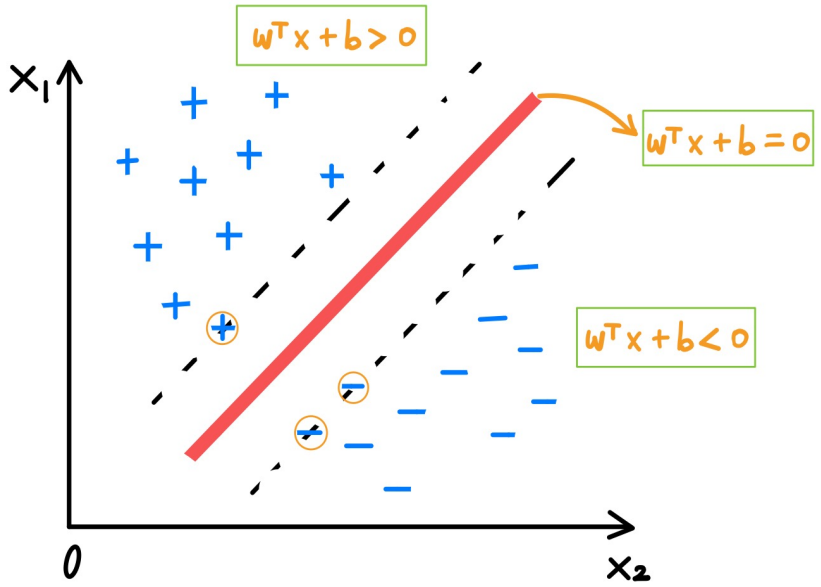
How to find the decision boundary (red line)? (1/3)

- Goal: decision boundary yielding largest distance between two classes.
- Given the training data $D = \{(x_1, y_1), \cdots (x_m, y_m)\}$, $y_i \in \{-1, +1\}$.
- Result 1: given a hyperplane (w, b), the distance to (w, b) from any x is

$$d = \frac{|w^\top x + b|}{\|w\|}. \qquad \text{(proof later)} \tag{1}$$

- Result 2: let (w, b) be any separation hyperplane. For all training samples

$$\begin{cases} w^\top x_i + b \geq +1, & \text{if } y_i = +1, \\ w^\top x_i + b \leq -1, & \text{if } y_i = -1. \end{cases} \qquad \text{(proof later)} \tag{2}$$

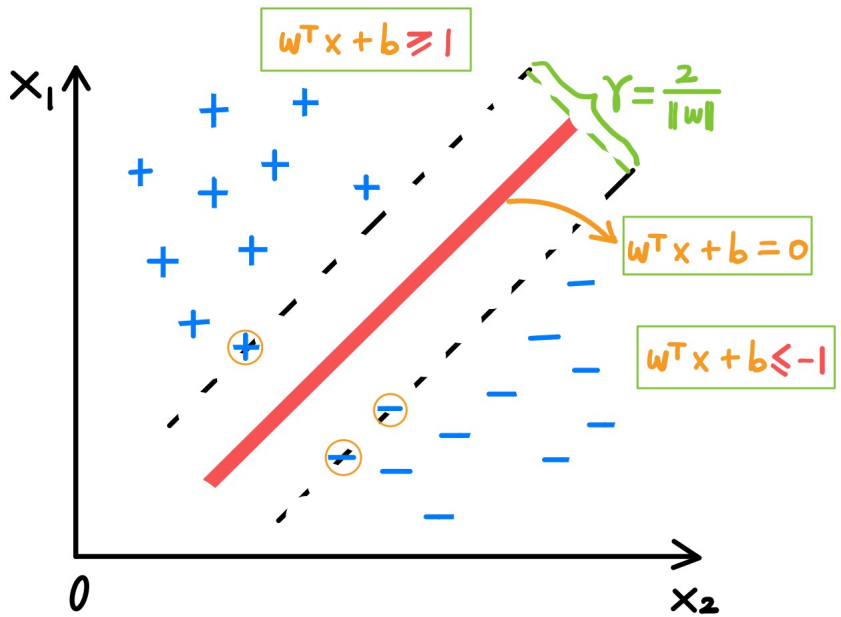


How to find the red line? (2/3)

- Training samples on the boundary satisfies $w^\top x + b = \pm 1$.
- Training samples on the boundary are called *support vectors*.
- The distance between two boundaries is

$$\gamma = \frac{2}{\|w\|}.$$

It is called *margin*.



How to find the red line? (3/3)

- To find a hyperplane with largest margin, we solve

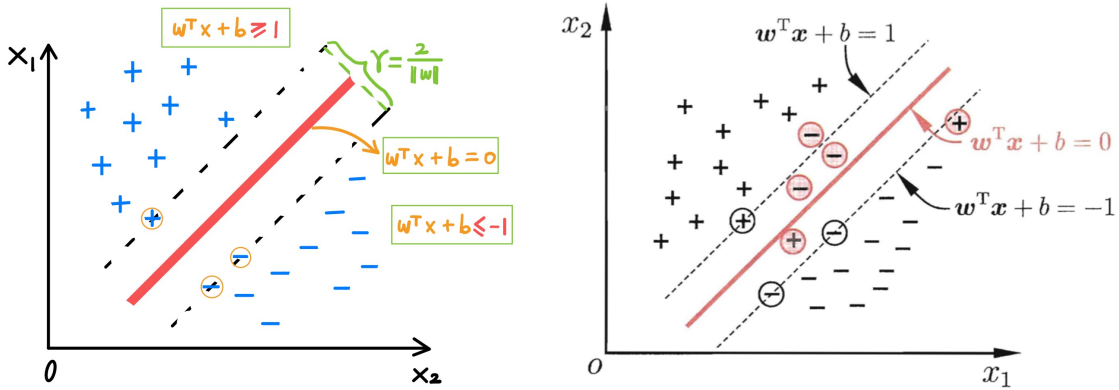
$$\begin{aligned} &\max_{w,b} \frac{2}{\|w\|} \\ &\text{s.t. } y_i(w^\top x_i + b) \geq 1, \quad i = 1, \cdots, m. \end{aligned}$$

- Equivalent to

$$\begin{aligned} &\min_{w,b} \frac{1}{2}\|w\|^2 \\ &\text{s.t. } y_i(w^\top x_i + b) \geq 1, \quad i = 1, \cdots, m. \end{aligned} \tag{3}$$

- This is the math model for support vector machine; convex optimization.
- What if NOT linearly separable ↷
  - SVM with soft margin: linear SVM;
  - Kernel SVM: nonlinear SVM.

SVM with soft margin (1/2)

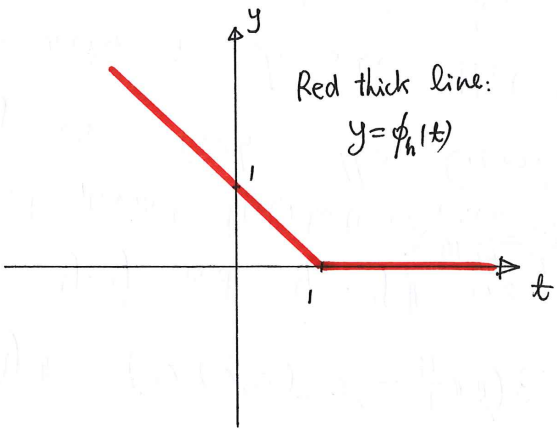


- (**Hard**) margin: all samples are classified correctly.
- **Soft** margin: allow some mistakes; $y_i(w^\top x_i + b)$ is violated for some i.
  - The number of violation should be as small as possible.
  - Consider the following general SVM

$$\min_{w \in \mathbb{R}^n, b \in \mathbb{R}} \left\{ \frac{1}{m} \sum_{i=1}^{m} \phi_{\mathsf{h}}(y_i(w^\top x_i + b)) + \gamma \|w\|^2 \right\}, \tag{4}$$

where $\phi_h(t) := \max\{0, 1 - t\}$ is the hinge loss function.

SVM with soft margin (2/2)



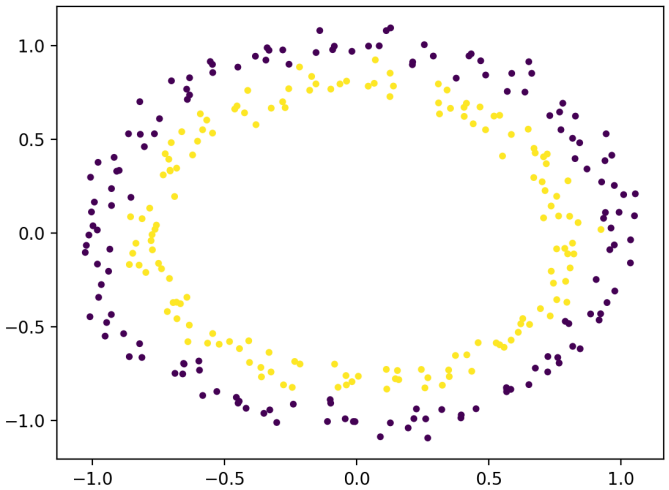
Recall

$$\min_{w\in\mathbb{R}^n, b\in\mathbb{R}} \left\{ \frac{1}{m}\sum_{i=1}^{m} \phi_{\mathsf{h}}(y_i(w^\top x_i + b)) + \gamma\|w\|^2 \right\}.$$

Remark.

- If $y_i(w^\top x_i + b) < 1$, $\phi_i(\cdots) \neq 0$. It leads to additional cost, to penalize wrong classification.
- Comparison of (3) and (4): in (3) $y_i(w^\top x_i + b) < 1$ does not satisfy constraint $\Rightarrow$ add infinite penalty $\Rightarrow$ not allow any mistake; in (4), some mistakes are allowed by adding cost.
- (4) is also a convex optimization.
- Regression view of (4): the first term is the regression term, and the second term is a ℓ^2 regularization.

Kernel SVM



- Kernel SVM

$$\min_{f\in\mathcal{H}_K, b\in\mathbb{R}} \left\{ \frac{1}{m}\sum_{i=1}^{m} \phi_{\mathsf{h}}(y_i(f(x_i)+b)) + \gamma\|f\|_K^2 \right\},$$

  where $\mathcal{H}_K := \left\{\sum_{j=1}^m c_j K_{x_j} : c_1,\ldots,c_m \in \mathbb{R}^n\right\}$, $\|f\|_K^2 := c^\top \mathbb{K} c$, and ϕ_h is hinge loss.
- $K_{x_j}(x) := K(x_j, x)$.
- $\min_{f\in\mathcal{H}_K, b\in\mathbb{R}} \cdots$ is equivalent to $\min_{c_1,\cdots,c_m,b} \cdots$.
- Kernel SVM can do nonlinear separation (classification).

Prediction of SVM

- For linear SVM,

$$(w^*, b^*) = \arg \min_{w \in \mathbb{R}^n, b \in \mathbb{R}} \left\{ \frac{1}{m} \sum_{i=1}^{m} \phi_{\mathsf{h}}(y_i(w^\top x_i + b)) + \gamma \|w\|^2 \right\}.$$

let

$$f(x) := (w^*)^\top x + b^*;$$

- For kernel SVM,

$$(f^*, b^*) = \arg \min_{f \in \mathcal{H}_K, b \in \mathbb{R}} \left\{ \frac{1}{m} \sum_{i=1}^{m} \phi_{\mathsf{h}}(y_i(f(x_i) + b)) + \gamma \|f\|_K^2 \right\},$$

let

$$f(x) := f^*(x) + b^*.$$

- Prediction. For the new data point x, the predicted label is $\operatorname{sign}(f(x))$.

  Intuition. $y_i(w^\top x_i + b) \geq 0 \Rightarrow$ the sign of y_i is the sign of $w^\top x_i + b$, and the sign of y_i is the predicted label.

Proof of (1)

The distance of the vector $x^{(1)}$ to the hyperplane (w, b) is

$$d = \frac{|w^\top x^{(1)} + b|}{\|w\|},$$

where $|\cdots|$ denotes absolute value, and recall (w, b) means $\{x : w^\top x + b = 0\}$.

Proof. Let x and y be any two points on (w, b). Then

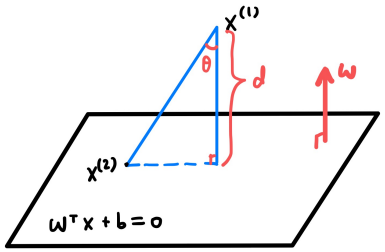
$$w^\top x + b = 0, \quad w^\top y + b = 0 \Rightarrow w^\top (x - y) = 0.$$

Thus, w is the normal vector of (w, b).

Let $x^{(2)}$ be any point on (w, b). Then

$$|\langle x^{(1)} - x^{(2)}, \frac{w}{\|w\|} \rangle| = \|x^{(1)} - x^{(2)}\| \cos\theta = d.$$

Note that $|\langle x^{(1)} - x^{(2)}, \frac{w}{\|w\|} \rangle| = |\frac{w^\top x^{(1)} - w^\top x^{(2)}}{\|w\|}| = |\frac{w^\top x^{(1)} + b}{\|w\|}|$. Done.



Proof of (2)

For finite training samples, if there exist $\hat{w}$ and $\hat{b}$ such that

$$\hat{w}^\top x_i + \hat{b} > 0, \quad i = 1, 2, \cdots, N$$

there must exist w and b such that

$$w^\top x_i + b \geq 1, \quad i = 1, 2, \cdots, N.$$

Proof. Let

$$a := \min_{i=1,\cdots,N} (\hat{w}^\top x_i + \hat{b}).$$

Then $a > 0$ and for each $i = 1, \cdots, N$

$$\hat{w}^\top x_i + \hat{b} \geq a > 0$$

which implies that

$$\frac{\hat{w}^\top}{a} x_i + \frac{\hat{b}}{a} \geq 1.$$

Let $w = \frac{\hat{w}^\top}{a}$ and $b = \frac{\hat{b}}{a}$.

Logistic Regression

Motivation (1/2)

- Logistic regression is indeed a classification algorithm.
- Only consider the binary classification with data $\{(x_i, y_i)\}_{i=1}^m$: $y_i \in \{-1, 1\}$.
- Recall the Bayes classifier

$$f(x) = \log \frac{\mathbb{P}(\text{Class} = 1|X = x)}{\mathbb{P}(\text{Class} = -1|X = x)},$$

where $x = (x_1, \cdots, x_n)^\top \in \mathbb{R}^n$ is the feature vector of one sample (instance).

- For naive Bayes classifier, we simplify the model by the conditional independence assumption (*)

$$\begin{aligned} & \mathbb{P}(\text{Class} = c|X = x) = \mathbb{P}(\text{Class} = c|X_1 = x_1, \cdots, X_n = x_n) \\ \propto\ & \mathbb{P}(X_1 = x_1, \cdots, X_n = x_n|\text{Class} = c)\mathbb{P}(\text{Class} = c) \\ \overset{(*)}{=}\ & \Pi_{j=1}^n \mathbb{P}(X_j = x_j|\text{Class} = c)\mathbb{P}(\text{Class} = c). \end{aligned}$$

Motivation (2/2)

- We simplify the Bayes classifier through another direction by assuming

$$\log \frac{\mathbb{P}(\text{Class} = 1|X = x)}{\mathbb{P}(\text{Class} = -1|X = x)} = \beta^\top x + \beta_0,$$

  where $\beta \in \mathbb{R}^n$ and $\beta_0 \in \mathbb{R}$.

- Consequently, $\mathbb{P}(\text{Class} = 1|X = x) = e^{\beta^\top x + \beta_0} \mathbb{P}(\text{Class} = -1|X = x)$. Using $\mathbb{P}(\text{Class} = 1|X = x) + \mathbb{P}(\text{Class} = -1|X = x) = 1$, we get

$$\begin{cases} \mathbb{P}(\text{Class} = -1|X = x) = \dfrac{1}{1 + e^{\beta^\top x + \beta_0}} \\ \mathbb{P}(\text{Class} = 1|X = x) = \dfrac{1}{1 + e^{-(\beta^\top x + \beta_0)}}. \end{cases}$$

- In either case of $y \in \{-1, +1\}$,

$$\mathbb{P}(\text{Class} = y|X = x) = \frac{1}{1 + e^{-y(\beta^\top x + \beta_0)}} := \sigma\left(y(\beta^\top x + \beta_0)\right),$$

  where $\sigma(t) = \frac{1}{1+e^{-t}}$ is the sigmoid function.

Logistic regression

- Estimate β and β_0 by MLE: given dataset $\{(x_i, y_i)\}_{i=1}^m$, log-likelihood is given by

$$\begin{aligned}\ell(\beta, \beta_0) &= \log \Pi_{i=1}^m \mathbb{P}(\text{Class} = y_i | X_i = x_i) \\ &= -\sum_{i=1}^m \log\left(1 + e^{-y_i(\beta^\top x_i + \beta_0)}\right). \end{aligned} \tag{5}$$

- Equation (5) is a 2nd differentiable and **concave** function of (β, β^0). Convex optimization theory can be used to find the estimator:

$$(\beta^*, \beta_0^*) = \arg\min_{\beta, \beta_0} \sum_{i=1}^m \log\left(1 + e^{-y_i(\beta^\top x_i + \beta_0)}\right)$$

- Gradient descent: $x^{n+1} = x^n - \nabla f(x^n)$.
  Newton's method[4]: $x^{n+1} = x^n - [\text{Hess} f(x^n)]^{-1} \nabla f(x^n)$, where $\text{Hess} f(x) = [\text{Hess} f(x)]_{i,j=1,\cdots,d} \in \mathbb{R}^{d\times d}$ with $[\text{Hess} f(x)]_{i,j} = \frac{\partial^2 f}{\partial x_i \partial x_j}$.
- Predicted label of x: sign $\left((\beta^*)^\top x + \beta_0^*\right)$.

[4] In logistic regression, Newton's method is called *iteratively reweighted least squares* (IRLS).

Prove (5) is concave (1/2)

Let $\tilde{\beta} = \begin{pmatrix} \beta_0 \\ \beta \end{pmatrix}$ and $\tilde{x}_i = \begin{pmatrix} 1 \\ x_i \end{pmatrix}$. Then

$$\ell(\beta, \beta_0) = -\sum_{i=1}^{m} \log\left(1 + e^{-y_i(\beta^\top x_i + \beta_0)}\right) := \ell(\tilde{\beta}) = -\sum_{i=1}^{m} \log\left(1 + e^{-y_i \tilde{\beta}^\top \tilde{x}_i}\right).$$

- Gradient of ℓ.

$$\frac{\partial \log(1 + e^{-y_i t})}{\partial t} = \frac{-e^{-y_i t} y_i}{1 + e^{-y_i t}}.$$

Let $t = \tilde{\beta}^\top \tilde{x}_i = \langle \tilde{\beta}, \tilde{x}_i \rangle$. Then $\frac{\partial t}{\partial \tilde{\beta}} = \tilde{x}_i$. By chain rule,

$$\begin{aligned} \nabla \ell(\tilde{\beta}) &= \sum_{i=1}^{m} \frac{e^{-y_i \tilde{\beta}^\top \tilde{x}_i} y_i \tilde{x}_i}{1 + e^{-y_i \tilde{\beta}^\top \tilde{x}_i}} \\ &= \left(\sum_{i=1}^{m} \frac{e^{-y_i \tilde{\beta}^\top \tilde{x}_i} y_i \tilde{x}_i^1}{1 + e^{-y_i \tilde{\beta}^\top \tilde{x}_i}}, \cdots, \sum_{i=1}^{m} \frac{e^{-y_i \tilde{\beta}^\top \tilde{x}_i} y_i \tilde{x}_i^j}{1 + e^{-y_i \tilde{\beta}^\top \tilde{x}_i}}, \cdots, \sum_{i=1}^{m} \frac{e^{-y_i \tilde{\beta}^\top \tilde{x}_i} y_i \tilde{x}_i^{n+1}}{1 + e^{-y_i \tilde{\beta}^\top \tilde{x}_i}} \right)'. \end{aligned}$$

- Hessian matrix of ℓ. The (k, j) element is $\frac{\partial^2 \ell(\tilde{\beta})}{\partial \tilde{\beta}_k \partial \tilde{\beta}_j} = \frac{\partial (\nabla \ell(\tilde{\beta}))_j}{\partial \tilde{\beta}_k}$:

Prove (5) is concave (2/2)

$$\frac{\partial}{\partial t}\frac{e^{-y_i t}}{1+e^{-y_i t}} = \frac{-y_i e^{-y_i t}}{(1+e^{-y_i t})^2}.$$

Let $t = \tilde{\beta}^\top \tilde{x}_i$. Then $\frac{\partial \tilde{\beta}^\top \tilde{x}_i}{\tilde{\beta}_i^k} = \tilde{x}_i^k$.

By Chain rule,

$$\begin{aligned}[\text{Hessian}(\ell)]_{kj} &= \frac{\partial}{\partial \tilde{\beta}_k}\sum_{i=1}^m \frac{e^{-y_i\tilde{\beta}^\top\tilde{x}_i} y_i \tilde{x}_i^j}{1+e^{-y_i\tilde{\beta}^\top\tilde{x}_i}} \\ &= \sum_{i=1}^m \frac{-e^{-y_i\tilde{\beta}^\top\tilde{x}_i} y_i^2 \tilde{x}_i^k \tilde{x}_i^j}{(1+e^{-y_i\tilde{\beta}^\top\tilde{x}_i})^2}.\end{aligned}$$

Thus,

$$\text{Hessian}(\ell) = -\sum_{i=1}^m \frac{e^{-y_i\tilde{\beta}^\top\tilde{x}_i} y_i^2}{(1+e^{-y_i\tilde{\beta}^\top\tilde{x}_i})^2}\tilde{x}_i\tilde{x}_i^\top.$$

Moreover, for any $\xi \in \mathbb{R}^{n+1}$,

$$\xi^\top \text{Hessian}(\ell)\xi = -\sum_{i=1}^m \frac{e^{-y_i\tilde{\beta}^\top\tilde{x}_i} y_i^2}{(1+e^{-y_i\tilde{\beta}^\top\tilde{x}_i})^2}\xi^\top\tilde{x}_i\tilde{x}_i^\top\xi < 0,$$

if no collinearity problem.

Logistic regression with regularization

- The logistic regression algorithm with regularization parameter $\lambda \geq 0$ is

$$(\beta^*, \beta_0^*) = \arg\min_{\beta, \beta_0} \sum_{i=1}^{m} \log\left[1 + e^{-y_i(\beta^\top x_i + \beta_0)}\right] + \lambda\|\beta\|^2.$$

- The predicted function is the sign of the score function $(\beta^*)^\top x + \beta_0^*$

$$\text{sign}\left((\beta^*)^\top x + \beta_0^*\right).$$

Why regularization?

- If we set $\lambda = 0$, we may suffer from overfitting.
- Let $\mathcal{E}(\beta, \beta_0) = \sum_{i=1}^{n} \log \left[1 + e^{-y_i(\beta^\top x_i + \beta_0)}\right]$. When data are linearly separable, $\mathcal{E}(\beta, \beta_0)$ may not have a minimizer. In fact, if there is a hyperplane (β, β_0) such that $y_i(\beta^\top x_i + \beta_0) > 0$ for any i, then $\mathcal{E}(\cdot)$ has no minimizer.

  Proof: for any $p > 1$, we have $\mathcal{E}(p\beta, p\beta_0) < \mathcal{E}(\beta, \beta_0)$. There is no minimizer.

Evaluating the performance of classifiers

Score-based classifier

• Recall that the predicted label of e.g SVM and logistic regression is based on a function (linear or nonlinear) $f(x)$ which has real values[5]. The classifier we defined is sign$(f(x))$. Equivalently speaking,

$$f(x) \geq 0 \Rightarrow x \in \text{Class 1, or "Positive"},$$
$$f(x) < 0 \Rightarrow x \in \text{Class 2, or "Negative"}.$$

• Sometimes, we hope to give one class more preference. For example, when we design a fire alarm, it won't hurt too much if the alarm bell rings when there is indeed no fire; but not vise versa!!

• We set a threshold λ:

$$f(x) \geq \lambda \Rightarrow x \in \text{Class 1, or "Positive"},$$
$$f(x) < \lambda \Rightarrow x \in \text{Class 2, or "Negative"}.$$

• If $\lambda > 0$, it is more difficult to belong to class 1 and easier to belong to class 2.

[5]This function is called score function.

- For a specified λ and a sample x, let $f_{\mathcal{C}}(x)$ be the predicted class of x.
- $f_{\mathcal{C}}(x) =$ Class P when $f(x) \geq \lambda$, and $f_{\mathcal{C}}(x) =$ Class N when $f(x) < \lambda$.
- Based on above criterion[6], we have the following table (confusion matrix).

	Positive	Negative	Total
Positive	True Positive (TP)	False Positive (FP)	Predicted Positive (PP)
Negative	False Negative (FN)	True Negative (TN)	Predicted Negative (PN)
Total	Actual Positive (AP)	Actual Negative (AN)	Total Population

- True positive rate:

$$TPR := \frac{TP}{AP}$$

False positive rate:

$$FPR := \frac{FP}{AN}$$

[6] given one λ and given one classification model

	Positive	Negative	Total
Positive	True Positive (TP)	False Positive (FP)	Predicted Positive (PP)
Negative	False Negative (FN)	True Negative (TN)	Predicted Negative (PN)
Total	Actual Positive (AP)	Actual Negative (AN)	Total Population

Example: compute TPR and FPR for the following table.

Truth	Predicted
P	P
N	P
P	P
N	N
N	N
P	N
P	P
P	P
N	N
N	N
N	N
N	P
P	P

Exercise

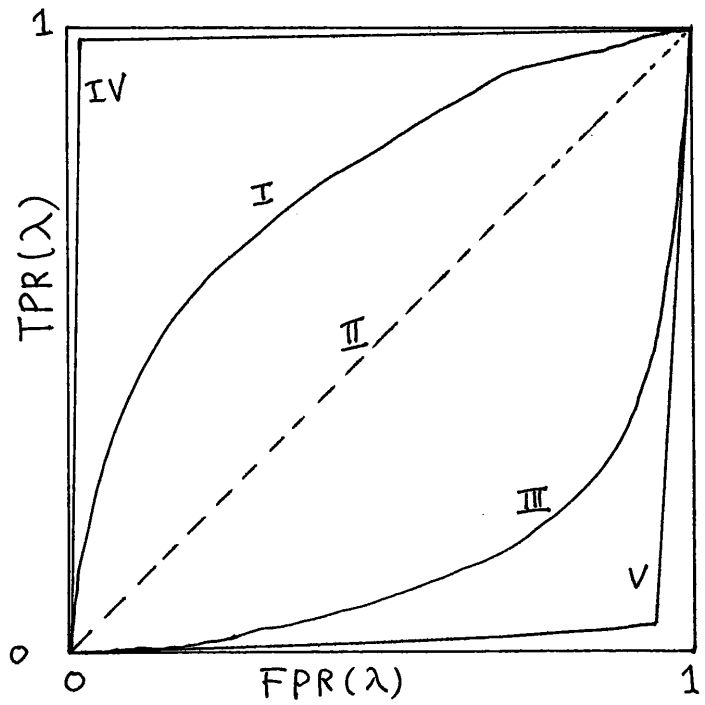
For the following classification prediction, compute the TPR and FPR.

Predicted label	True label
P	P
P	N
N	P
P	P
N	N
N	N
N	N
N	N
N	P
P	P
N	P

For a specified λ, let $f_{\mathcal{C}}(x) = \text{P}$ if $f(x) \geq \lambda$, and $f_{\mathcal{C}}(x) = \text{N}$ if $f(x) < \lambda$.

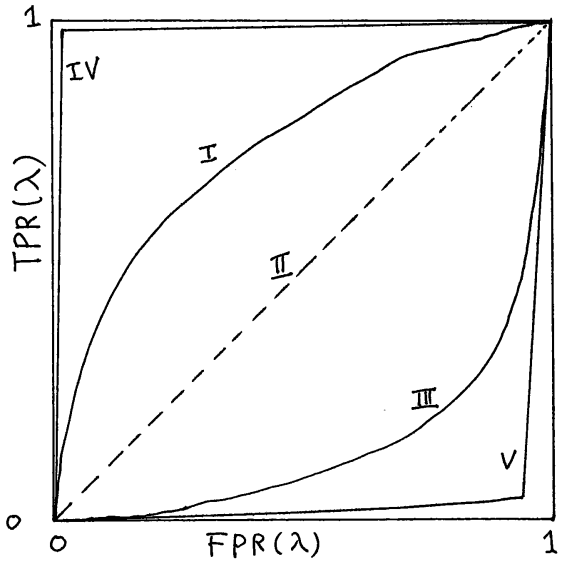
- $\lambda = -\infty$, the classifier predicts everything as positive:
  $TN = FN = 0 \Rightarrow TPR := TPR(\lambda) = 1$, and $FPR := FPR(\lambda) = 1$.
  Of course, this is just a useless predictor.
- $\lambda = \infty$, the classifier predicts everything as negative:
  $TP = FP = 0 \Rightarrow TPR = TPR(\lambda) = 0$, and $FPR = FPR(\lambda) = 0$.
  Of course, this is another useless predictor.
- As λ ranges from $-\infty$ to $+\infty$, the functinons $\lambda \mapsto FPR(\lambda)$ and $\lambda \mapsto TPR(\lambda)$ both decrease:
  When λ is smaller, there are more samples to be predicted as positive, and thus TP increases, but AP does not change. Hence, $TPR(\lambda)$ is larger.
- **Performance evaluation**:
  - the larger TP and TN, the better; the smaller FP and FN, the better.
  - the larger TPR, the better; the smaller FPR, the better.
  - λ cannot be too large or too small $\Rightarrow$ ROC curve $\curvearrowright$

The Receiver Operating Characteristics (ROC) Curve (1/2)



- ROC curve is $(FPR(\lambda), TPR(\lambda))$ for different threshold λ.
- Five ROC curves: I, II, III, IV, V. Each curve corresponds to one classifier.
- Usage of ROC:
  - Determine a "suitable" threshold for one classifier by considering its associated ROC curve.
  - Compare the performance of different classifiers by comparing their associated ROC curves.

ROC curve (2/2)

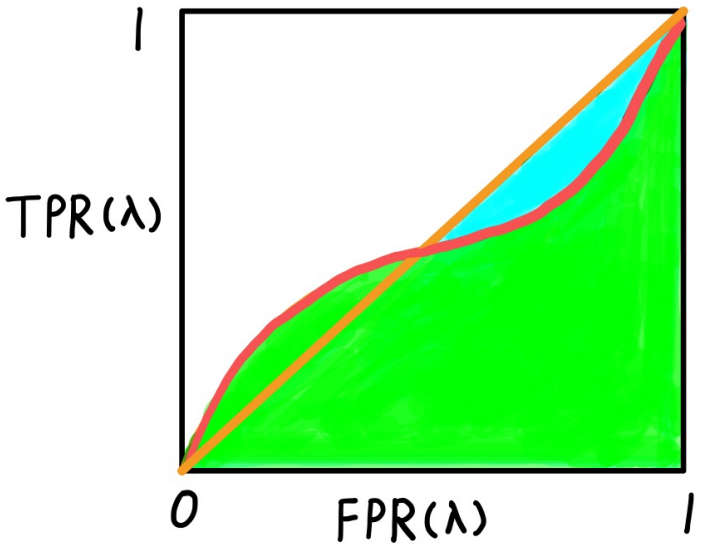


Recall the performance evaluation criterion:

- the larger TPR, the better;
- the smaller FPR, the better.

- Suitable threshold for the classifier IV: corner point.
- Comparison of performance of different classifiers:
  - $IV > I > II > III > V$.
  - What if one curve is not fully covered by the other? AUC ↷

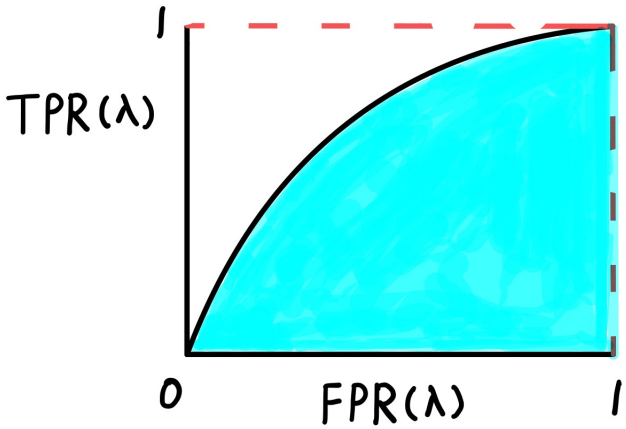
The Area Under ROC Curve (AUC)



- AUC[7] is defined as the area under the ROC curve. So AUC $\in [0, 1]$.
- The larger AUC score, the better the classifier performs.

[7] also called AUC score.

Compute AUC



- Recall f is the score function.
- Let X and Y be two r.v.s. and $X \perp Y$.
- $\text{FPR}(\lambda) = \mathbb{P}\left(f(X) > \lambda | X \in AN\right)$ and $\text{TPR}(\lambda) = \mathbb{P}(f(Y) > \lambda | Y \in AP)$

$$
\begin{aligned}
AUC &= \int_0^1 \text{TPR}\, d\text{FPR} = -\int_\infty^{-\infty} \mathbb{P}(f(Y) > \lambda | Y \in AP)\mathbb{P}(f(X) \in \lambda | X \in AN) d\lambda \\
&= \int_{-\infty}^{\infty} \mathbb{P}(f(Y) > \lambda | Y \in AP)\mathbb{P}(f(X) \in \lambda | X \in AN) d\lambda \\
&= \frac{\int_{-\infty}^{\infty} \mathbb{P}(f(Y) > \lambda, Y \in AP)\mathbb{P}(f(X) \in \lambda, x \in AN)\, d\lambda}{\mathbb{P}(Y \in AP)\mathbb{P}(X \in AN)} \\
&= \frac{\mathbb{P}(Y \in AP, X \in AN, f(Y) > f(X))}{\mathbb{P}(Y \in AP)\mathbb{P}(X \in AN)}
\end{aligned}
$$

In the case of discrete sample space,

$$
AUC = \frac{\frac{1}{m^2}\sum_{x \in AN}\sum_{y \in AP} \mathbf{1}_{\{f(y) > f(x)\}}}{\frac{\#AP}{m} \times \frac{\#AN}{m}} = \frac{\sum_{x \in AN}\sum_{y \in AP} \mathbf{1}_{\{f(y) > f(x)\}}}{\#AP \times \#AN}.
$$

For the following classification output, compute the AUC.

True Label	Predicted Score
P	1.7
P	8.1
P	7.8
P	9.3
N	6.2
N	6.1
N	1.2
N	2.9
N	5.8
N	6.3

$$AUC = \frac{\sum_{x \in AN} \sum_{y \in AP} \mathbf{1}_{\{f(y) > f(x)\}}}{\#AP \times \#AN}.$$

For the following classification output, compute the AUC.

True Label	Predicted Score
P	2.0
N	6.9
N	9.2
N	2.8
P	1.0
P	7.0
N	5.3
P	8.1
P	9.6
P	1.1