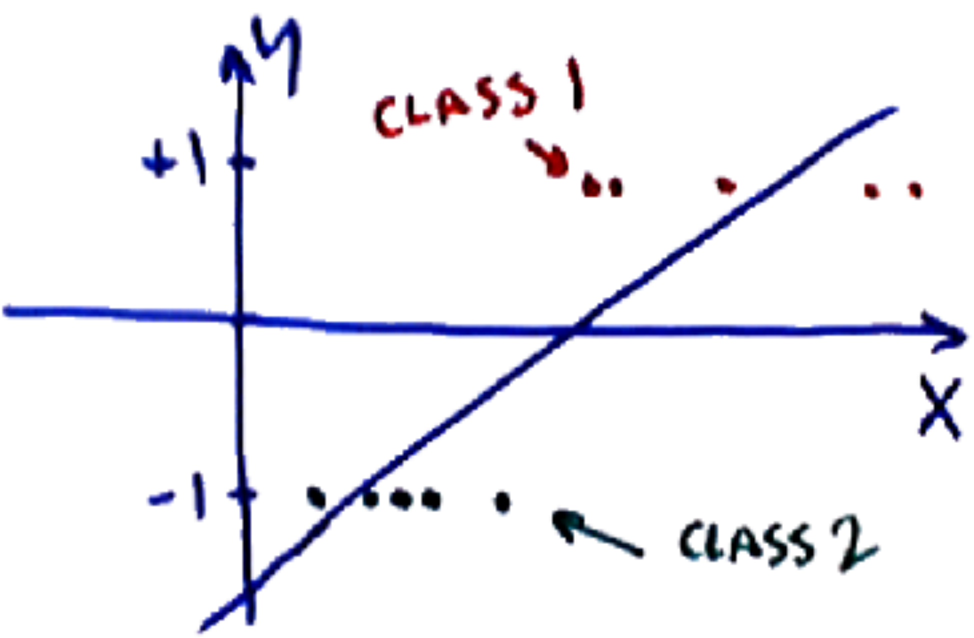
Classification

Classification methods

- Geometrical:
  - Nearest Neighbor
  - Logistic regression
  - Support Vector Machine
- Symbolisme:
  - Decision Tree
- Connectionism:
  - Perceptron
  - Neural networks
- Bayesian:
  - Naive Bayes

Linear regression



We can use the Mean Squared Error criterion with a linear regressor to perform classification (although this is clearly suboptimal).

We compute a **linear discriminant function** $G(W, X) = W'X$ and compare it to a threshold T. If $G(W, X)$ is larger than T, we classify X in class 1, if it is smaller than T, we classify X n class 2.

- To compute W, we simply minimize the quadratic loss function

$$\mathcal{L}(W) = \frac{1}{P}\sum_{i=1}^{P}\frac{1}{2}(y^i - W'X^i)^2$$

where $y^i = +1$ if training sample X^i is of class 1 and $y^i = -1$ if training sample X^i is of class 2.

- This is called the Adaline algorithm (Widrow-Hoff 1960).

Logistic regression

- Key idea: predict the probability of the classes. Binary case: y = +1 or -1.

- decision rule: $y = F(W'X)$, with $F(a) =$ 1/(1 + exp(-a)) (sigmoid function).
- loss function: $L(W, y^i, X^i) = 2\log(1 + \exp(-y^i W' X^i))$
- gradient of loss: $\frac{\partial L(W, y^i, X^i)}{\partial W}' = -\left(Y^i - F(W'X)\right) X^i$
- update rule: $W(t+1) = W(t) + \eta(t)(y^i - F(W(t)'X^i))X^i$

Linear vs. Logistic regression

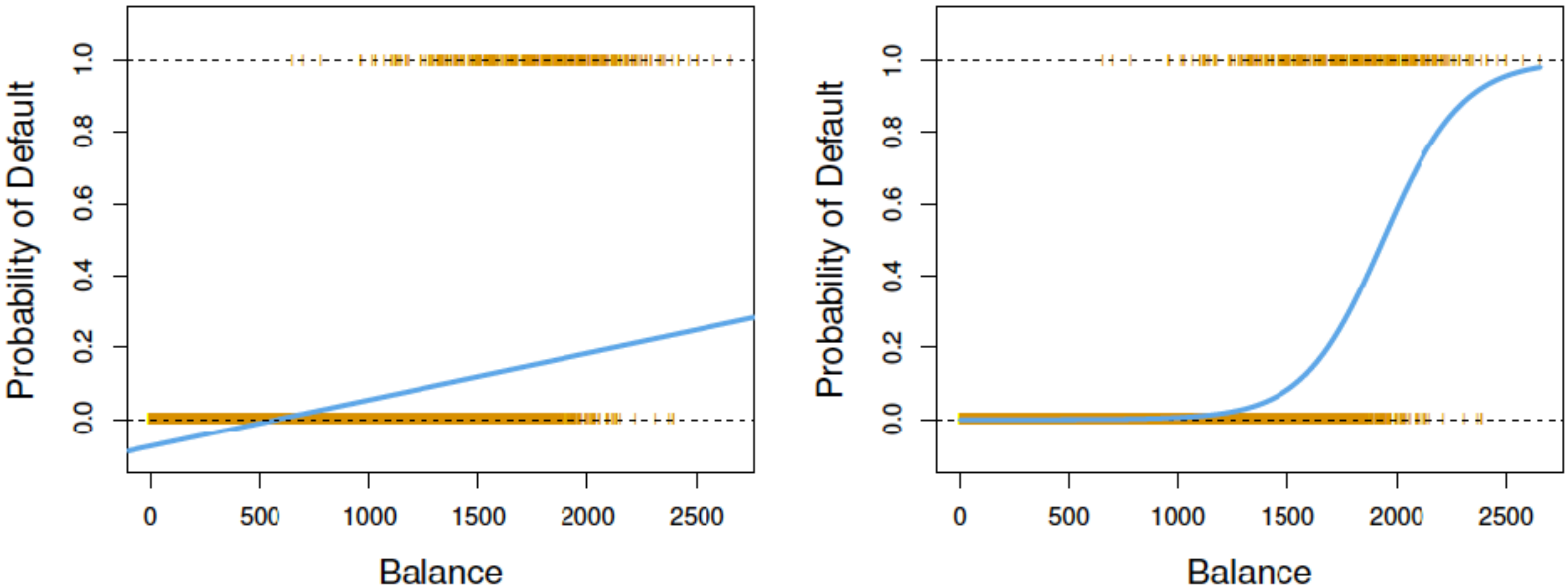
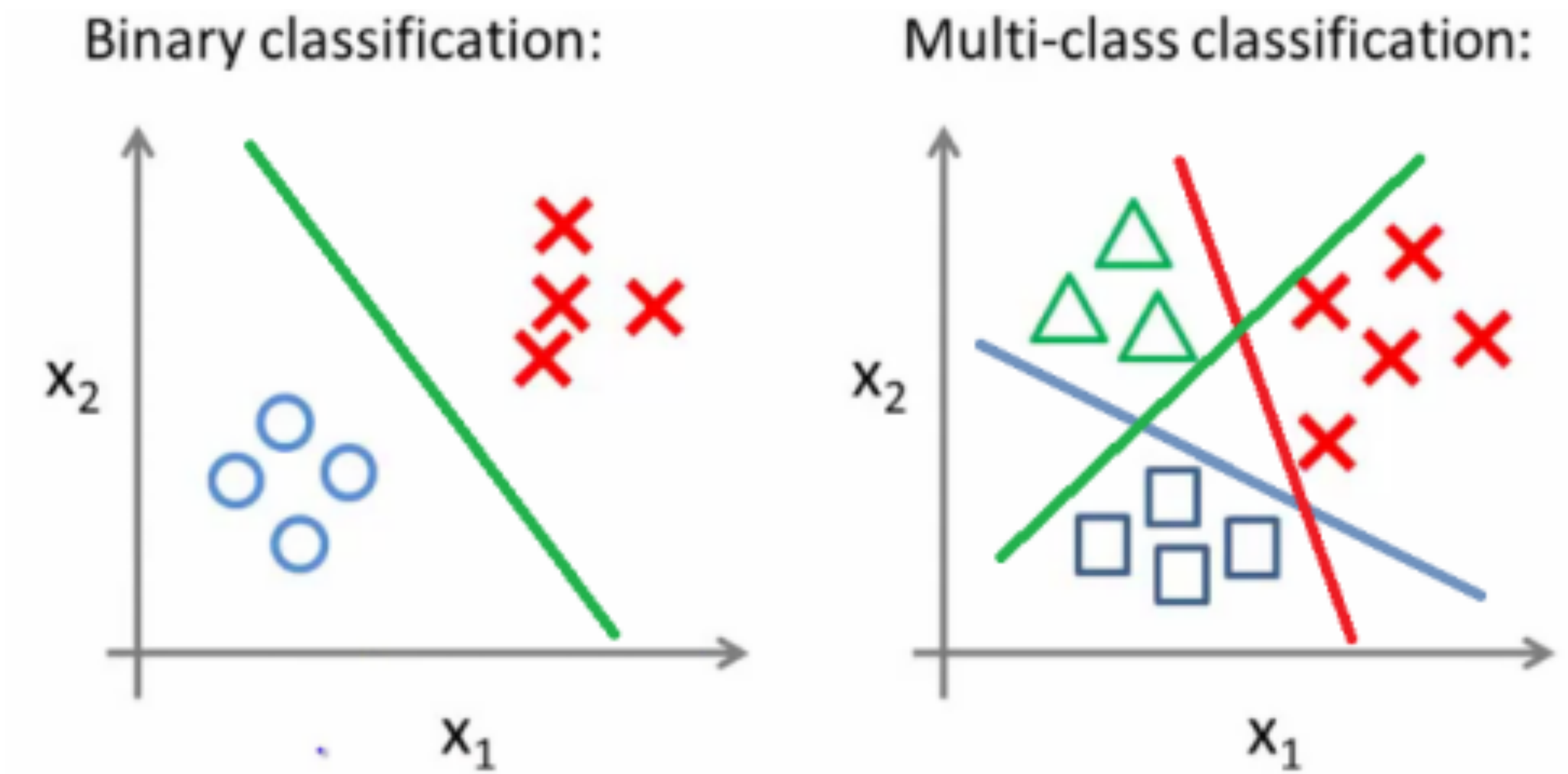


Figure: Left: linear regresion; Right: logistic regression

Multi-class classification



https://utkuufuk.github.io/2018/06/03/one-vs-all-classification/

Multi-class classification

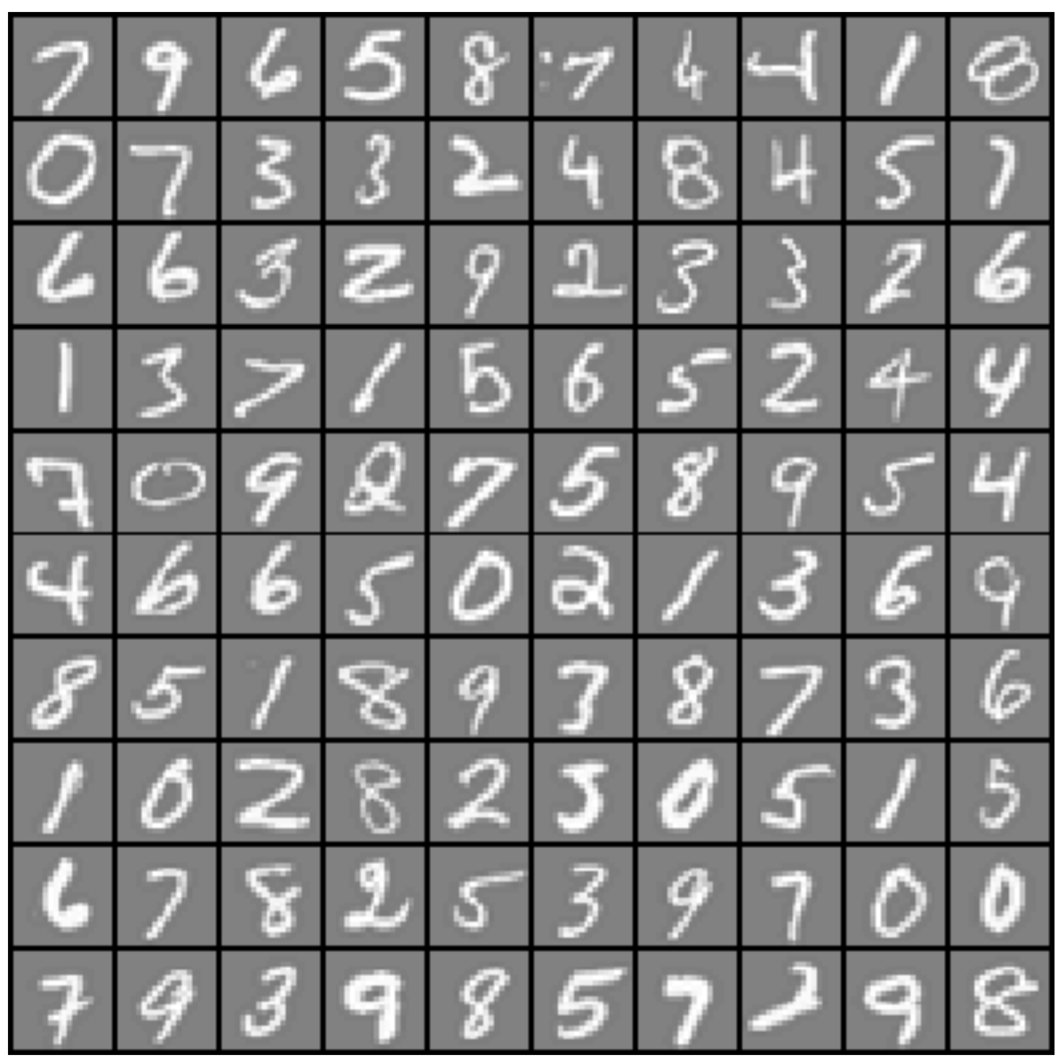
- 多分类学习的一条思路是“拆解法“：
  - 将多个任务拆分为若干个二分类任务求解。
- 1对1: 将C个类别两两配对，从而产生C(C-1)/2个二分类任务。测试时，通过C(C-1)/2个分类结果投票。
- 1对其余：将每一个类的样例作为正例，所有其他的类作为反例来训练C个分类器。测试时，通常选择置信度最高的分类器的结果。
- 多对多(Many vs. Many)：需要特殊设计若干类为正例，若干类为反例。

Multi-class Logistic regression

- How to fit the model to the data?

Let $y_{ic} = 1$ if sample x_i belongs to class c

Let $y_{ic} = 0$ if sample x_i does not belong to class c



Multi-class Logistic regression

We fit the best w_c to maximize the log-likelihood.

The likelihood is: $\prod_{i=1}^{N} \prod_{c=1}^{C} p(y_i = c|x_i)^{y_{ic}}$.

The log-likelihood is: $\sum_{i=1}^{N} \sum_{c=1}^{C} y_{ic} \log p(y_i = c|x_i)$.

How to define $\log p(y|x)$?

Multi-class Logistic regression

- Softmax (vs. Max)

If z_j is the unique maximum among z_c for $1 \leq c \leq C$

$$\frac{e^{\beta z_j}}{\sum_{c=1}^{C} e^{\beta z_c}} \to 1 \qquad \text{as } \beta \to +\infty$$

Multi-class Logistic regression

- Softmax

  - 逻辑回归只能处理二分类问题，现实应用中很多问题为多分类：
    - 将商品评论分为好评、中评和差评
    - 手写数字识别中，将手写数字分类为 0、1、…、9
  - 对于C分类问题(C>2)，Softmax函数代替Logistic函数：

$$p(y_i|\boldsymbol{x}_i) = \frac{e^{\boldsymbol{w}_{y_i}^{\mathrm{T}}\boldsymbol{x}_i}}{\sum_{c=1}^{C} e^{\boldsymbol{w}_c^{\mathrm{T}}\boldsymbol{x}_i}}$$

  - 其中w_c为第c类的参数向量

Softmax函数定义

$$\sigma(\boldsymbol{z})_j = \frac{e^{z_j}}{\sum_{c=1}^{C} e^{z_c}}$$

Class-imbalance

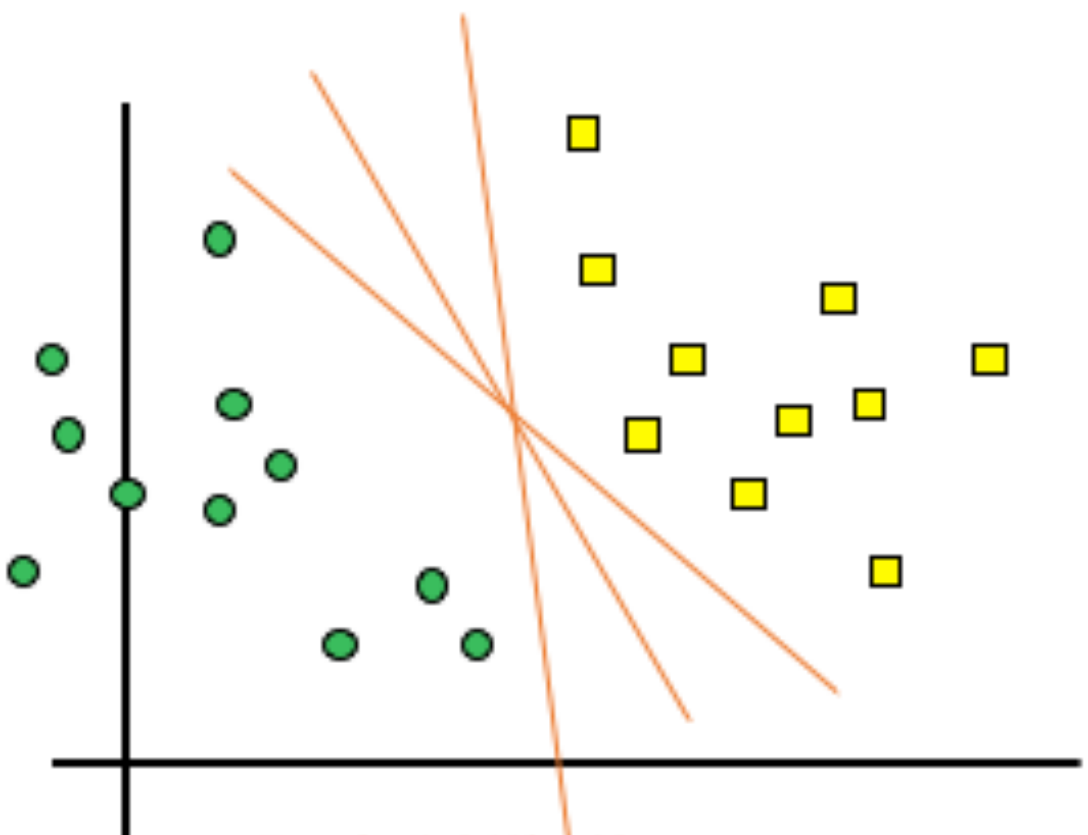
- 类别不平衡：分类任务中不同类别的训练样例数目差别很大。
- 一般方法：under-sampling, over-sampling, threshold-moving
- Read: p.66 机器学习，周志华.

Classification methods

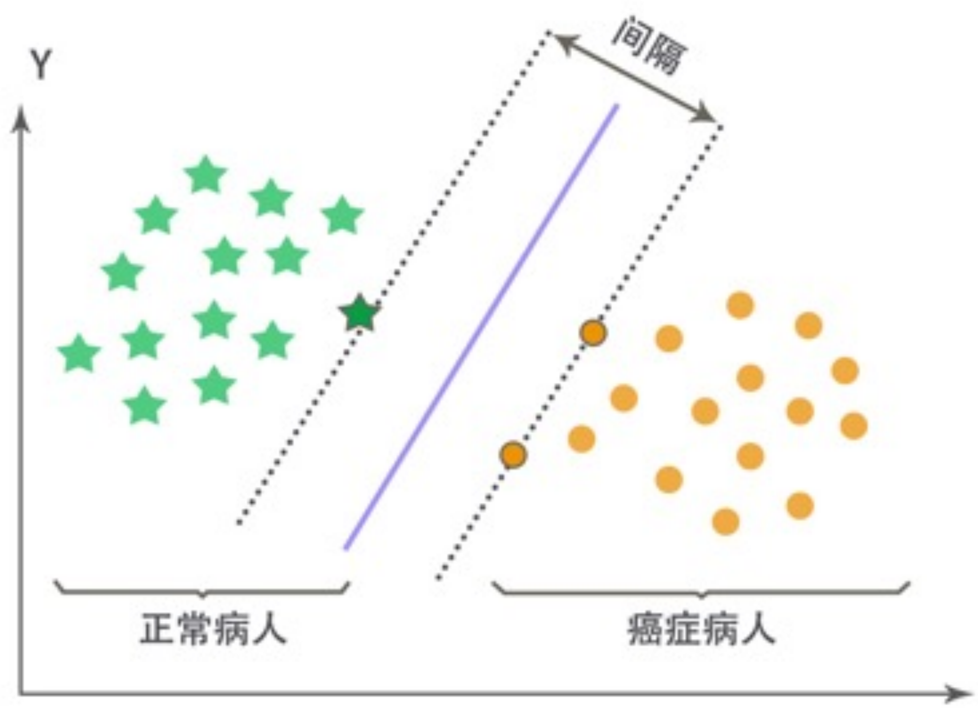
- Geometrical:
  - Nearest Neighbor
  - Logistic regression
  - Support Vector Machine
- Symbolism:
  - Decision Tree
- Connectionism:
  - Perceptron
  - Neural networks
- Bayesian:
  - Naive Bayes

Maximal Margin hyperplane

Maximum Margin Classifiers



哪一条分割线更好？



间隔最大化

SVM: maximal margin hyperplane

- SVM: the separating hyperplane such that the minimum distance of any training point to the hyperplane is the largest.

- 根据支持向量机模型设定

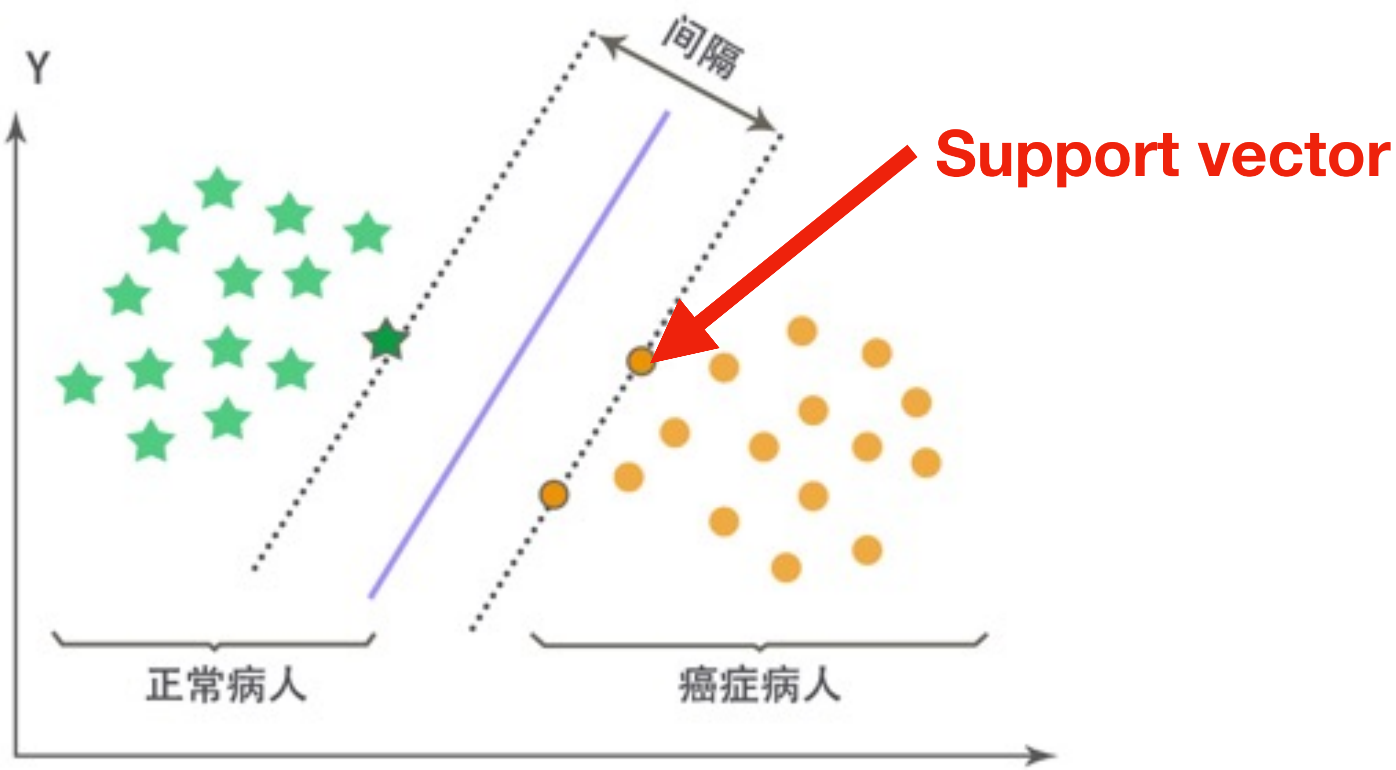
$$\left.\begin{array}{l} f(\boldsymbol{x}_i) > 0 \leftrightarrow y_i = +1 \\ f(\boldsymbol{x}_i) < 0 \leftrightarrow y_i = -1 \end{array}\right\} \rightarrow y_i f(\boldsymbol{x}_i) > 0$$

- 样本点到超平面距离：

$$\frac{y_i f(\boldsymbol{x}_i)}{||\boldsymbol{w}||} = \frac{y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b)}{||\boldsymbol{w}||}$$

Support Vector Machine

- 间隔：样本点到决策超平面的最小距离
- 目标函数（间隔最大化）：
- $$\arg\max_{\boldsymbol{w},b}\{\frac{1}{\|\boldsymbol{w}\|}\min_i[y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b)]\}$$



Linearly separable case

- 目标函数： $\arg\max_{\boldsymbol{w},b}\{\frac{1}{\|\boldsymbol{w}\|}\min_i[y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b)]\}$
- 假设：$\min_i[y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b)]$=1（w和b同比例缩放不影响结果）
- 则最优化问题为：

$$\max_{\boldsymbol{w},b}\frac{1}{||\boldsymbol{w}||}\quad s.t.\ y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b\right)\geq 1, i=1,2,\ldots,n$$

- 等价于：

$$\min_{\boldsymbol{w},b}\frac{1}{2}\|\boldsymbol{w}\|^2\quad s.t.\ \ y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b\right)\geq 1, i=1,2,\ldots,n$$

Linearly non-separable case

- 如果数据线性不可分，则增加松弛变量（Slack Variables）$\xi_i \geq 0$，使得间隔函数加上松弛变量大于等于1。此时，约束条件变成：

$$y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right) \geq 1 - \xi_i$$

- 目标函数：

$$\min_{\xi_i} \min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

$$s.t.\quad y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right) \geq 1 - \xi_i, i = 1,2,\dots,n; \xi_i \geq 0, i = 1,2,\dots,n$$

or use the hard constraint $\sum_{i=1}^{n} \xi_i \leq c$

Understanding the slack variable

Appendix: Primal-Dual support vector classifiers

$$\beta = w$$
$$\beta_0 = b$$

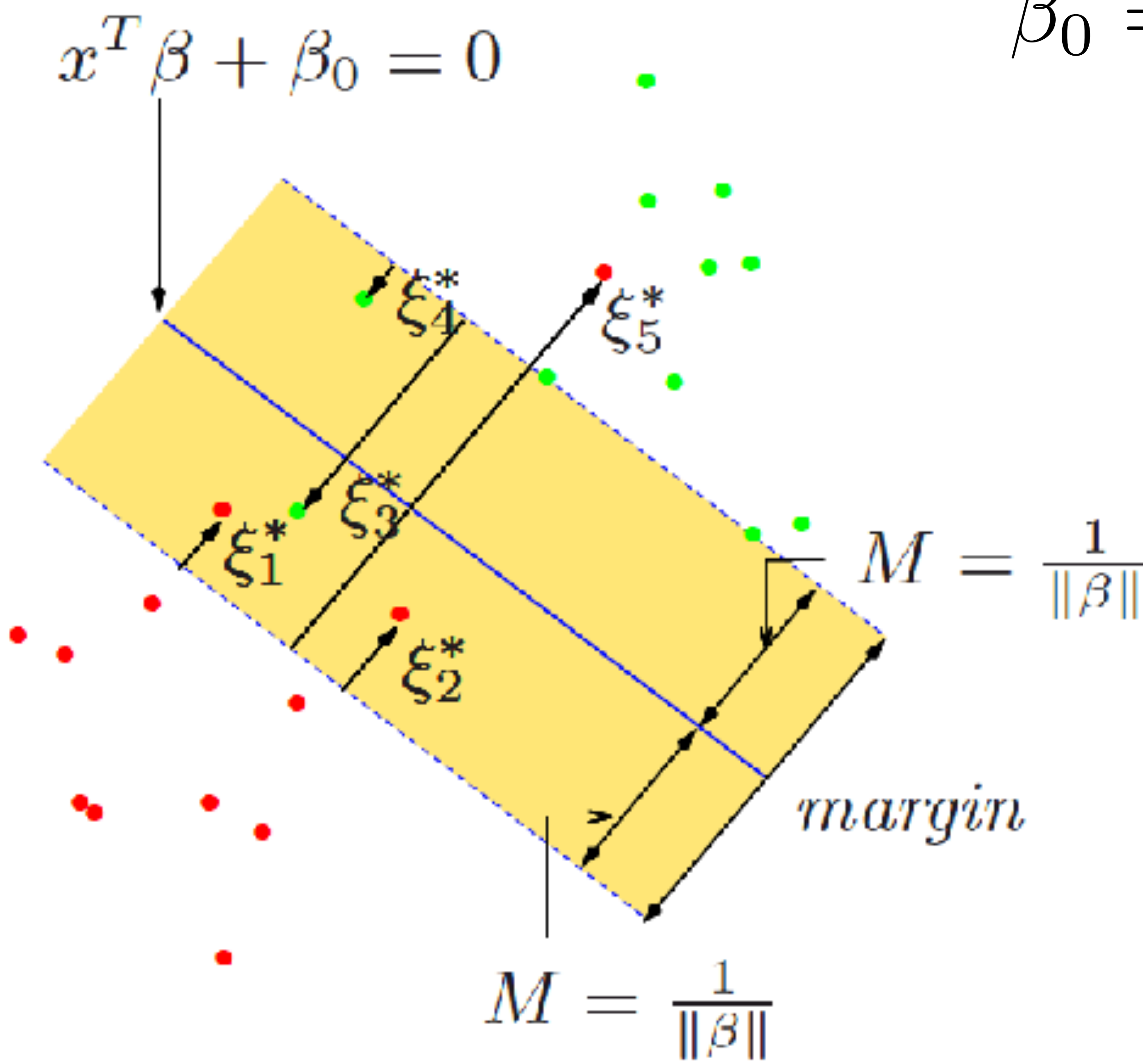


Figure: Separating hyperplane with margin

Understanding the slack variable

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

$$s.t.\quad y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) \geq 1 - \xi_i, i = 1,2,\ldots,n; \xi_i \geq 0, i = 1,2,\ldots,n$$

$\xi_i = 0$: the i-th observation is on the correct side of the margin

$\xi_i > 0$: the i-th observation is on the wrong side of the margin

$\xi_i > 1$: the i-th observation is on the wrong side of the hyperplane

thus mis-classified!

Understanding the objective

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n} \xi_i$$

$$s.t.\quad y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) \geq 1 - \xi_i, i = 1,2,\ldots,n; \xi_i \geq 0, i = 1,2,\ldots,n$$

Because any point that is misclassified has $\xi_i > 1$,

$\sum_i \xi_i$ is an upper bound on the number of misclassified data.

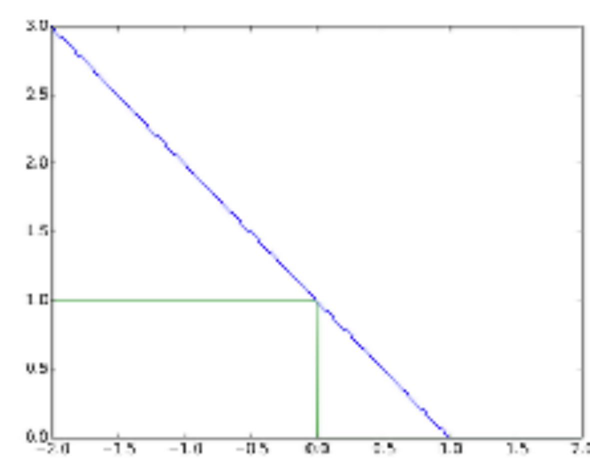
- A larger C assigns larger penalty to training errors.
- A smaller C gives more penalty to the model complexity.

Equivalent to hinge loss

$$\min_{\xi_i} \min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

$$s.t. \quad y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right) \geq 1 - \xi_i, i = 1,2,\dots,n; \xi_i \geq 0, i = 1,2,\dots,n$$

$$\min_{w,b} \frac{1}{2}||w||^2 + C\sum_{i=1}^{n} \underbrace{\max(0, 1 - y_i(w^T x_i + b))}_{\text{Hinge Loss}}$$



Solve Linearly separable case

- 目标函数： $\arg\max_{\boldsymbol{w},b}\{\frac{1}{\|\boldsymbol{w}\|}\min_i[y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b)]\}$
- 假设：$\min_i[y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b)]$=1（w和b同比例缩放不影响结果）
- 则最优化问题为：

$$\max_{\boldsymbol{w},b}\frac{1}{||\boldsymbol{w}||}\quad s.t.\ y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b\right)\geq 1, i=1,2,\ldots,n$$

- 等价于：

$$\min_{\boldsymbol{w},b}\frac{1}{2}\|\boldsymbol{w}\|^2\quad s.t.\ \ y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b\right)\geq 1, i=1,2,\ldots,n$$

Solve Linearly separable case

Lagrange multiplier method

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{n} \boldsymbol{\alpha}_i \{y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) - 1\}$$

- 原问题：极小极大问题,假设解为$\hat{p}$

$$\min_{\boldsymbol{w},b} \max_{\boldsymbol{\alpha}} L(\boldsymbol{w}, b, \boldsymbol{\alpha})$$

- 对偶问题：极大极小问题,假设解为$\hat{d}$

$$\max_{\boldsymbol{\alpha}} \min_{\boldsymbol{w},b} L(\boldsymbol{w}, b, \boldsymbol{\alpha})$$

$\hat{p} \geq \hat{d}$, 满足强对偶性时，$\hat{p} = \hat{d}$，原问题和对偶问题等价

Slater条件:duality gap = 0

Slater's theorem provides a *sufficient condition* for strong duality to hold. Namely, if

- The primal problem is convex;
- It is strictly feasible, that is, there exists $x_0 \in \mathbf{R}^n$ such that

$$Ax_0 = b, \quad f_i(x_0) < 0, \quad i = 1, \ldots, m,$$

then, strong duality holds: $\hat{p} = \hat{d}$, and the dual problem is attained. (Proof)

- $A\boldsymbol{x}_0 = b$，优化问题的等式约束；$\mathrm{f}_i(\boldsymbol{x}_0) \leq 0$, 优化问题的不等式约束
- 支持向量机的原问题满足Slater条件，因此可以通过对偶问题来求原问题

一般优化问题	支持向量机
$A\boldsymbol{x}_0 = b$	无
$\mathrm{f}_i(\boldsymbol{x}_0) \leq 0$	$1 - y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) \leq 0$

除了支持向量，大部分样本点均满足$\mathbf{1} - \boldsymbol{y}_i(\boldsymbol{w}^{\mathbf{T}}\boldsymbol{x}_i + \boldsymbol{b}) < \mathbf{0}$

Lagrange multiplier method

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{n} \boldsymbol{\alpha}_i \left\{y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right) - 1\right\}$$

- 对偶问题： $\max\limits_{\boldsymbol{\alpha}} \min\limits_{\boldsymbol{w},b} L(\boldsymbol{w}, b, \boldsymbol{\alpha})$
- 先求解内部的最小化问题
- 拉格朗日函数 $L(\boldsymbol{w}, b, \boldsymbol{\alpha})$ 分别对参数 $\boldsymbol{w}$ 和 $\boldsymbol{b}$ 求偏导并令其等于0可得：

$$\frac{\partial L}{\partial \boldsymbol{w}} = 0 \Rightarrow \boldsymbol{w} = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow 0 = \sum_{i=1}^{n} \alpha_i y_i$$

Lagrange multiplier method

- 将上述结果代入 $L(\boldsymbol{w}, b, \boldsymbol{\alpha})$ 中，可得：

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{n} \alpha_i \left[y_i \left(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i + b \right) - 1 \right]$$

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w} - \boldsymbol{w}^{\mathrm{T}} \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i - b \sum_{i=1}^{n} \alpha_i y_i + \sum_{i=1}^{n} \alpha_i$$

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j$$

Dual Problem

- 关于$\boldsymbol{w}$和b的问题转化为关于α的最优化问题：

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i\alpha_j y_i y_j \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j - \sum_{i=1}^{n}\alpha_i$$

$$s.t.\ \alpha_i \geq 0, i = 1,2,\ldots,n$$

$$\sum_{i=1}^{n}\alpha_i y_i = 0$$

- 二次凸规划问题，一般的数值计算方法可求解。
- 更高效的求解算法：**SMO算法**　**when sample size n is large.**

SVM example

- 给定 3 个数据点：正例点 $\boldsymbol{x}_1 = (3,3)^{\mathrm{T}}, \boldsymbol{x}_2 = (4,3)^{T}$，负例点$\boldsymbol{x}_3 = (1,1)^{\mathrm{T}}$，求线性可分支持向量机
- 目标函数：

$$\min_{\boldsymbol{\alpha}} \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (\boldsymbol{x}_i \boldsymbol{x}_j) - \sum_{i=1}^{n} \alpha_i$$

$$= \frac{1}{2}(18\alpha_1^2 + 25\alpha_2^2 + 2\alpha_3^2 + 42\alpha_1\alpha_2 - 12\alpha_1\alpha_3 - 14\alpha_2\alpha_3) - \alpha_1 - \alpha_2 - \alpha_3$$

$$\text{s.t.} \;\; \alpha_1 + \alpha_2 - \alpha_3 = 0; \alpha_i \geq 0, i = 1,2,3$$

SVM example

- 将 $\alpha_1+\alpha_2=\alpha_3$ 代入目标函数，得到关于 α_1 和 α_2 的函数：

$$s(\alpha_1,\alpha_2)=4\alpha_1^2+7.5\alpha_2^2+10\alpha_1\alpha_2-2\alpha_1-2\alpha_2$$

- 对 α_1 和 α_2 求偏导并令其等于零，易知 $s(\alpha_1,\alpha_2)$ 在点 $(1.5,-1)$ 处取得极值。而该点不满足条件 $\alpha_2\geq 0$，所以最小值在边界处达到。

- 当 $\alpha_1=0$ 时，最小值 $s\left(0,\frac{2}{13}\right)=-\frac{2}{13}=-0.1538$

- 当 $\alpha_2=0$ 时，最小值 $s\left(\frac{1}{4},0\right)=-\frac{1}{4}=-0.25$

- 即 $s(\alpha_1,\alpha_2)$ 在 $\alpha_1=0.25,\alpha_2=0$ 时达到最小，此时 $\alpha_3=\alpha_1+\alpha_2=0.25$

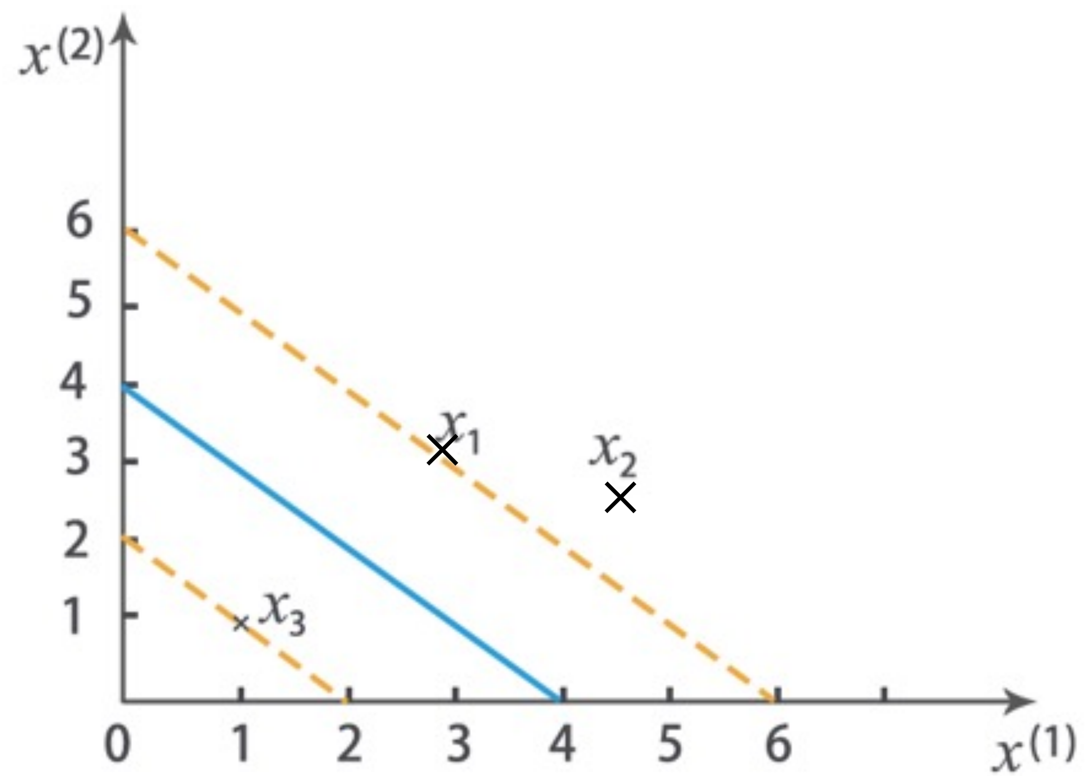
SVM example

- $\alpha_1 = \alpha_3 = 1/4$，对应的点 x_1 和 x_3 是支持向量
- 代入公式可得：

$$w_1 = w_3 = 0.5, b = -2$$

- 分离超平面为：$\frac{1}{2}x_1 + \frac{1}{2}x_3 - 2 = 0$
- 决策函数为：

$$f(x) = \text{sign}(\frac{1}{2}x_1 + \frac{1}{2}x_3 - 2)$$



Solve Linearly non-separable case

- 如果数据线性不可分，则增加松弛变量（Slack Variables）$\xi_i \geq 0$，使得间隔函数加上松弛变量大于等于1。此时，约束条件变成：

$$y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right) \geq 1 - \xi_i$$

- 目标函数：

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

$$s.t.\quad y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right) \geq 1 - \xi_i, i = 1,2,\ldots,n; \xi_i \geq 0, i = 1,2,\ldots,n$$

Lagrange multiplier method

- 拉格朗日函数

$$L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i\left(y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right) - 1 + \xi_i\right) - \sum_{i=1}^{n}\mu_i\xi_i$$

- 对应的KKT条件：

$$\begin{cases} \alpha_{\mathrm{i}} \geq 0 \\ y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) - 1 + \xi_i \geq 0 \\ \alpha_{\mathrm{i}}\left(y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) - 1 + \xi_i\right) = 0 \\ \mu_i \geq 0 \\ \xi_i \geq 0 \\ \mu_i\xi_i = 0 \end{cases}$$

$$\frac{\partial L}{\partial \boldsymbol{w}} = 0 \Rightarrow \boldsymbol{w} = \sum_{i=1}^{n}\alpha_i y_i \boldsymbol{x}_i$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow 0 = \sum_{i=1}^{n}\alpha_i y_i$$

$$\frac{\partial L}{\partial \xi_i} = C - \alpha_i - \mu_i = 0$$

Dual problem

代入函数 L 中，得到

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu}) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \, (x_i^T x_j) + \sum_{i=1}^{n} \alpha_i$$

- 对上式求关于 $\boldsymbol{\alpha}$ 的极大值，得到：

$$\max_{\boldsymbol{\alpha}} -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \, (\boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j) + \sum_{i=1}^{n} \alpha_i$$

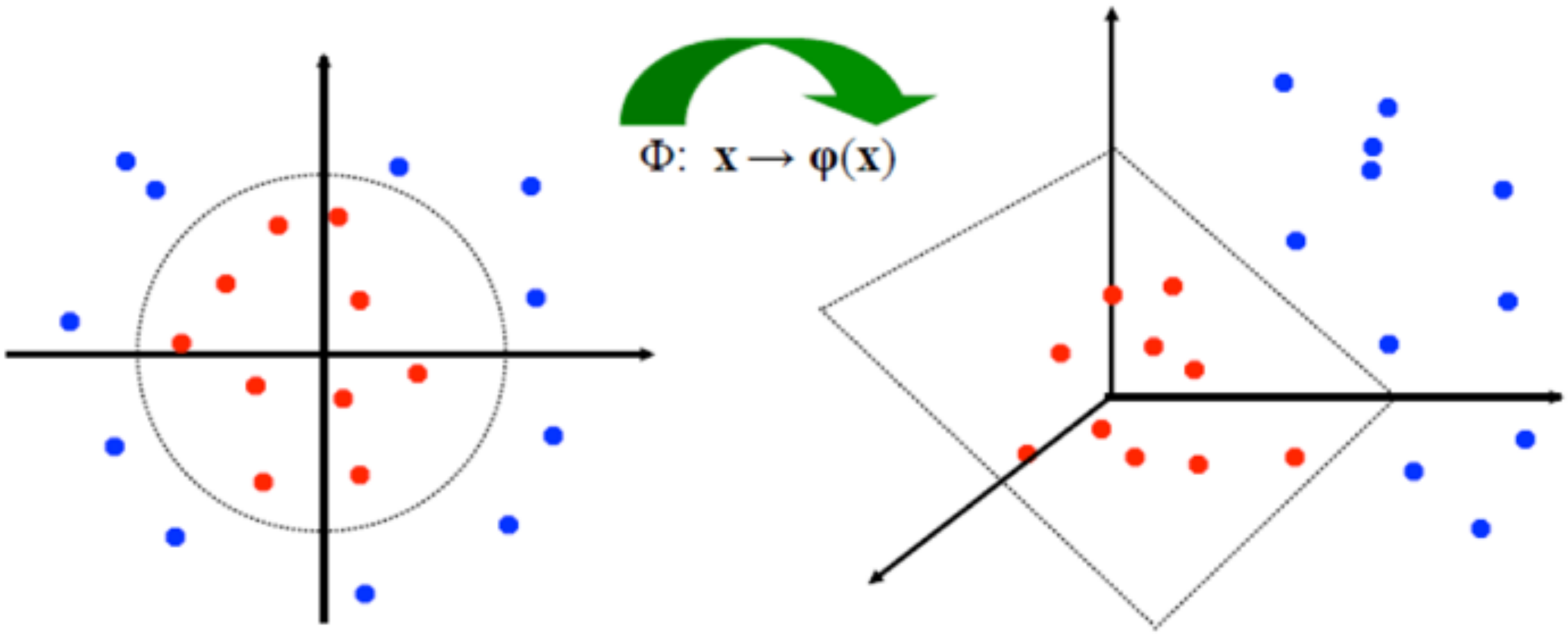
$$\text{s.t.} \quad \sum_{i=1}^{n} \alpha_i y_i = 0\,;$$

$$\alpha_i \geq 0;$$

$$0 \leq \alpha_i \leq C$$

Non-linear (Kernel) SVM

- 如果分类边界是非线性的，该怎么处理呢？
  - 可以将原始数据映射到更高维的空间中（核函数！）



Non-linear (Kernel) SVM

Kernel K is a function: $\mathcal{X} \times \mathcal{X} \to \mathbb{R}$

Linear SVM classifier: $f(x) = sign(\sum_{i=1}^{n} \alpha_i y_i x_i^T x + b)$

Non-Linear SVM: $f(x) = sign(\sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + b)$

- 支持向量机模型可以通过核方法来处理线性不可分的数据。
- 核方法的基本原理是把原坐标系里线性不可分的数据使用核函数（Kernel）投影到另一个空间，尽量使得数据在新的空间里线性可分。
- 要在支持向量机中使用核函数，只需要将对偶问题中目标函数中的内积项替换成核函数。

Property of Kernel

Kernel K is a function: $\mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$

- 核函数（核矩阵）满足对称和半正定性质
- 如果K_1和K_2是合法核函数：
- $\mathrm{K}_1 + \mathrm{K}_2$也是合法核函数
- $c\mathrm{K}_1$也是合法核函数，$c > 0$
- $a\mathrm{K}_1 + b\mathrm{K}_2$也是合法核函数,$a > 0, b > 0$

Some Examples of Kernel

Kernel K is a function: $\mathcal{X} \times \mathcal{X} \to \mathbb{R}$

- 多项式核函数： $(\boldsymbol{x}_1^{\mathrm{T}}\boldsymbol{x}_2 + 1)^d$ ，d为整数
- 高斯核函数(RBF核函数)： $\exp(-\frac{\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2}{2\delta^2}), \delta > 0$
- Fisher 核函数(Sigmoid核函数)：$\tanh\left(\beta\boldsymbol{x}_1^{\mathrm{T}}\boldsymbol{x}_2 + \theta\right), \beta > 0, \theta < 0$
- 拉普拉斯核函数：$\exp\left(-\frac{\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|}{\delta}\right), \delta > 0$

Non-Linear SVM: $f(x) = sign(\sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + b)$

Software for SVM

- LIBSVM 是台湾大学林智仁(Lin Chih-Jen)教授等开发设计的一个简单、易于使用和快速有效的SVM模式识别与回归的软件包
- 该软件对SVM所涉及的参数调节相对比较少，提供了很多的默认参数，利用这些默认参数可以解决很多问题；并提供了交互检验(Cross Validation)的功能
- 该软件可以解决C-SVM、ν-SVM、ε-SVR和ν-SVR等问题，包括基于一对一算法的多类模式识别问题

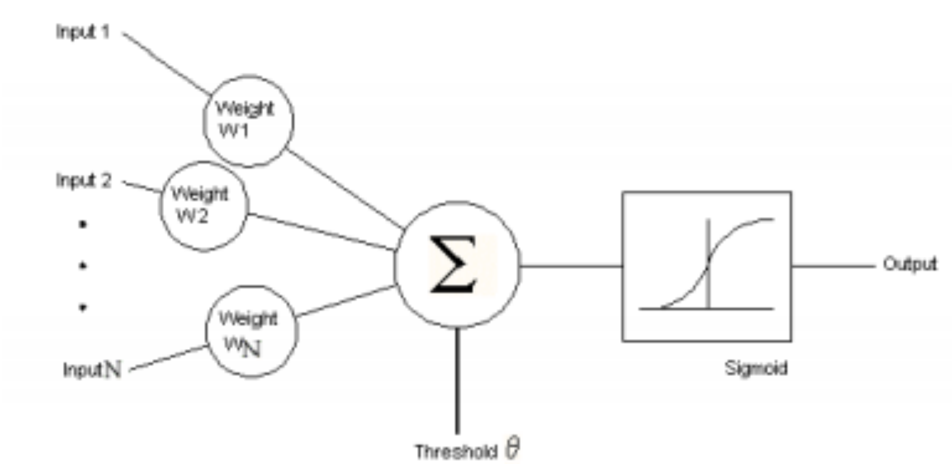
Classification methods

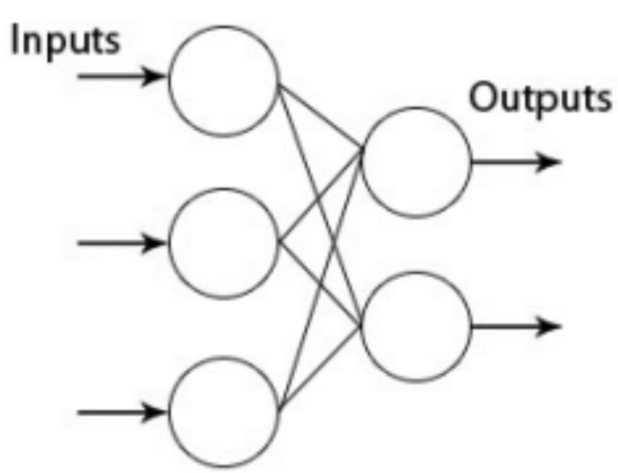
- Geometrical:
  - Nearest Neighbor
  - Logistic regression
  - Support Vector Machine
- Symbolism:
  - Decision Tree
- Connectionism:
  - Perceptron
  - Neural networks
- Bayesian:
  - Naive Bayes

Neural Networks - Brief history

感知机

- 在1960年左右，Frank Rosenblatt 发明了第一代神经网络
- 第一代神经网络，感知机，能够识别一些简单的图形，如三角形、正方形。人们意识到一种可以像人类一样感知，学习，记忆的人工智能或许可以被创造出来。
- 但是，Marvin Minsky（1969）指出，单层结构限制了感知机能够学习到的函数，例如一个异或函数就已经超出了它的学习能力





Neural Networks - Brief history

第二代 神经网络

- 1985年Geoffrey Hinton 在感知机的基础上用一些隐藏层代替原始的单一结构，开创了第二代神经网络
  - 通过后向传播算法（Back-Propagation，即BP）进行训练

- 在1989，Yann LeCun 等人构建了一个深度神经网络来完成手写体识别的任务
  - LeCun的算法取得了巨大的成功，但网络的训练时间却将近3天

Neural Networks - Brief history

第二代 神经网络

缺陷：

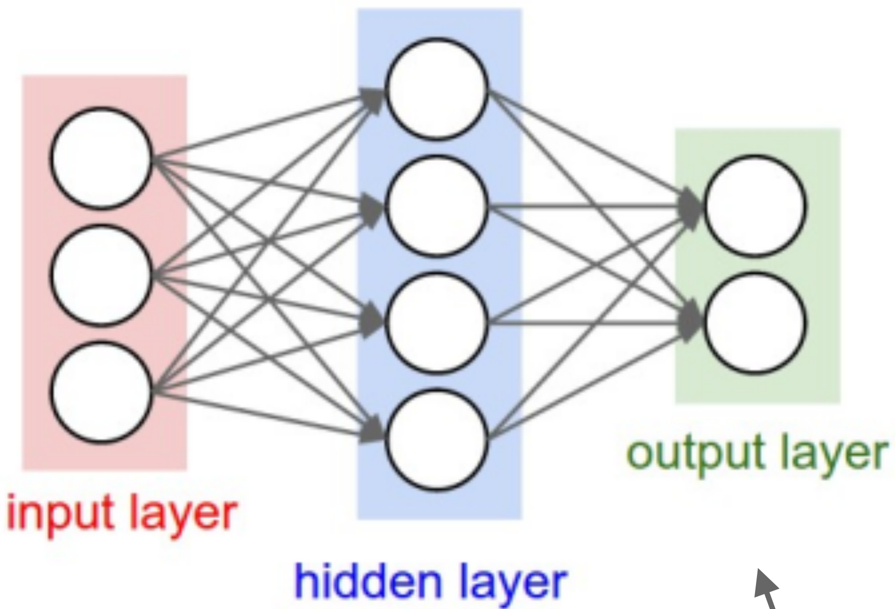
- 不能训练未标注的数据，但现实中大多数数据都是未标注的。
- 修正信号在通过多个隐藏层传输时被减弱
- 当包含的隐藏层过多时，学习速度太慢
- 会陷于局部最优解

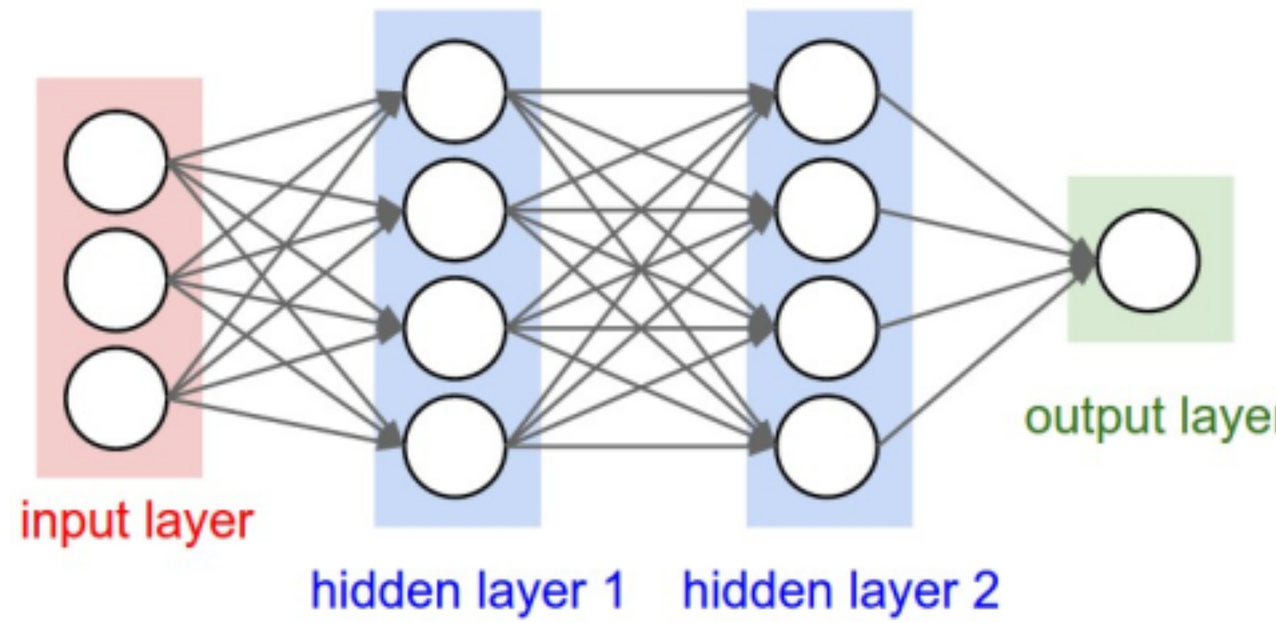
SVM 延缓了深度学习的发展

- 当人们努力去改进Hinton的神经网络时，1993-1995 Vladimir N. Vapnik,等人在原始的感知机的基础上发明了支持向量机（Support Vector Machine）

Neural Networks

Feed-Forward NN，前向传递网络构架：



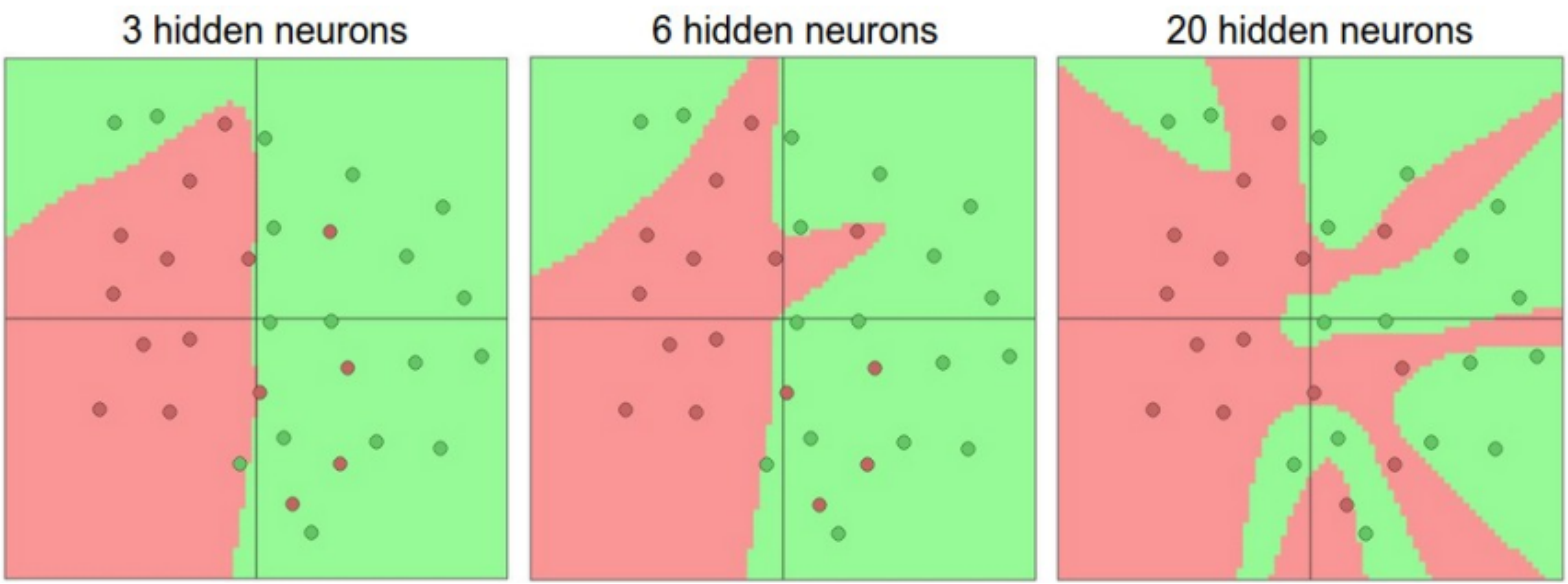


“2-layer NN”, or
“1-hidden-layer（隐层） NN”

“3-layer NN”, or
“2-hidden-layer NN”

“Fully-connected” Layers

Neural Networks



Classification methods

- Geometrical:
  - Nearest Neighbor
  - Logistic regression
  - Support Vector Machine
- Symbolisme:
  - Decision Tree
- Connectionism:
  - Perceptron
  - Neural networks
- Bayesian:
  - Naive Bayes

Naive Bayes

- 基于**贝叶斯定理**和特征**条件独立性假设**的分类方法
- 典型应用场景
  - 新闻分类
  - 疾病分类
  - 情感分类
  - 垃圾邮件分类

Bayes Formula

- 假设X, Y是一对随机变量，它们的联合概率$p(X = x, Y = y)$是指X取值x且Y取值y的概率，条件概率$p(Y = y \mid X = x)$是指变量在X取值x的情况下，变量Y取值y的概率
- 联合概率和条件概率满足

$$p(X, Y) = p(Y|X) \cdot p(X) = p(X|Y) \cdot p(Y)$$

- 进而得到贝叶斯定理

$$\overset{\text{后验概率}}{p(Y|X)} = \frac{\overset{\text{似然函数}}{p(X|Y)} \cdot \overset{\text{先验概率}}{p(Y)}}{\underset{\text{证据}}{p(X)}}$$

Bayes Formula

- 设特征向量$X = \{X_1, X_2, \dots, X_m, \dots, X_d\}$是$d$ 维随机向量，类标签$Y \in \{1,2,\dots,c\}$，样本数量为n
- $p(X, Y)$是X和Y的联合概率分布
- 贝叶斯算法通过学习联合概率分布，利用贝叶斯公式，计算后验概率分布
- 先验概率分布为

$$p(Y = k), k = 1,2, \dots, c$$

- 条件概率分布

$$p(X|Y = k) = p(X_1, \dots X_d|Y = k), k = 1,2, \dots, c$$

Bayes Formula

Maximum A-posteriori Inference(MAP)

- 利用贝叶斯定理进行预测

$$p(Y=\mathrm{k} \mid \mathrm{X}) = \frac{p(X \mid Y=\mathrm{k})\, p(Y=\mathrm{k})}{p(\mathrm{X})}$$

- 对于某一个样本X，$p(X)$取值固定，上述预测等价于

$$\max_{k} \mathrm{p}(X \mid Y = \mathrm{k}) \cdot \mathrm{p}(Y = \mathrm{k})$$

Naive Bayes Assumption

- 如何简化$p(X \mid Y=k)$的计算(X是一个d维向量)
- 条件独立性假设

$$p(X \mid Y=k) = p(X_1, X_2, \ldots, X_d \mid Y=k)$$

$$= p(X_1 \mid Y=k)\ p(X_2 \mid Y=k) \cdots p(X_d \mid Y=k)$$

Example: conditional independent Dice:

given Y=1, {X1 = 1,2,3} _||_ {X2 = 1,2,3};

given Y=-1, {X1 = 4,5,6} _||_ {X2 = 4,5,6}.

But {X1 = 1,2,3,4,5,6} _||_ {X2 = 1,2,3,4,5,6} is wrong!

Parameter estimation

- 即估计先验概率分布$p(Y=k)$和条件概率分布$p(X_m \mid Y=k)$，通常使用极大似然估计，先验概率$p(Y=k)$的极大似然估计是

$$p(Y=k)=\frac{\sum_{i=1}^{n} \mathrm{I}(y_i=k)}{n}, k=1,2,\ldots,c$$

- 其中I(·)是指示函数，参数为真时取值1，反之取值0估计条件概率分布时，考虑随机向量X的特征为离散或连续情况
- 特征为离散型时，设$X_n \in \{1,2,\ldots,s\}$, 条件概率$p(X_n=s \mid Y=k)$的极大似然估计为

$$p(X_m=s \mid Y=k)=\frac{\sum_{i=1}^{n} \mathrm{I}(x_{im}=s, y_j=k)}{\sum_{i=1}^{n} \mathrm{I}(y_i=k)}$$

Parameter estimation

- 当特征为连续型时，有两种处理方法
  1. 可以将连续变量离散化，人为设定离散区间，当连续特征离散化后，可利用上述方法估计条件概率分布
  2. 假定连续变量服从某种分布，然后用数据集训练此分布参数
     - 高斯分布（正态分布）通常用来表示连续变量的概率分布；设分布的均值和方差分别为μ和σ^2；对于某类$Y = k$，特征X_m的条件概率分布为

$$p(X_n = x_{im}|Y = k)=\frac{1}{\sqrt{2\pi}\sigma}e^{\frac{(x_{im}-\mu)^2}{2\sigma^2}}$$

     - 利用极大似然估计，可知样本均值和方差可作为μ和σ^2的估计