西安交通大学 XI'AN JIAOTONG UNIVERSITY | IAIR Est. 1986 Institute of Artificial Intelligence and Robotics | 人工智能学院 College of Artificial Intelligence, XJTU

《高级机器学习》第二章

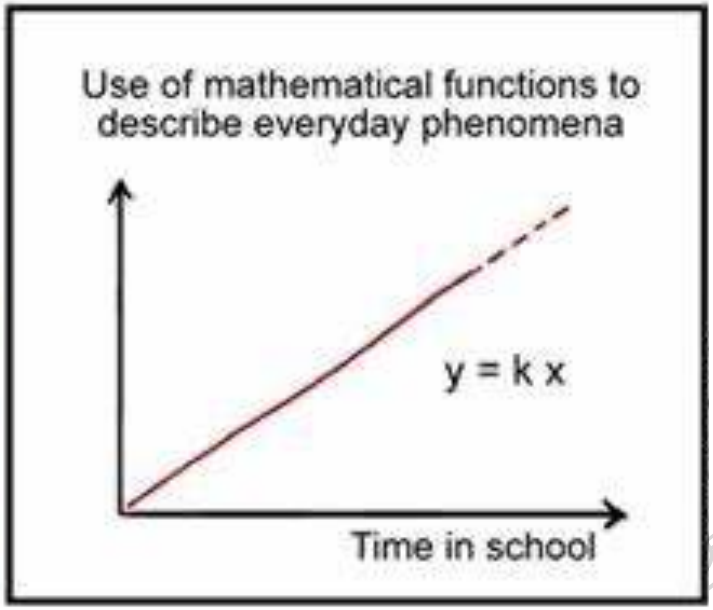
线性模型与核方法

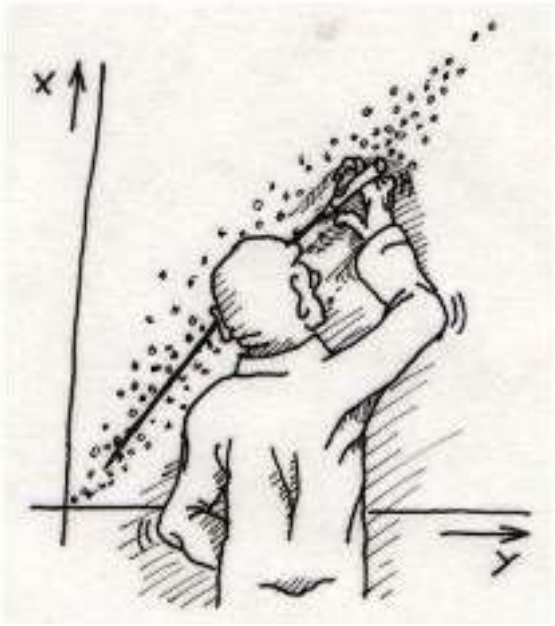
Linear Model and Kernel Method

魏平
西安交通大学人工智能学院
人工智能与机器人研究所

CONTENTS

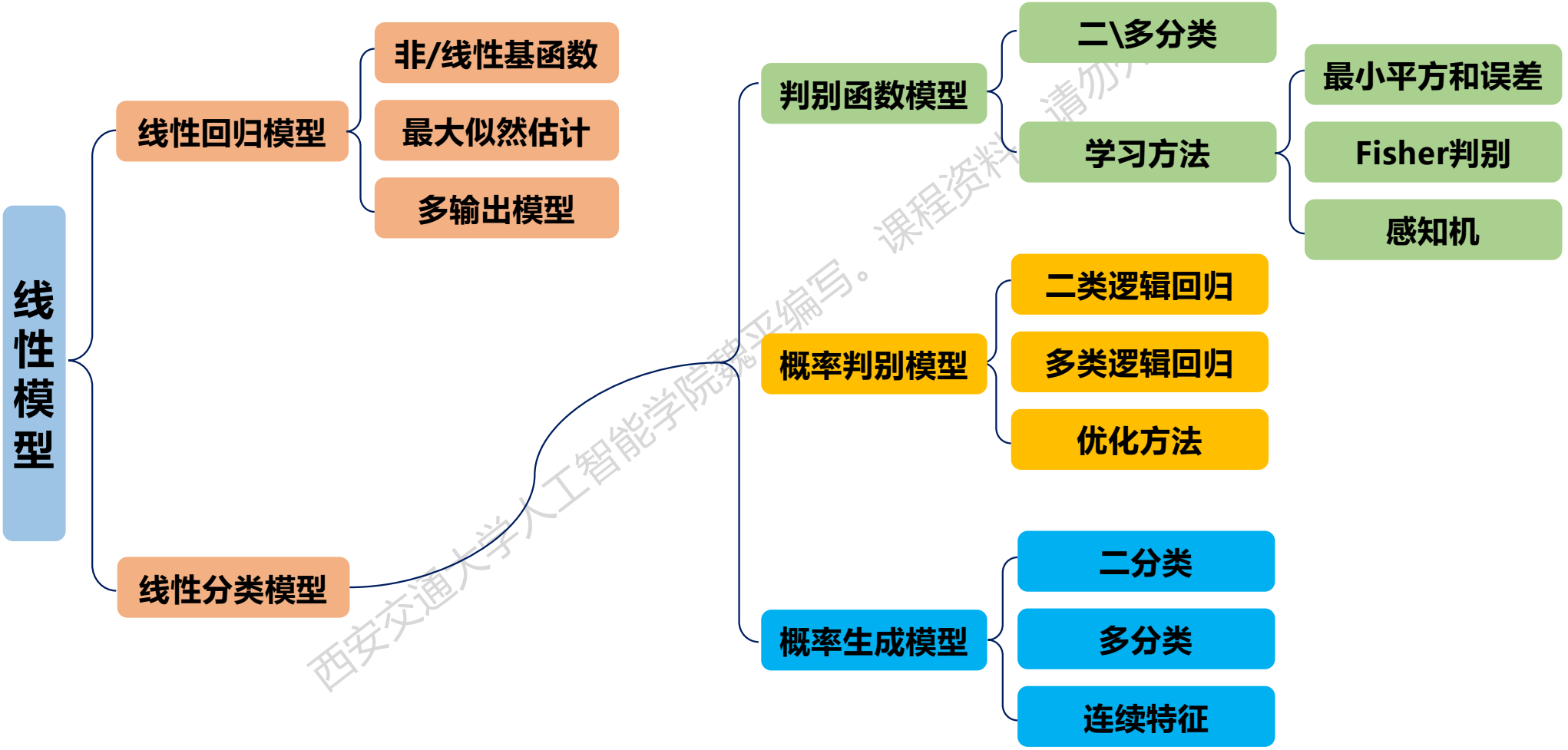
线性(Linear)和非线性(Nonlinear)





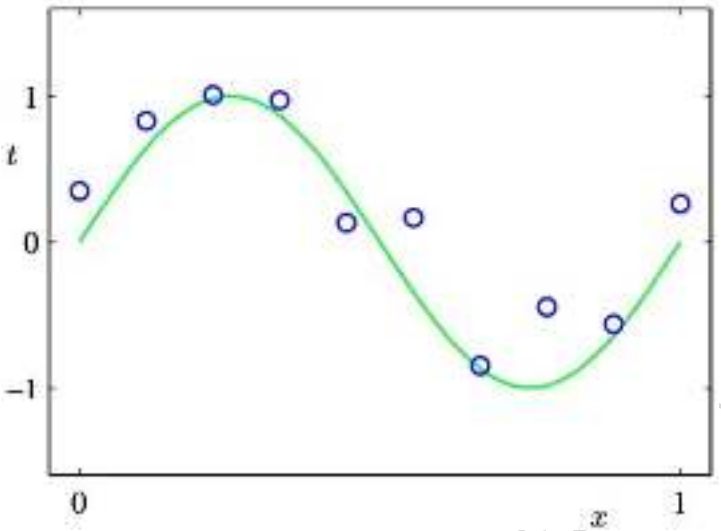
- **线性：参数以线性的形式存在，** $y(\mathrm{x}, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 \ldots + w_9 x^9$
- **非线性：参数以非线形的形式存在，** $y(\mathrm{x}, \mathbf{w}) = \exp\{w_0 w_1 + w_1 w_2 x + w_2 w_3 x\}$

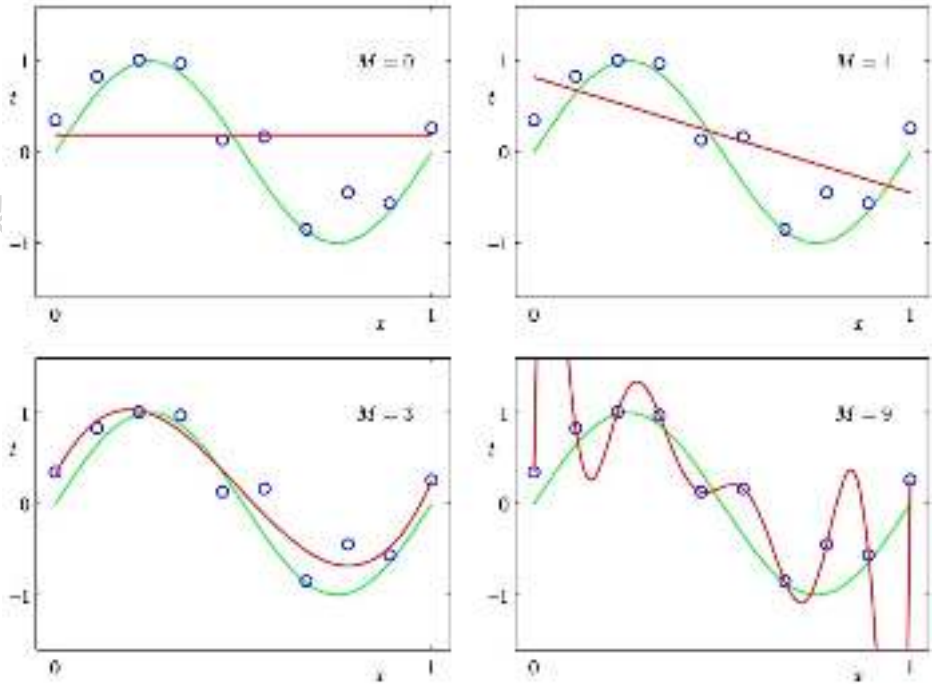
线性模型分类



2.1.1 线性回归模型

- 例：数据拟合，给定N个输入数据$(x_1, x_2, \ldots x_N)$以及对应函数值$(t_1, t_2, \ldots t_N)$，构建一个函数对一个新的输入x预测对应的目标值





假设1：

$y = \sin(ax)$

假设2:

$y(\mathrm{x}, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 \ldots + w_M x^M$

2.1.1 线性回归模型

- 训练数据包含N个输入数据 $(\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_N)$以及对应函数值 $(t_1, t_2, \ldots t_N)$，回归任务是对一个新的输入$\mathbf{x}$预测对应的目标值
- 可以利用线性模型建立输入输出之间的关系：

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + \cdots w_D x_D = \mathbf{w}^{\mathrm{T}} \mathbf{x}$$

$$\mathbf{w} = (w_0, w_1, \ldots, w_D)^{\mathrm{T}} \qquad \mathbf{x} = (1, x_1, \ldots, x_D)^{\mathrm{T}}$$

线性：参数$\mathbf{w}$的线性函数

- 输入线性组合限制了模型能力，引入**非线性基函数**对输入进行变换

$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x})$$

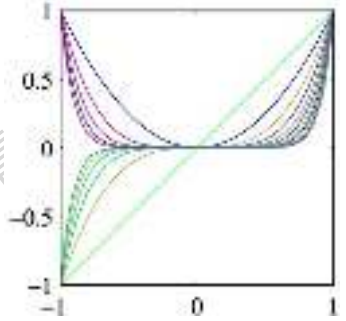
$$\mathbf{w} = (w_0, w_1, \ldots, w_{M-1})^{\mathrm{T}}$$

$$\boldsymbol{\phi} = (\phi_0, \phi_1, \ldots, \phi_{M-1})^{\mathrm{T}}, \phi_0(\mathbf{x}) = 1$$

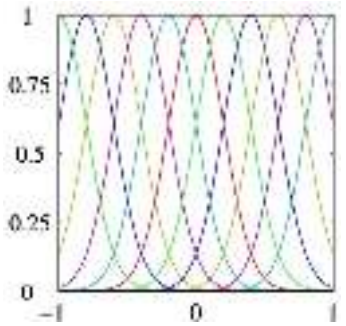
2.1.1线性回归模型

- 基函数 $\{\phi_j(\mathbf{x})\}$使得模型成为输入数据的非线性函数，提高了模型能力

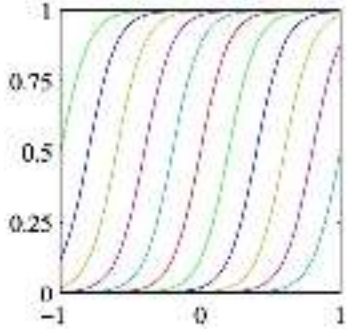
基函数	形式
恒等基函数	$\phi(\mathbf{x}) = \mathbf{x}$
多项式（幂）基函数	$\phi_j(\mathrm{x}) = x^j$
高斯（径向）基函数	$\phi_j(\mathrm{x}) = \exp\left\{-\frac{(x-\mu_j)^2}{2s^2}\right\}$
反曲基函数	$\phi_j(\mathrm{x}) = \sigma\left\{\frac{x-\mu}{s}\right\}$ $\sigma(a) = \frac{1}{1+\exp(-a)}$



幂基函数



高斯基函数



反曲基函数

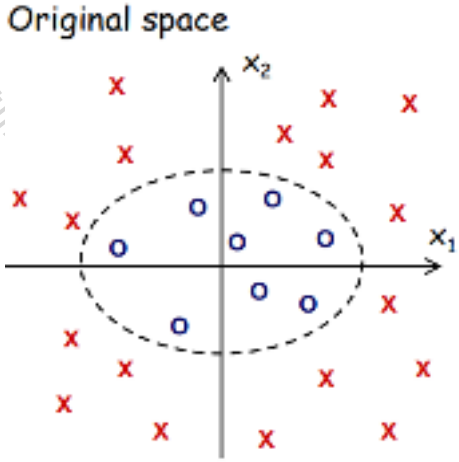
2.1.1 线性回归模型

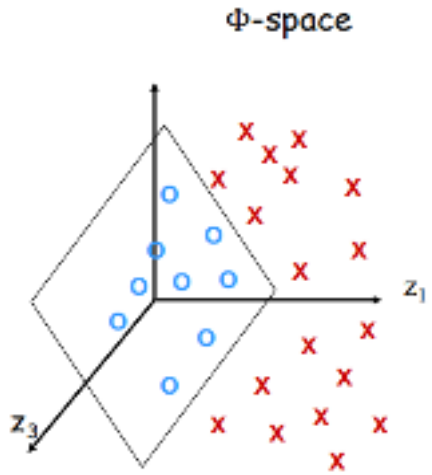
- 基函数 $\{\phi_j(\mathbf{x})\}$可以看作是对原始数据的特征提取或变换过程

输入图像

HOG 特征

例1：广义特征提取





$\phi: R^2 \rightarrow R^3$ $\phi(\boldsymbol{x}) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$

例2：特征变换

2.1.1求解线性回归模型

- 训练数据为 N 个输入数据 $\boldsymbol{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_N)$ 及对应函数值 $\mathbf{t} = (t_1, t_2, \ldots t_N)$，模型为线性回归模型 $y(\mathbf{x}, \mathbf{w}) = \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x})$
- 假设目标变量为线性模型加一个高斯噪声

$$t = y(\mathbf{x}, \mathbf{w}) + \epsilon$$

ϵ是一个0均值、精度(方差倒数)为β的高斯函数，则有

$$p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}), \beta^{-1})$$

- **似然函数为**：$p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1})$

2.1.1求解线性回归模型

- **参数的最大似然估计为**：

$$(\mathbf{w},\beta)_{\mathrm{ML}}=\arg\max_{\mathbf{w},\beta}\ \ln p(\mathbf{t}|\mathbf{X},\mathbf{w},\beta)$$

- **对数似然函数为**：

$$\ln p(\mathbf{t}|\mathbf{X},\mathbf{w},\beta)=\sum_{n=1}^{N}\ln\mathcal{N}(t_n|\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n),\beta^{-1})$$

$$=\frac{N}{2}\ln\beta-\frac{N}{2}\ln(2\pi)-\beta E_D(\mathbf{w})$$

平方和误差函数 (sum-of-squares error function)

$$E_D(\mathbf{w})=\frac{1}{2}\sum_{n=1}^{N}\{t_n-\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2$$

2.1.1 求解线性回归模型

- **对w 求导**:

$$\nabla \ln p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \sum_{n=1}^{N} \{t_n - \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_n)\} \boldsymbol{\phi}(\mathbf{x}_n)^{\mathrm{T}}$$

- **令导数为0得到**:

$$\mathbf{w}_{\mathrm{ML}} = \left(\boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{\Phi}\right)^{-1} \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{t}$$

$$\boldsymbol{\Phi} = \begin{bmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{bmatrix}$$

2.1.1 求解线性回归模型

- 随机梯度下降算法（Stochastic Gradient Descent, SGD）
  - 解析式有时很难得到；数据量大时，数据矩阵操作计算量大
  - 设误差函数由每个样本点的误差组成 $E = \sum_{n=1}^{N} E_n$，随机梯度下降算法在第τ步时，通过以下迭代式子更新参数

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E_n$$

序贯学习 (Sequential Learning)
or
在线学习 (On-line Learning)

  - 对平方和误差函数，通过以下式进行迭代

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} + \eta\{t_n - \mathbf{w}^{(\tau)\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}\boldsymbol{\phi}(\mathbf{x}_n)$$

2.1.1 多输出(Multiple Outputs)线性回归模型

- 计算的单点输出结果是一个K维向量 $\boldsymbol{t} = (t_1, t_2, \ldots t_K)^{\mathrm{T}}$
  - 思路一：对$\boldsymbol{t}$的每一个元素成分使用不同的基函数组，进行K个独立的单输出回归
  - 思路二：对$\boldsymbol{t}$的所有元素成分使用相同的基函数组，进行联合回归

- **联合回归模型：** $\boldsymbol{y}(\mathbf{x}, \boldsymbol{W}) = \boldsymbol{W}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x})$

  $\boldsymbol{y}$: $K\times1$维输出向量　$\boldsymbol{W}$: $M\times K$维参数矩阵　$\boldsymbol{\phi}(\mathbf{x})$: $M\times1$维基函数向量

- **多维高斯形式：** $p(\boldsymbol{t}|\mathbf{x}, \boldsymbol{W}, \beta) = \mathcal{N}(\boldsymbol{t}|\boldsymbol{W}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}), \beta^{-1}\boldsymbol{I})$

2.1.1 多输出(Multiple Outputs)线性回归模型

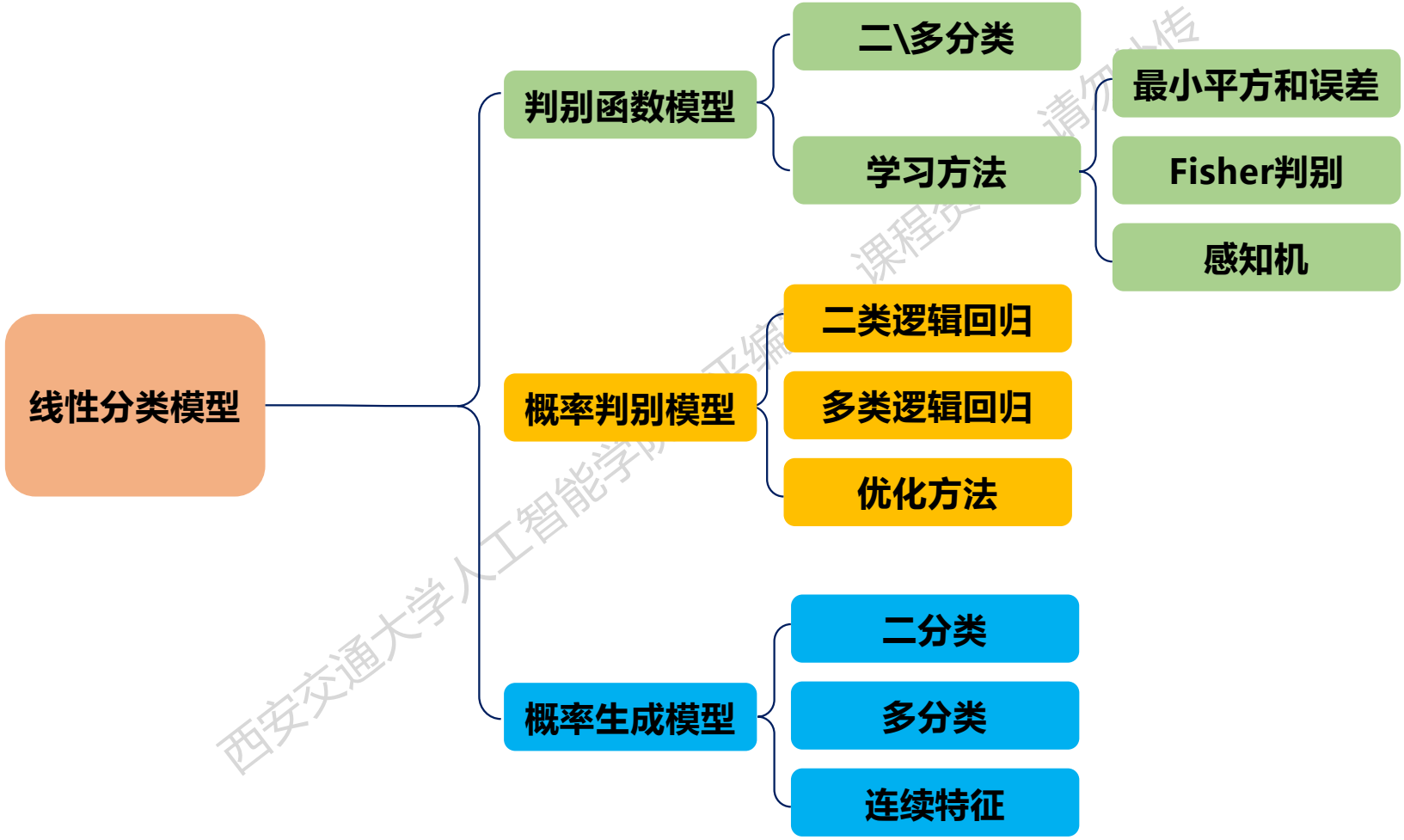
- 给定N个输入数据 $(\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_N)$及对应输出值 $(\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots \boldsymbol{t}_N)$，学习多输出线性回归模型的参数
  - 将输出数据组织为一个$N \times K$维矩阵$\boldsymbol{T}$，其第n行是$\boldsymbol{t}_n^{\mathrm{T}}$，将输入组织为$\boldsymbol{X}$
  - **对数似然函数为**：

$$\ln p(\boldsymbol{T}|\boldsymbol{X}, \boldsymbol{W}, \beta) = \sum_{n=1}^{N} \ln \mathcal{N}(\boldsymbol{t}_n|\boldsymbol{W}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}\boldsymbol{I})$$

$$= \frac{NK}{2}\ln(\frac{\beta}{2\pi}) - \beta \sum_{n=1}^{N} \left\| \boldsymbol{t}_n - \boldsymbol{W}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n) \right\|^2$$

  - **优化结果**：$\mathbf{W}_{\mathrm{ML}} = \left(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{T}$

2.1.2 线性分类模型



2.1.2 判别函数 (Discriminant Function)方法

- 判别函数的功能是将输入特征向量$\mathbf{x}$分配到K个类别中的某一类C_k，它将输入空间分为不同的决策区域，每个区域是一个类别，决策区域的边界称为决策边界(decision boundary)或决策平面(decision surface)

- 对二分类问题，最简单的线性判别函数定义为输入向量的线性函数

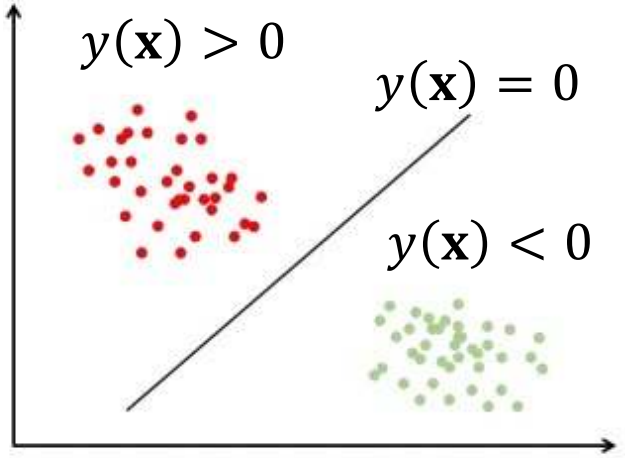
$$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$$

$\mathbf{w}$：权重　w_0：偏差

$$\mathbf{x}\text{的类别} = \begin{cases} C_1 & \text{若 } y(\mathbf{x}) \geq 0 \\ C_2 & \text{若 } y(\mathbf{x}) < 0 \end{cases}$$

决策边界

$$y(\mathbf{x}) = 0$$

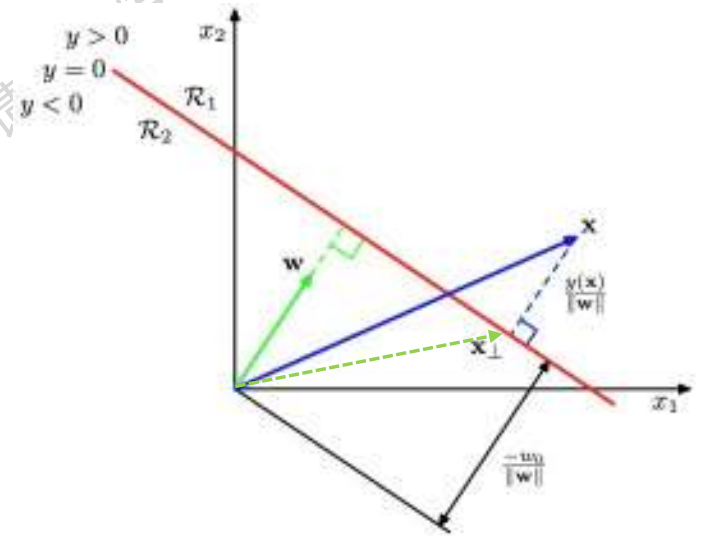


2.1.2 判别函数 (Discriminant Function)方法

决策边界性质

- $y(\mathbf{x}) = \boldsymbol{w}^{\mathrm{T}}\mathbf{x} + w_0 = 0$是决策边界，$\boldsymbol{w}$是决策平面的法向，即$\boldsymbol{w}$垂直于决策平面内的任意向量
- 任意向量$\mathbf{x}$到决策平面的有符号垂直距离r为

$$r = \frac{y(\mathbf{x})}{||\boldsymbol{w}||} = \frac{\boldsymbol{w}^{\mathrm{T}}\mathbf{x} + w_0}{||\boldsymbol{w}||}$$



$$\mathbf{x} = \mathbf{x}_\perp + r\frac{\boldsymbol{w}}{||\boldsymbol{w}||}$$

$$\boldsymbol{w}^{\mathrm{T}}\mathbf{x} + w_0 = \boldsymbol{w}^{\mathrm{T}}\mathbf{x}_\perp + w_0 + r\frac{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}}{||\boldsymbol{w}||}$$

$$y(\mathbf{x}) = 0 + r\frac{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}}{||\boldsymbol{w}||}$$

- 若$\boldsymbol{w}^{\mathrm{T}}\mathbf{x} = 0$，则$\mathbf{x}$垂直于$\boldsymbol{w}$
- $\frac{\boldsymbol{w}^{\mathrm{T}}\mathbf{x}}{||\boldsymbol{w}||}$是向量$\mathbf{x}$在向量$\boldsymbol{w}$上的投影

2.1.2 判别函数 (Discriminant Function)方法

- **多类别分类—K类判别式法，包含K个线性函数**

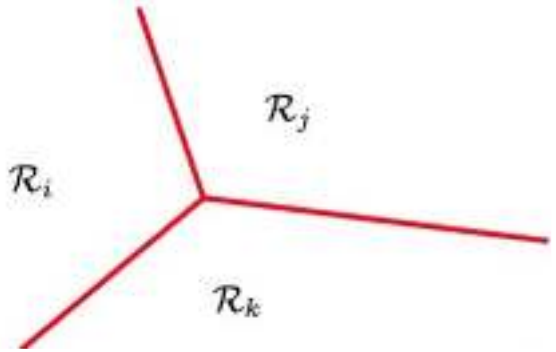
$$y_k(\mathbf{x}) = \mathbf{w}_k^{\mathrm{T}}\mathbf{x} + w_{k0}, \qquad k = 1, \ldots, K$$

$\mathbf{x}$ 的类别为C_k，如果$\forall j \neq k,\ y_k(\mathbf{x}) > y_j(\mathbf{x})$

- **C_k和C_j类之间的决策边界**

$$y_k(\mathbf{x}) = y_j(\mathbf{x})$$

$$(\mathbf{w}_k - \mathbf{w}_j)^{\mathrm{T}}\mathbf{x} + (w_{k0} - w_{j0}) = 0$$



K类判别式法

2.1.2 判别函数 (Discriminant Function)方法

判别函数学习方法1 — 最小平方和误差

- 在K类判别式方法中，每一类C_k由一个线性模型描述，$y_k(\mathbf{x}) = \mathbf{w}_k^{\mathrm{T}}\mathbf{x} + w_{k0}$，$k = 1, \ldots, K$。可以将$K$个判别式以向量式表达

$$\boldsymbol{y}(\mathbf{x}) = \widetilde{\boldsymbol{W}}^{\mathrm{T}}\tilde{\mathbf{x}}$$

$\widetilde{\boldsymbol{W}}$是一个矩阵，其第$k$列$\tilde{\boldsymbol{w}}_k = (w_{k0}, \mathbf{w}_k^{\mathrm{T}})^{\mathrm{T}}$　　$\tilde{\mathbf{x}} = (1, \mathbf{x}^{\mathrm{T}})^{\mathrm{T}}$

- 给定N个训练数据$\{\mathbf{x}_n, \boldsymbol{t}_n\}$，学习模型的参数$\widetilde{\boldsymbol{W}}$。$\boldsymbol{t}_n$是一个$K$维列向量类别标记，采取1-of-$K$编码方式，即若$\mathbf{x}_n$的类别为$C_k$，则$\boldsymbol{t}_n$的第$k$个元素为1，其他元素为0
- 定义矩阵$\boldsymbol{T}$，其第n行为$\boldsymbol{t}_n^{\mathrm{T}}$；矩阵$\widetilde{\boldsymbol{X}}$，其第$n$行为$\tilde{\mathbf{x}}_n^{\mathrm{T}}$。平方和误差函数为：

$$E_D(\widetilde{\boldsymbol{W}}) = \frac{1}{2}\mathrm{Tr}\{(\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{W}} - \boldsymbol{T})^{\mathrm{T}}(\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{W}} - \boldsymbol{T})\} \xrightarrow{\text{对}\widetilde{\boldsymbol{W}}\text{求导优化}} \widetilde{\boldsymbol{W}} = (\widetilde{\boldsymbol{X}}^{\mathrm{T}}\widetilde{\boldsymbol{X}})^{-1}\widetilde{\boldsymbol{X}}^{\mathrm{T}}\boldsymbol{T}$$

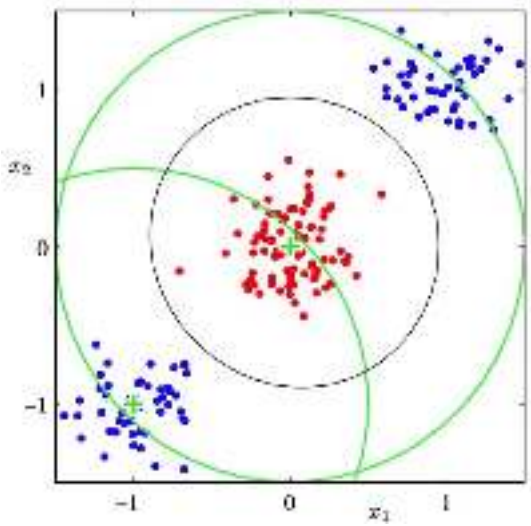
2.1.2 概率判别模型(Probabilistic Discriminative Models)

逻辑回归 (Logistic Regression)模型

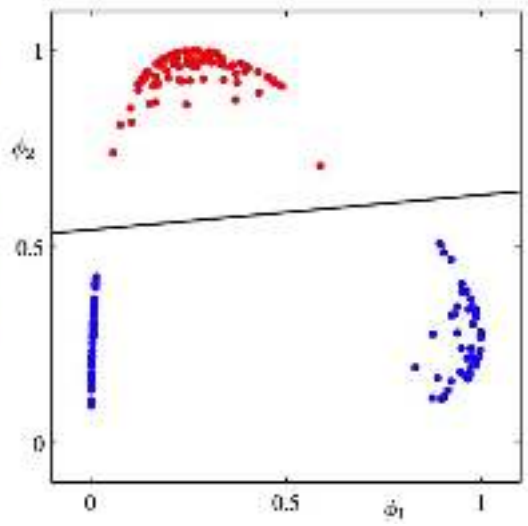
- 二分类问题

$$p(C_1|\boldsymbol{\phi}) = y(\boldsymbol{\phi}) = \sigma(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi})$$

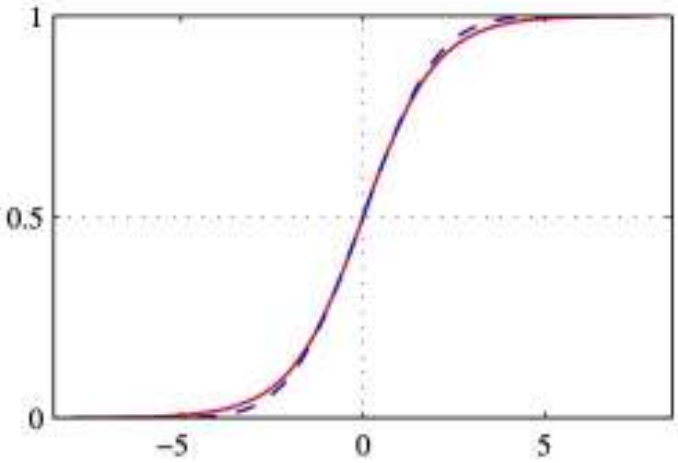
$$p(C_2|\boldsymbol{\phi}) = 1 - p(C_1|\boldsymbol{\phi})$$



原特征空间



逻辑回归



logistic sigmoid 函数 $\sigma(\mathrm{a})$

$$\sigma(\mathrm{a}) = \frac{1}{1+\exp(-a)}$$

$$\sigma(-\mathrm{a}) = 1 - \sigma(\mathrm{a})$$

$$\frac{d\sigma}{da} = \sigma(1-\sigma)$$

2.1.2 概率判别模型(Probabilistic Discriminative Models)

逻辑回归 (Logistic Regression)模型

- 最大似然估计

给定N个数据样本集 $\{(\boldsymbol{\phi}_n, t_n)|t_n \in \{0,1\}, \boldsymbol{\phi}_n = \boldsymbol{\phi}(\mathbf{x}_n), n = 1,2,\dots,N\}$，　$t_n = 1$表示C_1

似然函数为

$$p(\boldsymbol{t}|\mathbf{w}) = \prod_{n=1}^{N} y_n^{t_n}\{1-y_n\}^{1-t_n} \qquad \boldsymbol{t} = (t_1,\dots,t_N)^{\mathrm{T}} \qquad y_n = p(C_1|\boldsymbol{\phi}_n) = \sigma(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}_n)$$

负对数似然： $$E(\mathbf{w}) = -\ln p(\boldsymbol{t}|\mathbf{w}) = -\sum_{n=1}^{N}\{t_n \ln y_n + (1-t_n)\ln(1-y_n)\}$$

优化目标函数： $$\mathbf{w}^* = \arg\min\ E(\mathbf{w})$$

2.1.2 概率判别模型(Probabilistic Discriminative Models)

逻辑回归 (Logistic Regression)模型

- 迭代再加权最小平方优化算法 Iterative Reweighted Least Squares (IRLS)

牛顿-拉夫森迭代： $\mathbf{w}^{(\text{new})} = \mathbf{w}^{(\text{old})} - \mathbf{H}^{-1}\nabla E(\mathbf{w})$　　海森矩阵 $\mathbf{H} = \frac{\partial^2 E(\mathbf{w})}{\partial \mathbf{w}}$

公式计算： $\nabla E(\mathbf{w}) = \sum_{n=1}^{N}(y_n - t_n)\boldsymbol{\phi}_n = \boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{y} - \boldsymbol{t})$　　矩阵$\boldsymbol{\Phi}$，第n行为$\boldsymbol{\phi}_n^{\mathrm{T}}$

$$\mathbf{H} = \nabla\nabla E(\mathbf{w}) = \sum_{n=1}^{N} y_n(1-y_n)\boldsymbol{\phi}_n\boldsymbol{\phi}_n^{\mathrm{T}} = \boldsymbol{\Phi}^{\mathrm{T}}\mathbf{R}\boldsymbol{\Phi}$$

$\mathbf{R}$：$N\times N$对角矩阵，元素$R_{nn} = y_n(1-y_n)$

$$\mathbf{w}^{(\text{new})} = \mathbf{w}^{(\text{old})} - \left(\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{R}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{y} - \boldsymbol{t}) = \left(\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{R}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{R}\boldsymbol{z}$$

$$\boldsymbol{z} = \boldsymbol{\Phi}\mathbf{w}^{(\text{old})} - \mathbf{R}^{-1}(\boldsymbol{y} - \boldsymbol{t})$$

2.1.2 概率判别模型(Probabilistic Discriminative Models)

多类逻辑回归 (Multiclass Logistic Regression)模型

$$p(C_k|\boldsymbol{\phi}) = y_k(\boldsymbol{\phi}) = \frac{\exp(a_k)}{\sum_j \exp(a_j)} \qquad a_k = \mathbf{w}_k^{\mathrm{T}}\boldsymbol{\phi}$$

- 定义：$\boldsymbol{t}_n$为K维列向量类别标记，1-of-K编码，即若 $\boldsymbol{\phi}_n$的类别为C_k，$\boldsymbol{t}_n$的第k个元素为1，其他元素为0； $\mathbf{T}: N\times K$，第n行为$\boldsymbol{t}_n^{\mathrm{T}}$； $y_{nk} = y_k(\boldsymbol{\phi}_n)$

似然函数：

$$p(\mathbf{T}|\mathbf{w}_1, \dots, \mathbf{w}_K) = \prod_{n=1}^{N}\prod_{k=1}^{K} p(C_k|\boldsymbol{\phi})^{t_{nk}} = \prod_{n=1}^{N}\prod_{k=1}^{K} y_{nk}^{t_{nk}}$$

负对数似然：

$$E(\mathbf{w}_1, \dots, \mathbf{w}_K) = -\ln p(\mathbf{T}|\mathbf{w}_1, \dots, \mathbf{w}_K) = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk}\ln y_{nk}$$

优化目标函数：

$$\mathbf{w}^* = \arg\min E(\mathbf{w}_1, \dots, \mathbf{w}_K)$$

2.1.2 概率生成模型(Probabilistic Generative Models)

二分类问题

将输入特征向量**x**分配到2个类别中的某一类C_k $(k = 1,2)$

$$p(C_1|\mathbf{x}) = \frac{p(\mathbf{x}|C_1)p(C_1)}{p(\mathbf{x})} = \frac{p(\mathbf{x}|C_1)p(C_1)}{p(\mathbf{x}|C_1)p(C_1) + p(\mathbf{x}|C_2)p(C_2)}$$

$$= \frac{1}{1 + \exp(-a)} = \sigma(\mathrm{a}) \qquad a = \ln\frac{p(\mathbf{x}|C_1)p(C_1)}{p(\mathbf{x}|C_2)p(C_2)}$$

多分类问题

将输入特征向量**x**分配到K个类别中的某一类C_k $(k = 1, \dots, K)$

$$p(C_k|\mathbf{x}) = \frac{p(\mathbf{x}|C_k)p(C_k)}{\sum_j p(\mathbf{x}|C_j)p(C_j)} = \frac{\exp(a_k)}{\sum_j \exp(a_j)} \qquad a_k = \ln p(\mathbf{x}|C_k)p(C_k)$$

2.1.2 概率生成模型(Probabilistic Generative Models)

连续输入特征

输入为连续特征，设类条件密度服从高维高斯分布，所有类具有相同协方差矩阵，则

$$p(\mathbf{x}|C_k) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right\}$$

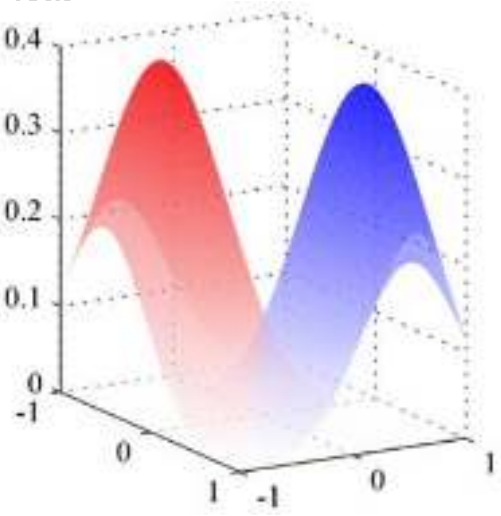
对二分类问题：

$$p(C_1|\mathbf{x}) = \sigma(\mathrm{a})$$

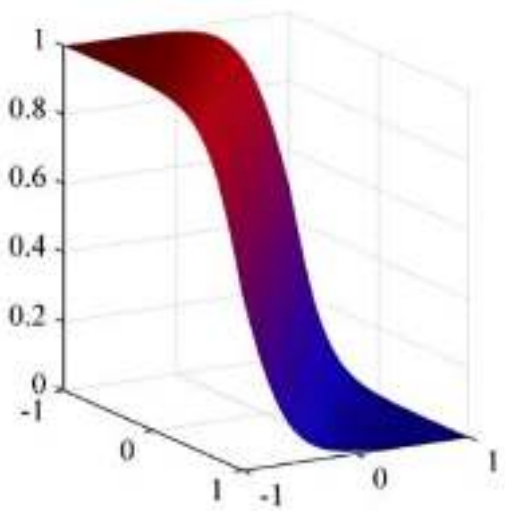
$$a = \ln\frac{p(\mathbf{x}|C_1)p(C_1)}{p(\mathbf{x}|C_2)p(C_2)} = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$$

$$\mathbf{w} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

$$w_0 = -\frac{1}{2}\boldsymbol{\mu}_1^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2 + \ln\frac{p(C_1)}{p(C_2)}$$



类条件密度



类后验概率$p(C_1|\mathbf{x})$

2.1.2 概率生成模型(Probabilistic Generative Models)

连续输入特征

输入为连续特征，设类条件密度服从高维高斯分布，所有类具有相同协方差矩阵，则

$$p(\mathbf{x}|C_k) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right\}$$

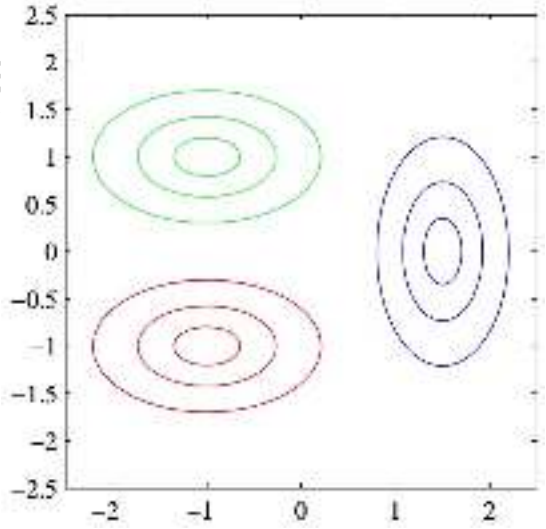
对多分类问题：

$$p(C_k|\mathbf{x}) = \frac{\exp(a_k)}{\sum_j \exp(a_j)}$$

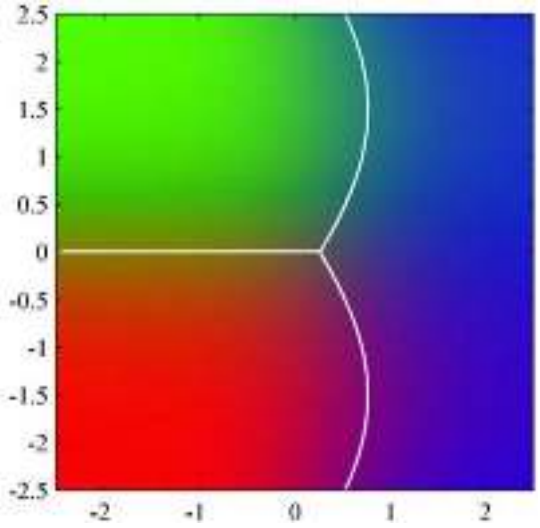
$$a_k = \ln p(\mathbf{x}|C_k)p(C_k) = \mathbf{w}_k^{\mathrm{T}}\mathbf{x} + w_{k0}$$

$$\mathbf{w}_k = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k$$

$$w_{k0} = -\frac{1}{2}\boldsymbol{\mu}_k^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \ln p(C_k)$$



类条件密度



类后验概率

2.1.2 概率生成模型(Probabilistic Generative Models)

连续输入特征

- 二分类模型最大似然估计

给定N个数据样本集 $\{(\mathbf{x}_n, t_n)|t_n \in \{0,1\}, n = 1,2,\dots,N\}$，$t_n = 1$表示$C_1$；先验概率 $p(C_1) = \pi$，$p(C_2) = 1 - \pi$，则

$$p(\mathbf{x}_n, C_1) = p(C_1)p(\mathbf{x}_n|C_1) = \pi\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_1, \boldsymbol{\Sigma})$$

$$p(\mathbf{x}_n, C_2) = p(C_2)p(\mathbf{x}_n|C_2) = (1-\pi)\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_2, \boldsymbol{\Sigma})$$

似然函数

$$p(\boldsymbol{t}|\pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}) = \prod_{n=1}^{N} [\pi\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_1, \boldsymbol{\Sigma})]^{t_n}[(1-\pi)\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_2, \boldsymbol{\Sigma})]^{1-t_n}$$

CONTENTS

- 线性模型
- 支撑向量机
- 核方法
- 应用例子

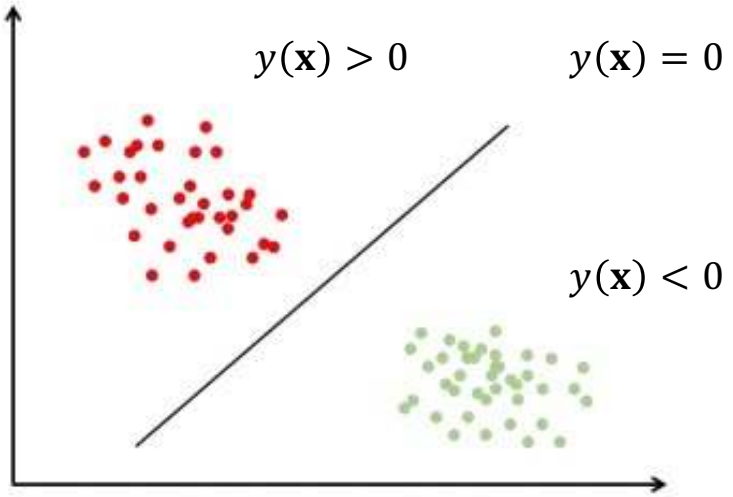
线性分类器

二分类

模型：$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$

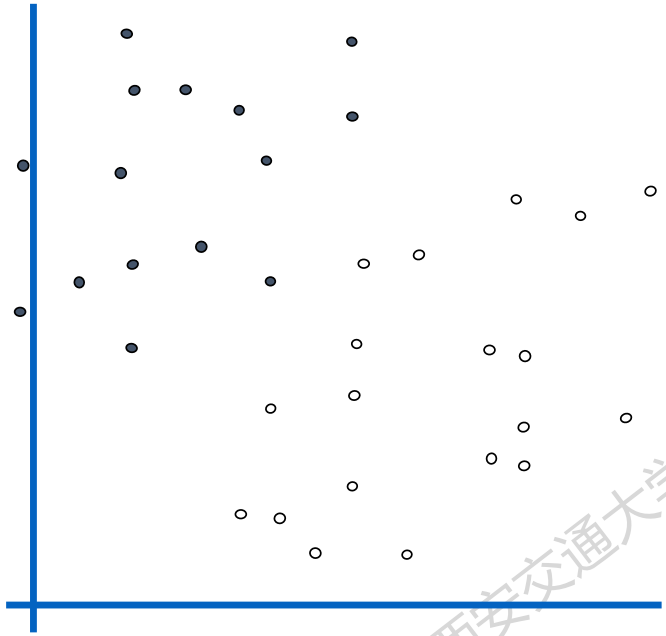
$\boldsymbol{w}$：权重向量　w_0：偏差向量　$\mathbf{x} = (x_1, x_2, \ldots x_d)$：输入向量

$$\mathbf{x}\text{的类别}= \begin{cases} 1 & \text{如果 } y(\mathbf{x}) \geq 0 \\ -1 & \text{如果 } y(\mathbf{x}) < 0 \end{cases}$$



线性分类

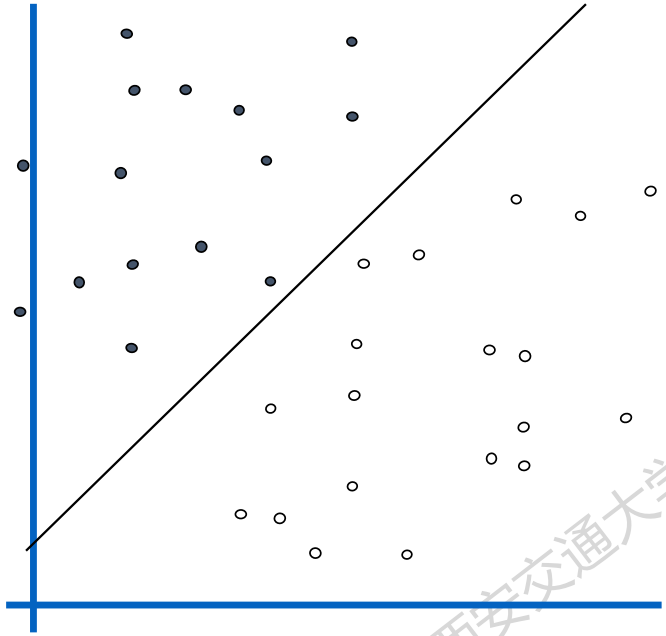
- 代表+1
- 代表-1



如何对这些数据分类?

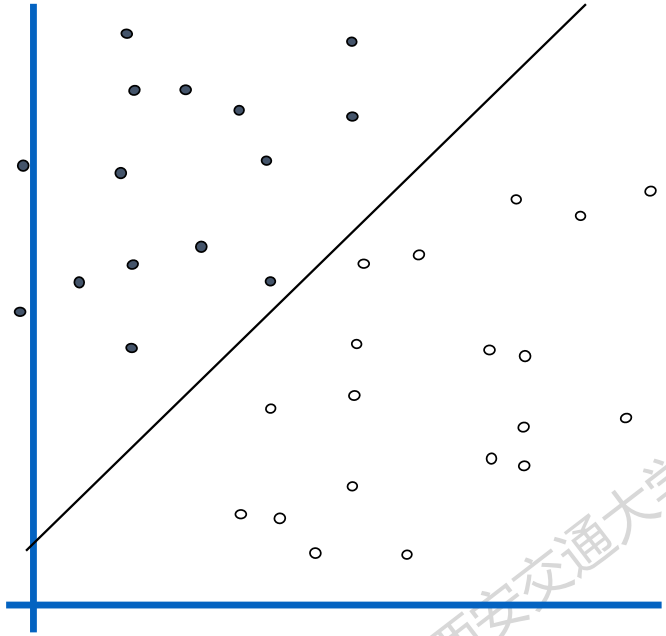
线性分类

- 代表+1
- 代表-1



线性分类

- 代表+1
- 代表-1



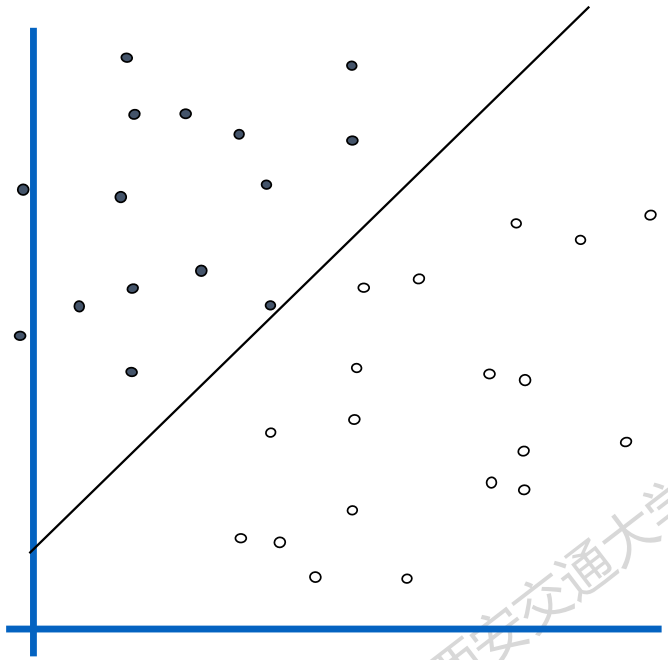
线性模型：$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$

$$\mathbf{x}\text{的类别}=\begin{cases}1 & \text{如果 } y(\mathbf{x}) \geq 0\\ -1 & \text{如果 } y(\mathbf{x}) < 0\end{cases}$$

分界边界：$\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0 = 0$

线性分类

- 代表+1
- 代表-1



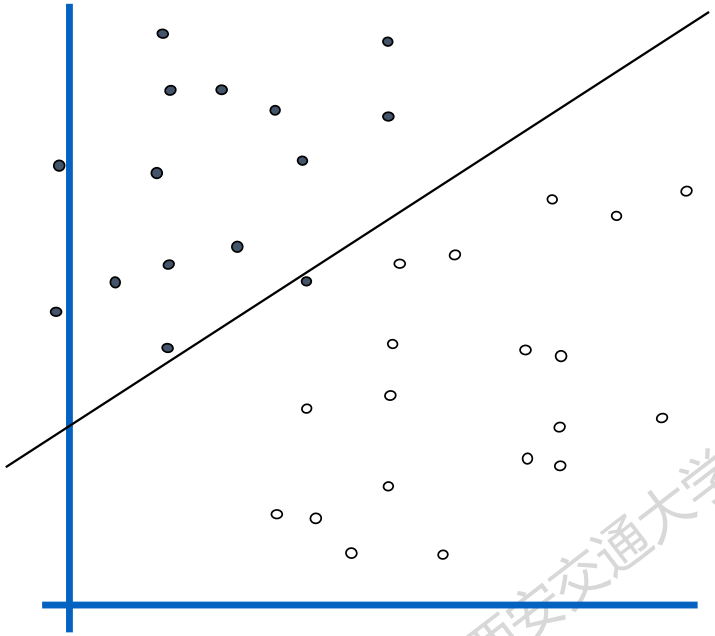
线性模型：$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$

$$\mathbf{x}\text{的类别}=\begin{cases} 1 & \text{如果 } y(\mathbf{x}) \geq 0 \\ -1 & \text{如果 } y(\mathbf{x}) < 0 \end{cases}$$

分界边界：$\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0 = 0$

线性分类

- 代表+1
- 代表-1



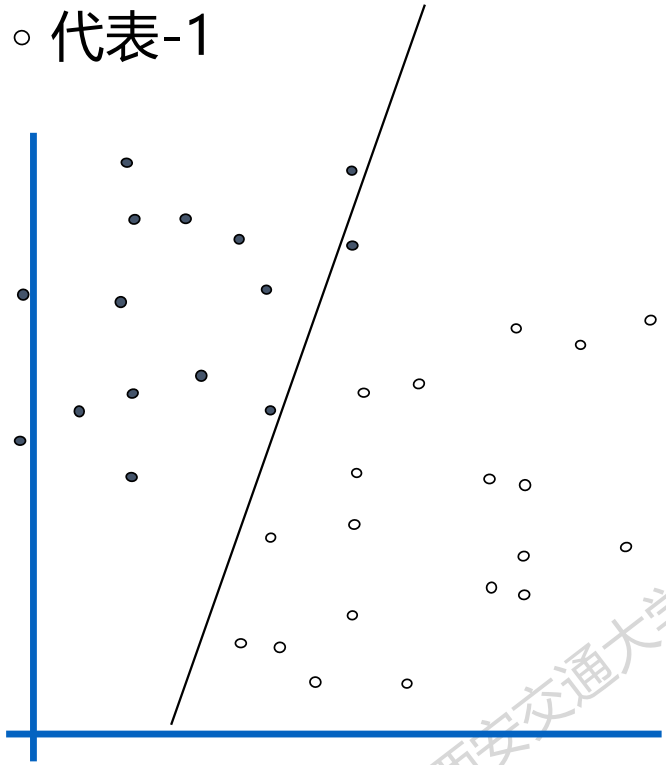
线性模型：$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$

$$\mathbf{x}\text{的类别}=\begin{cases}1 & \text{如果 } y(\mathbf{x}) \geq 0\\ -1 & \text{如果 } y(\mathbf{x}) < 0\end{cases}$$

分界边界：$\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0 = 0$

线性分类

- 代表+1
- 代表-1



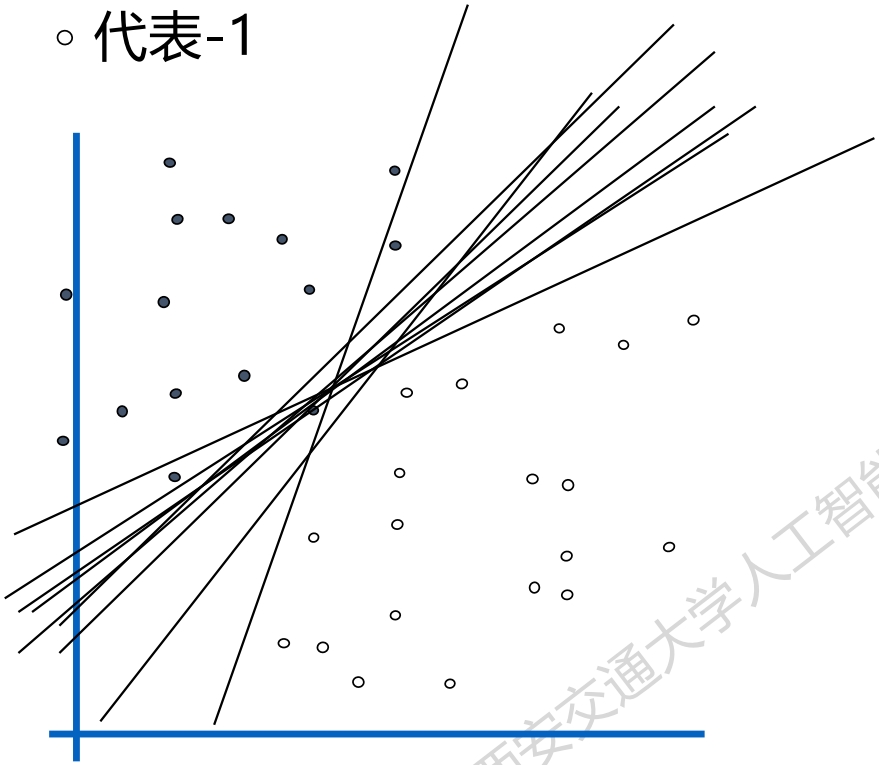
线性模型：$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$

$$\mathbf{x}\text{的类别}=\begin{cases} 1 & \text{如果 } y(\mathbf{x}) \geq 0 \\ -1 & \text{如果 } y(\mathbf{x}) < 0 \end{cases}$$

分界边界：$\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0 = 0$

线性分类

- 代表+1
- 代表-1



线性模型：$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$

$$\mathbf{x}\text{的类别}=\begin{cases}1 & \text{如果 } y(\mathbf{x}) \geq 0\\ -1 & \text{如果 } y(\mathbf{x}) < 0\end{cases}$$

分界边界：$\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0 = 0$

这些边界都可以将数据进行分类，但哪个最好?

最大间隔分类器

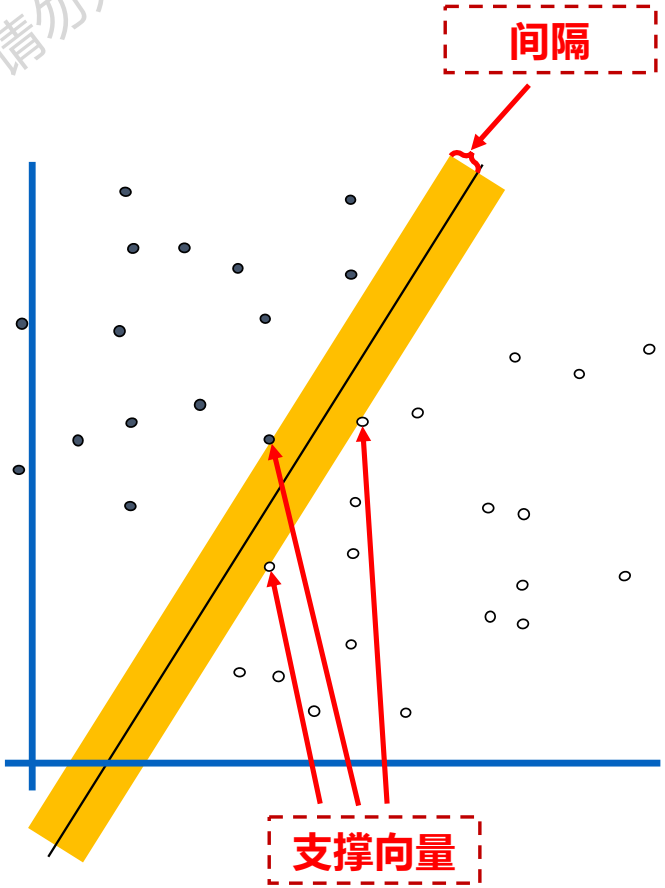
- **二分类器：** $y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}) + b$，$y(\mathbf{x}) > 0$，类为+1；$y(\mathbf{x}) < 0$，类为−1

- 任务：给定含有N个样本的训练集 $\{(\mathbf{x}_n, t_n) | \, t_n \in \{+1, -1\}, \; n = 1,2,\ldots,N\}$，学习模型参数

- 支撑向量机（Support Vector Machine, SVM）学习的决策边界是使训练样本的**间隔**最大，即决策边界是**最大间隔超平面**（max-margin hyperplane）

- **间隔：** 决策边界和任意样本点之间的最小距离

- 间隔的位置由样本点的子集确定，这些样本点称为**支撑向量 (support vectors)**



最大间隔分类器

- **二分类器：** $y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}) + b$，$y(\mathbf{x}) > 0$，类为+1；$y(\mathbf{x}) < 0$，类为−1

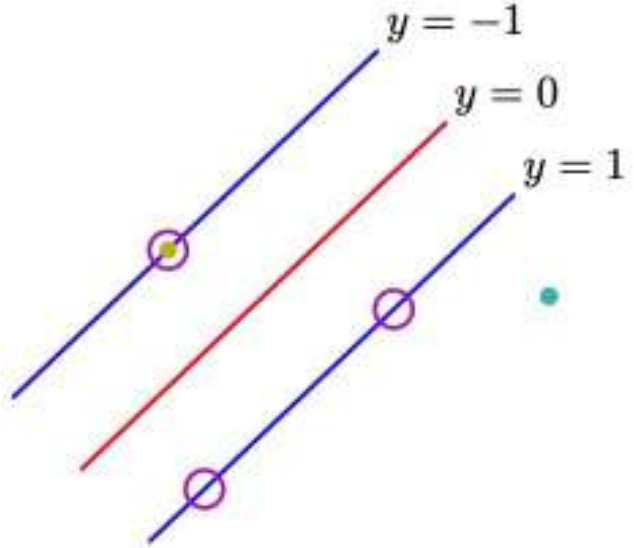
- 点$\mathbf{x}_n$到决策平面的距离为$\frac{|y(\mathbf{x}_n)|}{||\boldsymbol{w}||} = \frac{t_n y(\mathbf{x}_n)}{||\boldsymbol{w}||} = \frac{t_n(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)+b)}{||\boldsymbol{w}||}$，则最大间隔优化问题为

$$\arg\max_{\mathbf{w},\, b} \left\{ \min_n \left[\frac{t_n(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n) + b)}{||\boldsymbol{w}||} \right] \right\} \quad \Rightarrow \quad \arg\max_{\mathbf{w},\, b} \left\{ \frac{1}{||\boldsymbol{w}||} \min_n [t_n(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n) + b)] \right\}$$

- $\mathbf{w} \to \alpha\mathbf{w}$，$b \to \alpha b$，任一点$\mathbf{x}_n$到决策平面的距离$\frac{t_n(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)+b)}{||\boldsymbol{w}||}$不变，则可以使$\min_n [t_n(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n) + b)] = 1$

- 则对所有点都应满足

$$t_n(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n) + b) \geq 1, \quad n = 1, \ldots, N$$



最大间隔分类器

- **二分类器：** $y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}) + b$，$y(\mathbf{x}) > 0$，类为+1；$y(\mathbf{x}) < 0$，类为−1

- 上述最大间隔优化问题转化为

$$\arg\max_{\mathbf{w},\,b} \left\{\frac{1}{||\boldsymbol{w}||}\right\}$$

满足条件$t_n(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n) + b) \geq 1$，$n = 1, \ldots, N$

即：

$$\arg\min_{\mathbf{w},\,b} \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

$$\text{s.t.} \quad \forall n = 1, \ldots, N,$$

$$t_n(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n) + b) \geq 1$$

① 最大化间隔

② 对所有训练样本正确分类

求解SVM

- 拉格朗日方程

$$L(\mathbf{w}, b, \boldsymbol{a}) = \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} - \sum_{n=1}^{N} a_n\{t_n(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n) + \mathrm{b}) - 1\}$$

$$\boldsymbol{a} = (a_1, \ldots, a_N)^{\mathrm{T}}, \quad a_n \geq 0$$

对**w**和 b求导得到：

$$\mathbf{w} = \sum_{n=1}^{N} a_n t_n \boldsymbol{\phi}(\mathbf{x}_n)$$

$$0 = \sum_{n=1}^{N} a_n t_n$$

$$\arg\min_{\mathbf{w}, b} \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

$$\text{s.t.} \quad \forall n = 1, \ldots, N,$$

$$t_n(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n) + b) \geq 1$$

SVM的对偶问题

将上述等式代入拉格朗日方程得到：

对偶问题

$$\tilde{L}(\boldsymbol{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m k(\boldsymbol{x}_n, \boldsymbol{x}_m)$$

$$k(\boldsymbol{x}_n, \boldsymbol{x}_m) = \boldsymbol{\phi}(\mathbf{x}_n)^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_m)$$

满足条件 $a_n \geq 0, n = 1, \ldots N$

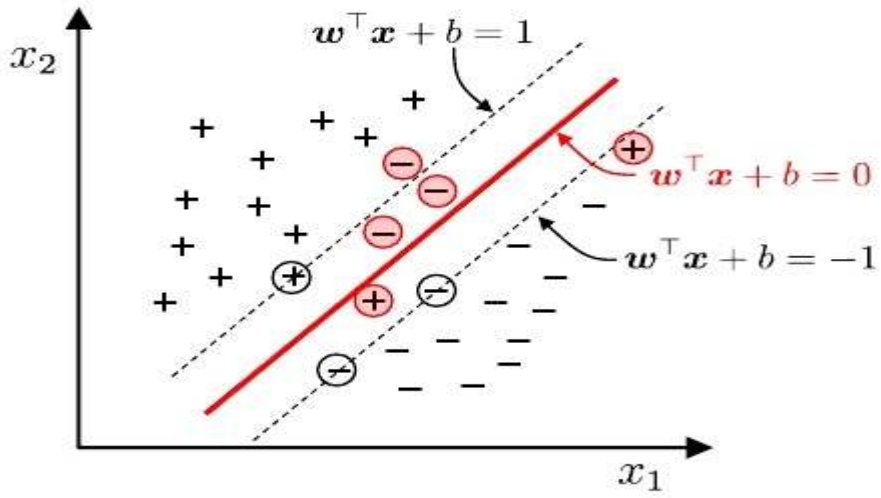
$$\sum_{n=1}^{N} a_n t_n = 0$$

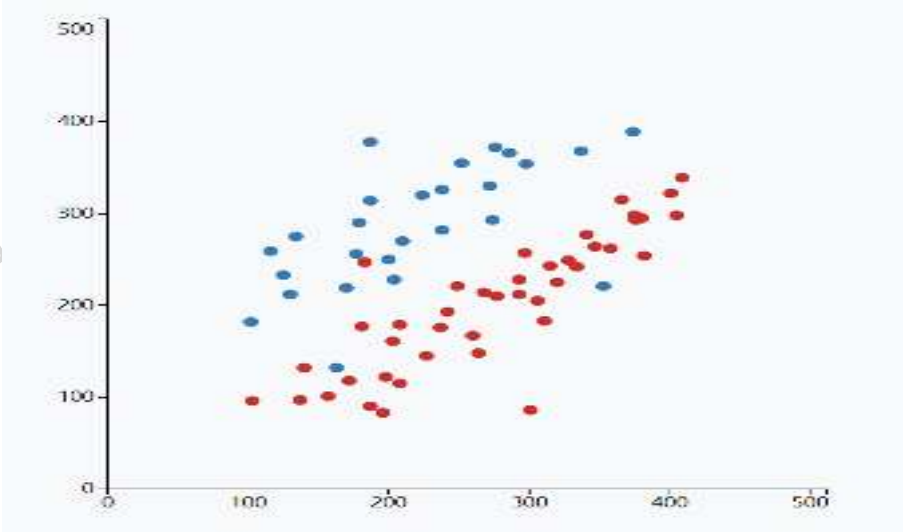
分类器形式

$$y(\mathbf{x}) = \sum_{n=1}^{N} a_n t_n k(\boldsymbol{x}, \boldsymbol{x_n}) + b$$

引入松弛变量的SVM

- 实际应用中样本一般都不是线性可分的，不同类的样本之间交错齿合





- **解决该问题的两个思路**

- 使用松弛变量和惩罚因子，允许错分样本的存在
- 使用核函数，引入非线性变换

引入松弛变量的SVM

- 使用松弛变量与惩罚因子，引入软间隔 (soft margin)，允许在一些样本上不满足约束，但对违反约束的样本进行惩罚

$$\arg\min_{\mathbf{w}} \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + \mathrm{C}\sum_{i=1}^{N}\xi_i$$

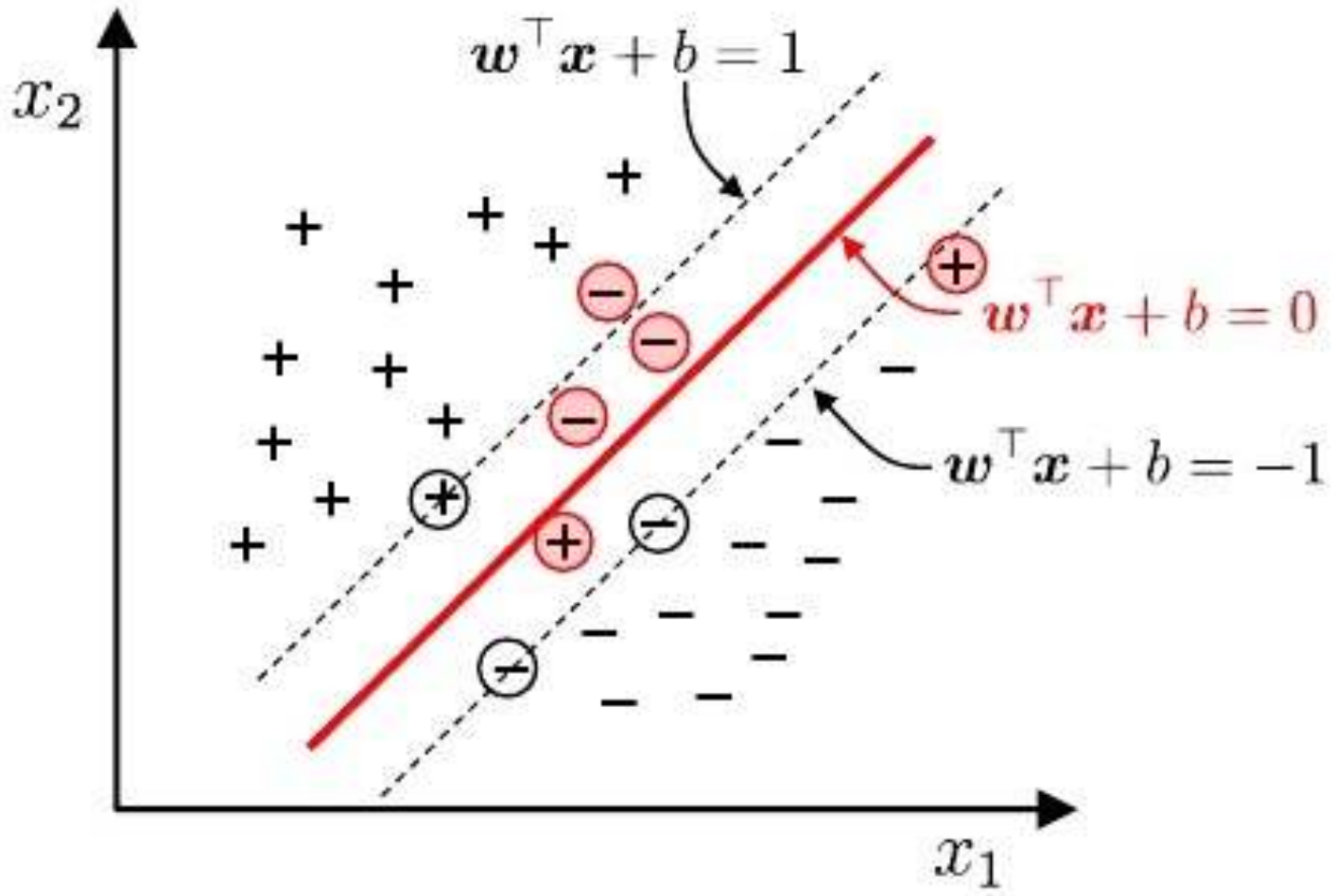
$$s.t. \quad \forall\, i = 1, \ldots N,$$

$$t_i y(\boldsymbol{x}_i) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

- ξ_i是松弛变量，如果它不为0，表示样本违反了不等式约束条
- 惩罚因子C是人工设定的大于0的参数，用来对违反不等式约束条件的样本进行惩罚

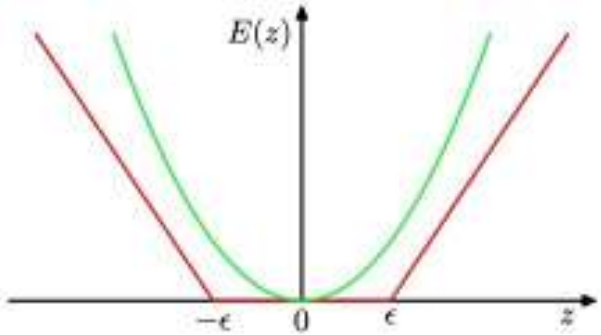
引入松弛变量的SVM



SVM用于回归

- 线性回归问题$y(\mathbf{x}) = \boldsymbol{w}^{\mathrm{T}}\mathbf{x} + b$中，优化正则化误差函数：

$$\frac{1}{2}\sum_{n=1}^{N}\{y_n - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$



- ϵ-*insensitive* 误差函数, $\epsilon > 0$

$$E_\epsilon(y(\mathbf{x}) - t) = \begin{cases} 0 & |y(\mathbf{x}) - t| < \epsilon \\ |y(\mathbf{x}) - t| - \epsilon, & \text{otherwise} \end{cases}$$

- 线性回归问题优化函数引入ϵ-insensitive函数

$$C\sum_{n=1}^{N} E_\epsilon(y(\mathbf{x}) - t) + \frac{1}{2}\|\mathbf{w}\|^2$$

SVM用于回归

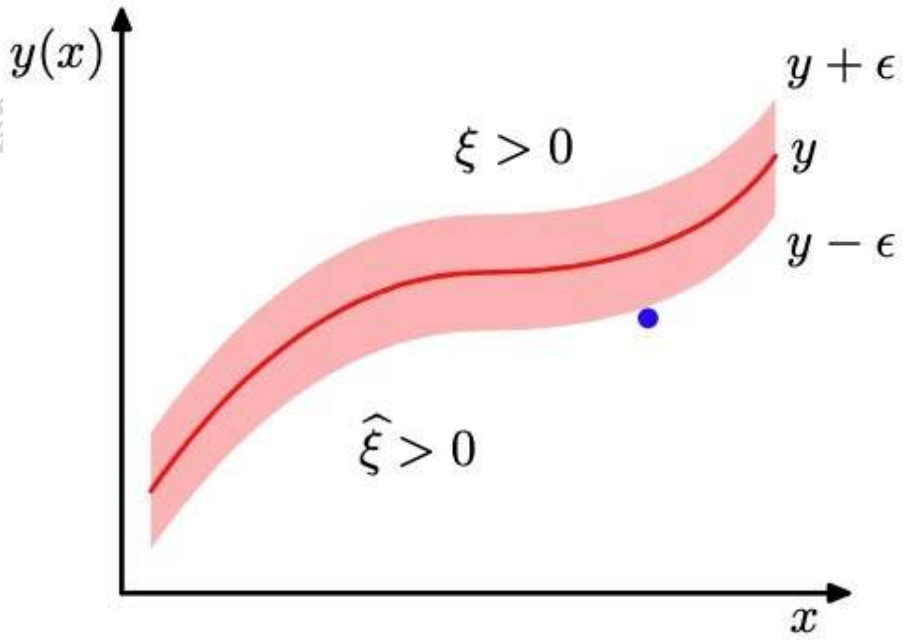
$$\arg\min_{\mathbf{w},\ \xi_n,\hat{\xi}_n} \ C\sum_{n=1}^{N}(\xi_n+\hat{\xi}_n)+\frac{1}{2}\|\mathbf{w}\|^2$$

$$s.t. \quad \forall\, i=1,\dots N,$$

$$t_n \le y(\mathbf{x}_n)+\epsilon+\xi_n$$

$$t_n \ge y(\mathbf{x}_n)-\epsilon-\hat{\xi}_n$$

$$\xi_n \ge 0,\ \hat{\xi}_n \ge 0$$



Structural SVM

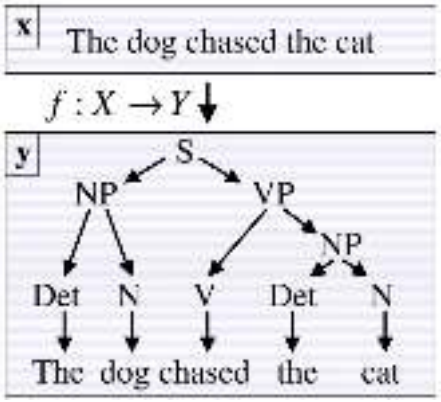
一般分类问题：

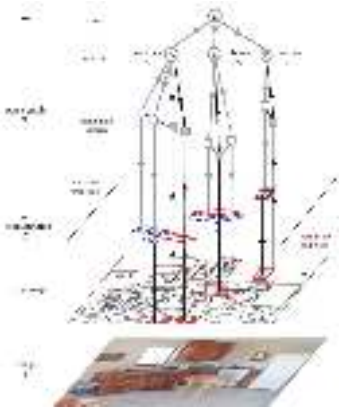
给定：带有离散类别标记的训练数据$(\mathbf{x}_1, \mathbf{y}_1), \ldots (\mathbf{x}_n, \mathbf{y}_n) \in \mathcal{X} \times \mathcal{Y}$

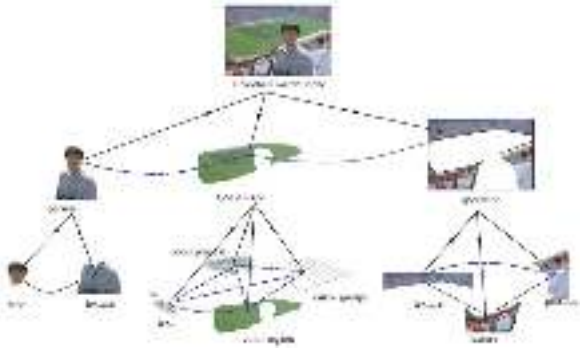
任务：学习从输入$\mathbf{x} \in \mathcal{X}$到输出$\mathbf{y} \in \mathcal{Y}$的映射$f: \mathcal{X} \rightarrow \mathcal{Y}$

结构化输出问题：

输出为结构的问题，如序列，字符串，树，图







结构化、非结构化

数据科学	特征	例子
结构化 structured	高度组织和整齐格式化的数据，可由表结构来逻辑表达和实现的数据	数字、符号
非结构化数据 non-structured	结构化数据之外的一切数据，结构不规则，没有预定义的数据模型，难以使用表格逻辑表来表现的数据	邮件、媒体、网站、应用

机器学习	特征	例子
变量数据 variable	数据由单变量或向量表达，向量的元素之间没有直接联系	类别、
结构化数据 structured /structural	数据由多种属性组成、属性之间具有联系，属性和关系共同构成某种复杂结构	树、图、序列、字符串

Structural SVM

- 将映射$f: \mathcal{X} \to \mathcal{Y}$转化为一个结构化判别函数，

$$F: \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$$

$$f(\mathbf{x}; \mathbf{w}) = \arg\max_{\mathbf{y} \in \mathcal{Y}} F(\mathbf{x}, \mathbf{y}; \mathbf{w})$$

F可以看作一个衡量$(\mathbf{x}, \mathbf{y})$相容性的函数

- F可表示为输入和输出的组合特征表达：

$$F(\mathbf{x}, \mathbf{y}; \mathbf{w}) = < \mathbf{w}, \Psi(\mathbf{x}, \mathbf{y}) >$$

Structural SVM

- 定义 $\delta\Psi_i(\mathbf{y}) = \Psi(\mathbf{x}_i, \mathbf{y}_i) - \Psi(\mathbf{x}_i, \mathbf{y})$

  - SVM_0:

$$\arg\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2$$

$$s.t. \quad \forall\, i = 1, \ldots n, \ \forall \mathbf{y} \in \mathcal{Y} \backslash \mathbf{y}_i$$

$$< \mathbf{w}, \delta\Psi_i(\mathbf{y}) \geq 1$$

Structural SVM

- Soft-Margin Maximization

  - SVM_1:

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

$$s.t. \quad \forall\, i = 1, \dots n, \ \forall \mathbf{y} \in \mathcal{Y}\backslash\mathbf{y}_i$$

$$< \mathbf{w}, \delta\Psi_i(\mathbf{y}) \geq 1 - \xi_i, \xi_i \geq 0$$

Structural SVM

- Slack Rescaling
  - $\mathrm{SVM}_1^{\Delta s}$:

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

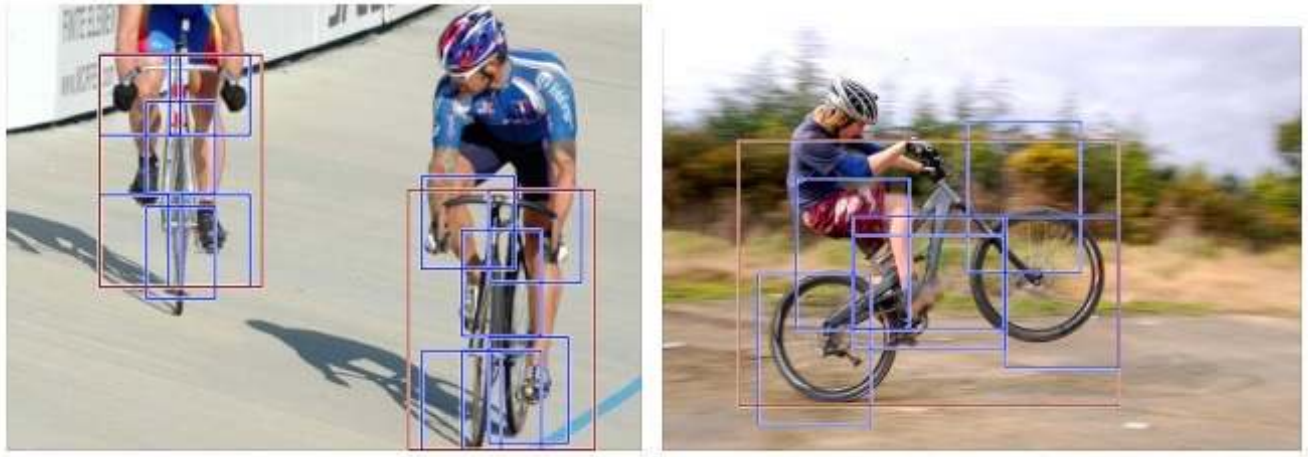
$$s.t. \quad \forall\, i = 1, \ldots n, \ \forall \mathbf{y} \in \mathcal{Y}\backslash\mathbf{y}_i$$

$$< \mathbf{w}, \delta\Psi_i(\mathbf{y}) \geq 1 - \frac{\xi_i}{\Delta(\mathbf{y}_i, \mathbf{y})}$$

Latent SVM (LSVM)

- 在许多问题中，输入-输出关系不能用简单的$(\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{Y}$刻画，还依赖于一些无法观测到的变量即$\mathbf{h} \in \mathcal{H}$,因此，预测函数定义为

$$f(\mathbf{x}; \mathbf{w}) = \arg\max_{(\mathbf{y},\mathbf{h}) \in \mathcal{Y} \times \mathcal{H}} =< \mathbf{w}, \Psi(\mathbf{x}, \mathbf{y}, \mathbf{h}) >$$



Deformable Part Models [Felzenszwalb, PAMI 2010]

Latent SVM (LSVM)

- Latent Structural SVM [Yu & Joachims, ICML09]

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

$$s.t. \quad \forall\, i = 1,\ldots n,\ \ \forall \mathbf{y} \in \mathcal{Y}$$

$$\max_{\mathbf{h}\in\mathcal{H}} < \mathbf{w}, \Psi(\mathbf{x}_i, \mathbf{y}_i, \mathbf{h}) - \max_{\hat{\mathbf{h}}\in\mathcal{H}} < \mathbf{w}, \Psi(\mathbf{x}_i, \mathbf{y}, \mathbf{h}) >$$

$$\geq \Delta(\mathbf{y}_i, \mathbf{y}, \hat{\mathbf{h}}) - \xi_i$$

CONTENTS

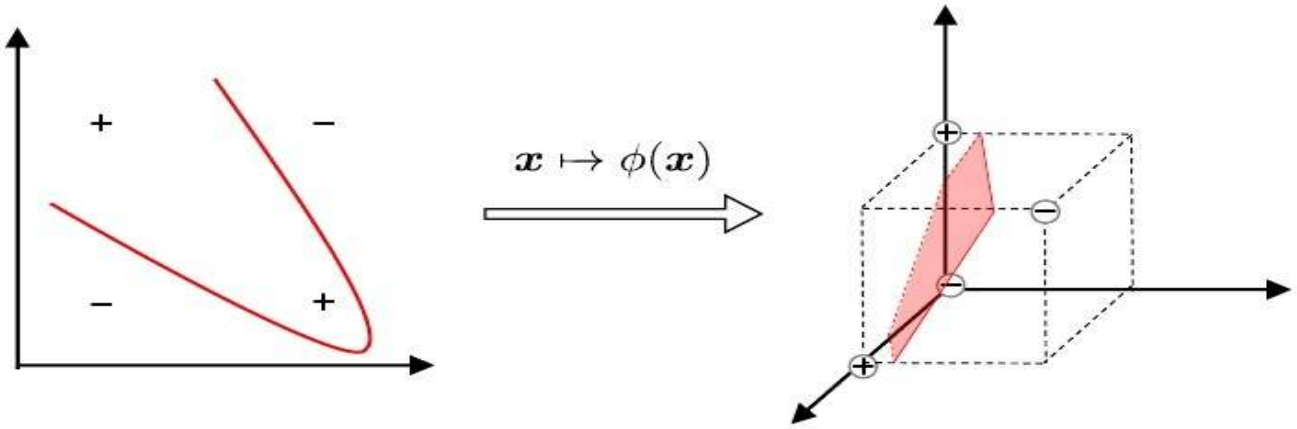
- 线性模型
- 支撑向量机
- 核方法
- 应用例子

核(kernel)方法的引入

- 对偶问题中：

$$\tilde{L}(\boldsymbol{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m \boldsymbol{\phi}(\mathbf{x}_n)^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_m)$$

- 若不存在一个能正确划分两类样本的超平面，将样本从原始空间映射到一个更高维的特征空间，使样本在这个特征空间内线性可分



核(kernel)

- 对于特征空间映射$\phi: R^M \to R^N$，核函数定义为：

$$k(\boldsymbol{x}, \boldsymbol{y}) = \phi(\boldsymbol{x})^{\mathrm{T}} \phi(\boldsymbol{y})$$

- 对于特征空间映射ϕ，如果$k(\cdot,\cdot)$能被表示为一个内积，则它是一个核：

$$k(\boldsymbol{x}, \boldsymbol{y}) = \phi(\boldsymbol{x})^{\mathrm{T}} \phi(\boldsymbol{y})$$

- 核函数是对称函数 $k(\boldsymbol{x}, \boldsymbol{y}) = k(\boldsymbol{y}, \boldsymbol{x})$

- N可以是非常大甚至无穷， $\phi: R^M \to R^N$可以是隐式的，非显式的

核的例子

- $\boldsymbol{x}$和$\boldsymbol{z}$是二维向量，$\boldsymbol{x} = (x_1, x_2)^{\mathrm{T}}, \boldsymbol{z} = (z_1, z_2)^{\mathrm{T}}$
- $k(\boldsymbol{x}, \boldsymbol{z}) = (\boldsymbol{x}^{\mathrm{T}}\boldsymbol{z})^2$是一个核函数，特征空间映射为

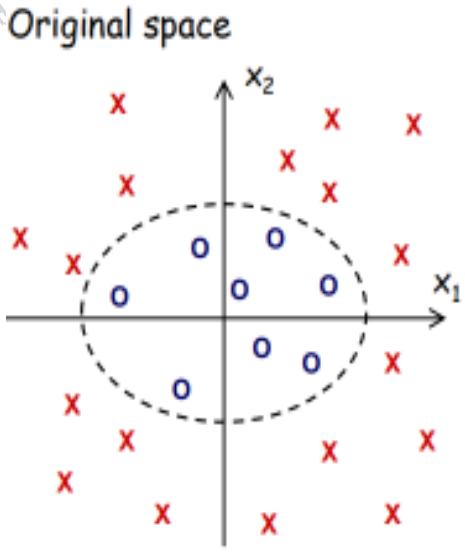
$$\phi: R^2 \to R^3 \qquad \phi(\boldsymbol{x}) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$$

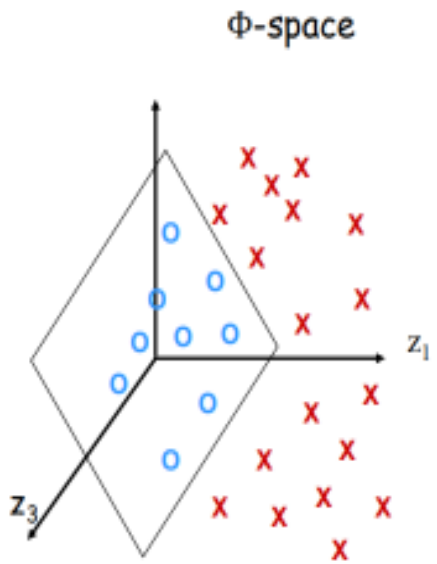
- $k(\boldsymbol{x}, \boldsymbol{z}) = (\boldsymbol{x}^{\mathrm{T}}\boldsymbol{z})^2$

$$= (x_1z_1 + x_2z_2)^2$$

$$= (x_1^2, x_2^2, \sqrt{2}x_1x_2) \cdot (z_1^2, z_2^2, \sqrt{2}z_1z_2)$$

$$= \phi(\boldsymbol{x})^{\mathrm{T}}\phi(\boldsymbol{z})$$





常见的核函数

- **多项式核 (Polynomial Kernel)**

$$k(\boldsymbol{x},\boldsymbol{y}) = \left(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y} + c\right)^{M}, \quad \mathrm{c} > 0, \quad M\text{为整数}$$

- **高斯核(Gaussian Kernel)：** $k(\boldsymbol{x},\boldsymbol{y}) = \exp\left[-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{2\sigma^2}\right]$

- **拉普拉斯核 (Laplace Kernel):** $k(\boldsymbol{x},\boldsymbol{y}) = \exp\left[-\frac{\|\mathbf{x}-\boldsymbol{y}\|}{\sigma}\right]$

- **反曲核(Sigmoidal Kernel)：** $k(\boldsymbol{x},\boldsymbol{y}) = \tanh\left(a\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y} + b\right)$

构建核

- **直接构造核函数：**确保选择的函数是一个有效的核函数，即它对应于某个(可能是无限维)特征空间中的内积
- **基于已有核构造新核：**给定已有核$k_1(\boldsymbol{x},\boldsymbol{y})$，$k_2(\boldsymbol{x},\boldsymbol{y})$，以下函数也是有效核：
  - $k(\boldsymbol{x},\boldsymbol{y}) = ck_1(\boldsymbol{x},\boldsymbol{y})$
  - $k(\boldsymbol{x},\boldsymbol{y}) = f(\boldsymbol{x})k_1(\boldsymbol{x},\boldsymbol{y})f(\boldsymbol{y})$
  - $k(\boldsymbol{x},\boldsymbol{y}) = q\big(k_1(\boldsymbol{x},\boldsymbol{y})\big)$
  - $k(\boldsymbol{x},\boldsymbol{y}) = \exp\big(k_1(\boldsymbol{x},\boldsymbol{y})\big)$
  - $k(\boldsymbol{x},\boldsymbol{y}) = k_1(\boldsymbol{x},\boldsymbol{y}) + k_2(\boldsymbol{x},\boldsymbol{y})$
  - $k(\boldsymbol{x},\boldsymbol{y}) = k_1(\boldsymbol{x},\boldsymbol{y})k_2(\boldsymbol{x},\boldsymbol{y})$
  - $k(\boldsymbol{x},\boldsymbol{y}) = k_3\big(\boldsymbol{\phi}(\boldsymbol{x}),\boldsymbol{\phi}(\boldsymbol{y})\big)$
  - $k(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{y}$
  - $k(\boldsymbol{x},\boldsymbol{y}) = k_a(\boldsymbol{x_a},\boldsymbol{y_a}) + k_b(\boldsymbol{x_b},\boldsymbol{y}_b)$
  - $k(\boldsymbol{x},\boldsymbol{y}) = k_a(\boldsymbol{x_a},\boldsymbol{y_a})k_b(\boldsymbol{x_b},\boldsymbol{y}_b)$

其中，c > 0是一个常数，f(·)是任意函数，q(·)是一个非负系数多项式，$\boldsymbol{\phi}(\boldsymbol{x})$是一个从$\boldsymbol{x}$到$R^M$的函数，$k_3(\cdot,\cdot)$是一个$R^M$上有效的核函数，$\boldsymbol{A}$是对称的正半正定矩阵，$\boldsymbol{x_a}$和$\boldsymbol{x_b}$是$\boldsymbol{x}=(\boldsymbol{x_a},\boldsymbol{x_b})$的变量，$k_a$和$k_b$是各自空间上的有效核函数

核方法—Kernel substitution

- 核的概念被表述为特征空间中的内积，这允许我们通过使用核替换(kernel substitution)来构建许多著名算法的扩展
- 基本思想：如果一个算法中输入向量$\boldsymbol{x}$以内积的形式出现，则可以将此输入向量的内积替换为其他核(kernel)，即将$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}$替换为$k(\boldsymbol{x},\boldsymbol{y})$
- 核方法提供强大的模块化，不需要改变底层的学习算法来适应核函数的特定选择；此外，我们可以替换不同的算法，同时保持相同的核
- 分类：Perceptron, SVM, KNN
- 回归：linear, ridge regression
- 聚类：k-means

CONTENTS

- 线性模型
- 支撑向量机
- 核方法
- 应用例子

应用

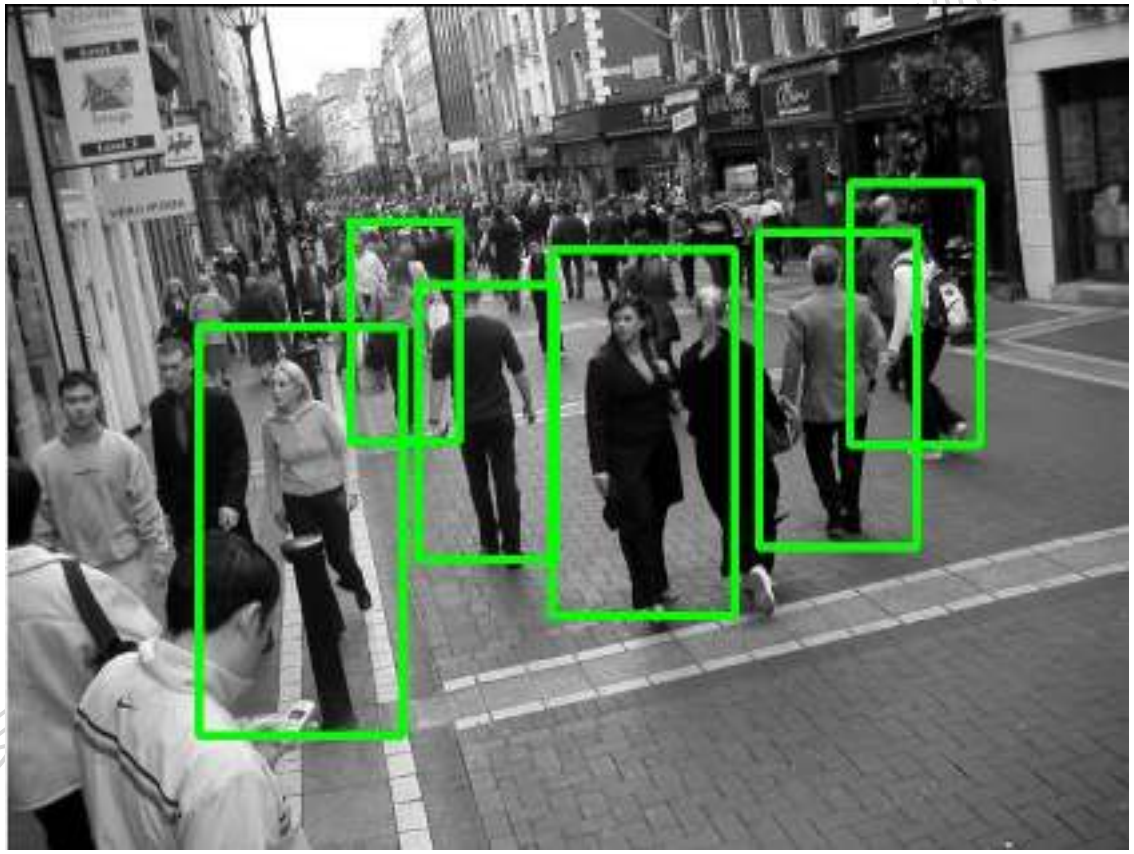
- 例1:行人检测
- 例2:并发行为检测及其结构化预测模型
- 例3:目标检测的隐结构模型
- 例4:行为识别的组合隐结构模型

行人检测

- N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005

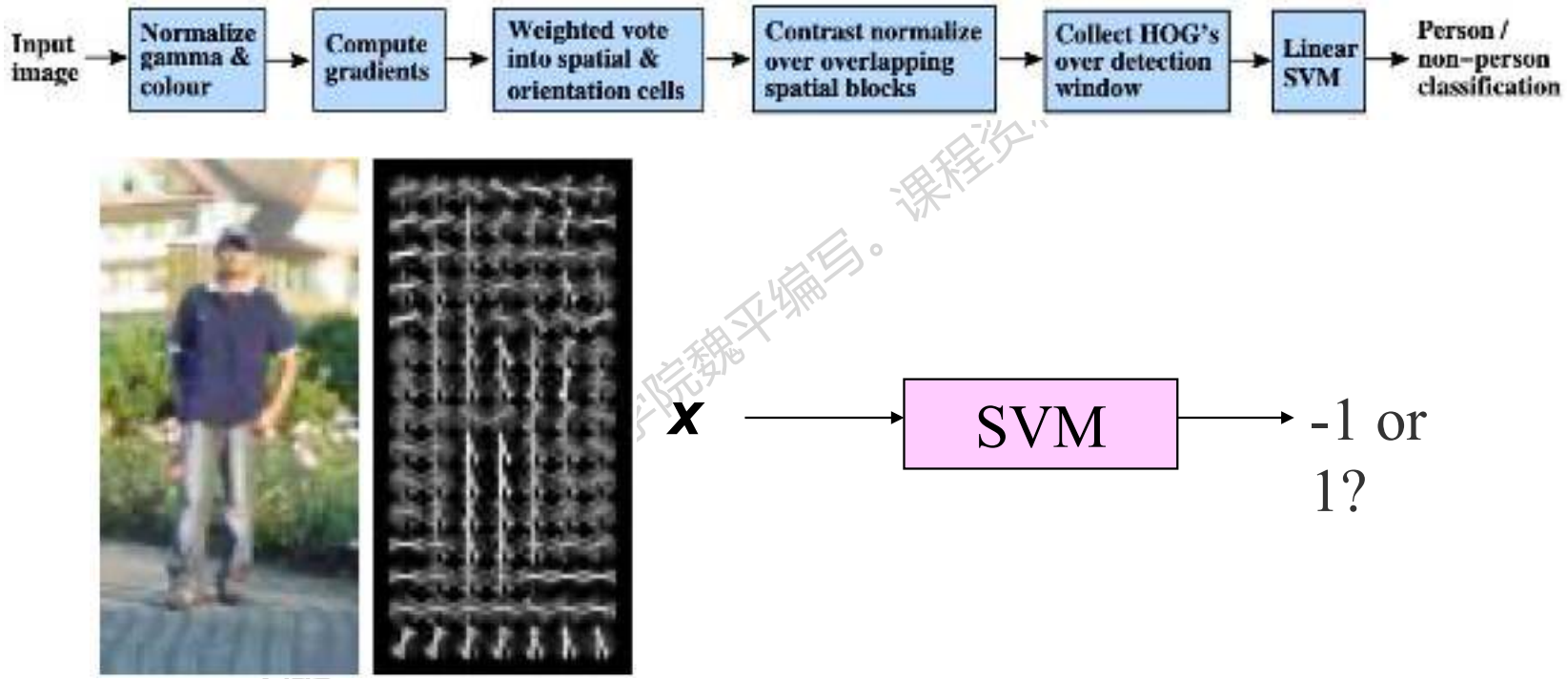
应用例子

- 行人检测



应用例子

- 行人检测：HOG+SVM



N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005

并发行为检测及其结构化预测模型

- Ping Wei, Nanning Zheng, Yibiao Zhao, Song-Chun Zhu. Concurrent Action Detection with Structural Prediction. *Proceedings of the IEEE International Conference on Computer Vision* (ICCV), pp. 3136-3143, 2013.

问题提出

- 行为识别是判断视频或图像中人的行为的类别

K. Guo, et al, Boston University

  - **以往工作将行为理解建模为一个分类(classification)问题**

提出一个正确的问题往往比解决问题更重要

——爱因斯坦

问题提出

- 分类问题的基本假设是分类输出只有一个结果，且候选类别应在同一个分类空间中，即候选类别具有可比性。

Q1: ✗

Is it a cat or brown?

Q2: ✗

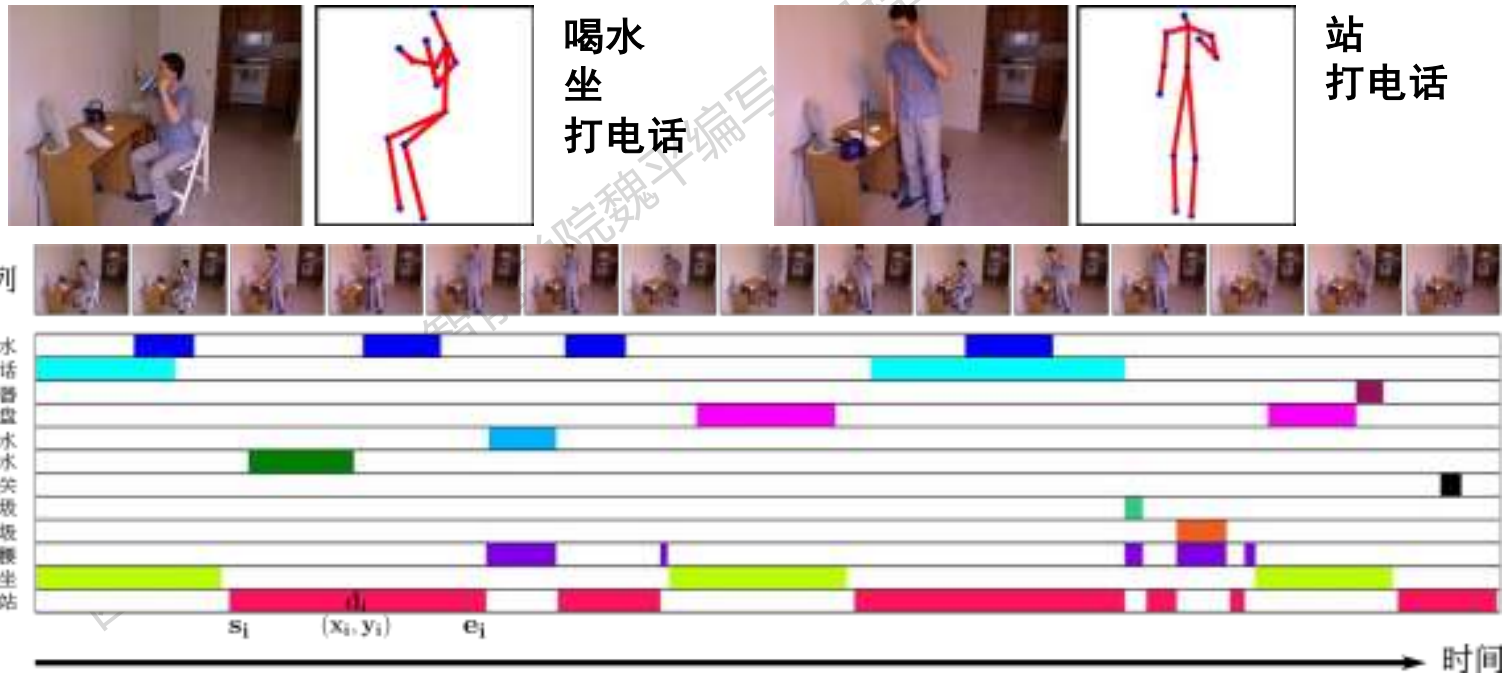
Is it standing or awake?

Q3: ✓

Is it a cat or a dog?

问题提出

- 并发行为：一个人在同一时刻发生的并行行为
- 任务目标：给定一个视频序列，检测并识别其中所有行为



数学模型

- 结构化预测模型

$$S(X,Y)=\underbrace{\sum_{i}^{M}\omega_{y_i}^{T}\rho_{y_i}(x_i)}_{\text{崩旱□柄□□}}+\underbrace{\sum_{i}^{M}\sum_{j\in N(i)}^{M}\omega_{y_i,y_j}^{T}r_{i,j}}_{\text{蜉痰□搏哄□}}$$

$X=\{x_i:i=1,2,...,M\}$ 视频特征序列

$Y=\{y_i:i=1,2,...,M\}$ 视频结构化解释序列

$\rho_{y_i}(x_i)$ 行为段特征

ω_{y_i} 一元行为模板

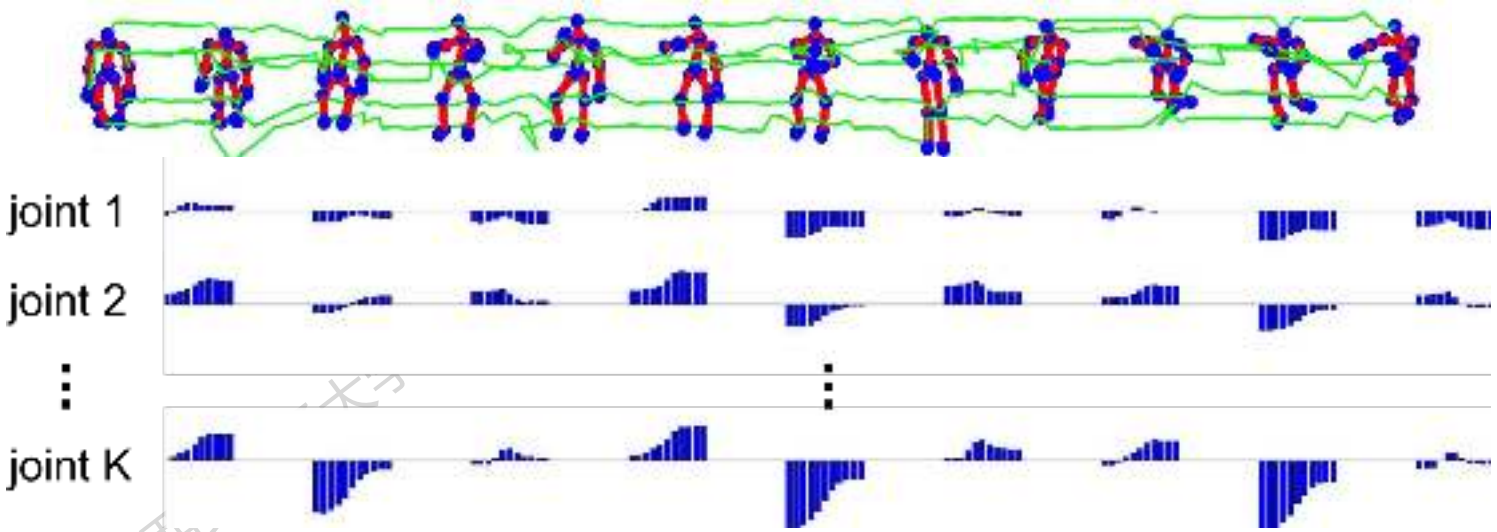
$r_{i,j}$ 时域语义关系特征

ω_{y_i,y_j} 语义关系模板

数学模型

- 小波行为特征和一元行为检测

$$\rho_{y_i}(x_i) = (f_{y_i}, 1) \qquad f_{y_i} = \beta_{y_i}^T x + b_{y_i}$$

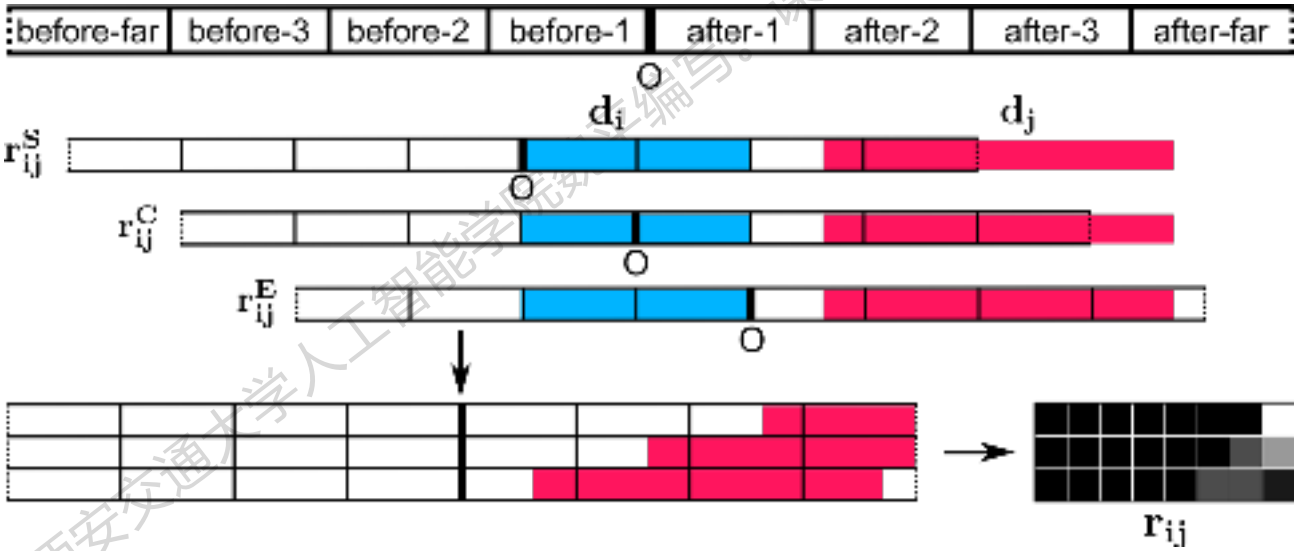


数学模型

- 行为的时域语义关系 – 组合时间逻辑特征

行为的时间逻辑， James F. Allen, 1984

before, equal, meet, overlap, during, start, finish



组合时间逻辑特征

模型学习

- 行为信息部的多核学习(multiple kernel learning)

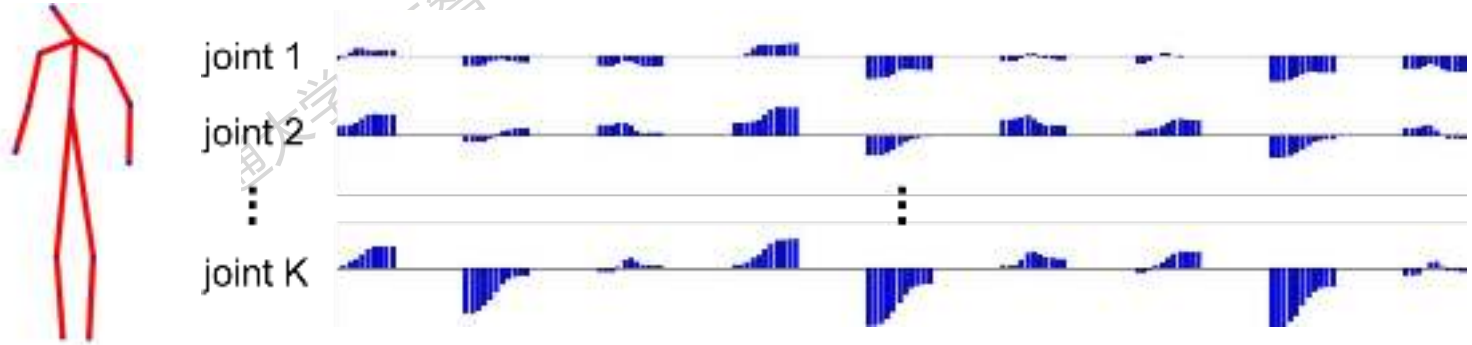
$$f_{y_i} = \beta_{y_i}^T x + b_{y_i} \qquad \textbf{学习参数} \qquad (\beta_{y_i}, b_{y_i})$$

多核学习

$$\min \quad \frac{1}{2}\left(\sum_{k=1}^{K} \alpha_k \| \beta_k \|_2\right)^2 + C\sum_{l=1}^{L} \xi_l$$

$$\text{w.s.t.} \quad \beta, b, \alpha_k \geq 0, \xi_l \geq 0$$

$$\text{s.t.} \quad z_l(\beta^T x_l + b) \geq 1 - \xi_l, \forall l \in 1, \ldots, L$$



模型学习

- 结构化支撑向量机学习(Structural SVM)

$$S(X,Y)=\sum_{i}^{M}\omega_{y_i}^{T}\rho_{y_i}(x_i)+\sum_{i}^{M}\sum_{j\in N(i)}^{M}\omega_{y_i,y_j}^{T}r_{i,j}$$

学习参数 $(\omega_{y_i},\omega_{y_i,y_j})$

公式推导：

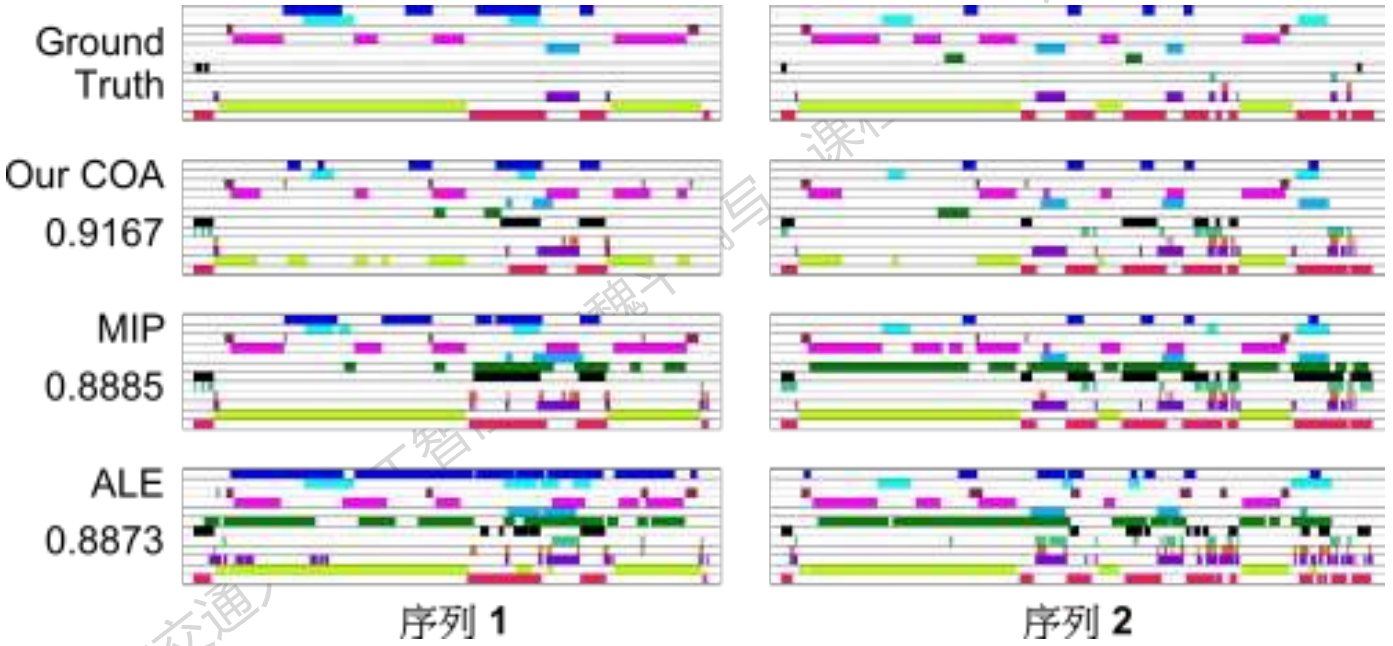
$$S(X,Y)=\sum_{i}^{M}\omega_{y_i}^{T}\rho_{y_i}(x_i)+\sum_{i}^{M}\sum_{j}^{M}\omega_{y_i,y_j}^{T}r_{i,j}$$

$$S(X,Y)=\sum_{i}^{M}\omega_{u}^{T}\varphi(x_i,y_i)+\sum_{i}^{M}\sum_{j}^{M}\omega_{b}^{T}\psi(r_{i,j},y_i,y_j)$$

$$\omega_u=\begin{bmatrix}\omega_{drink}\\ \omega_{call}\\ \vdots\\ \omega_{sit}\end{bmatrix}\quad \varphi(x_i,y_i)=\begin{bmatrix}\mathbf{0}\\ \mathbf{0}\\ \vdots\\ \rho_{y_i}(x_i)\\ \vdots\\ \mathbf{0}\end{bmatrix}\quad \omega_b=\begin{bmatrix}\omega_{drink,drink}\\ \omega_{drink,call}\\ \omega_{drink,sit}\\ \vdots\\ \omega_{sit,sit}\end{bmatrix}\quad \psi(x_i,y_j)=\begin{bmatrix}\mathbf{0}\\ \mathbf{0}\\ \vdots\\ r_{i,j}\\ \vdots\\ \mathbf{0}\end{bmatrix}$$

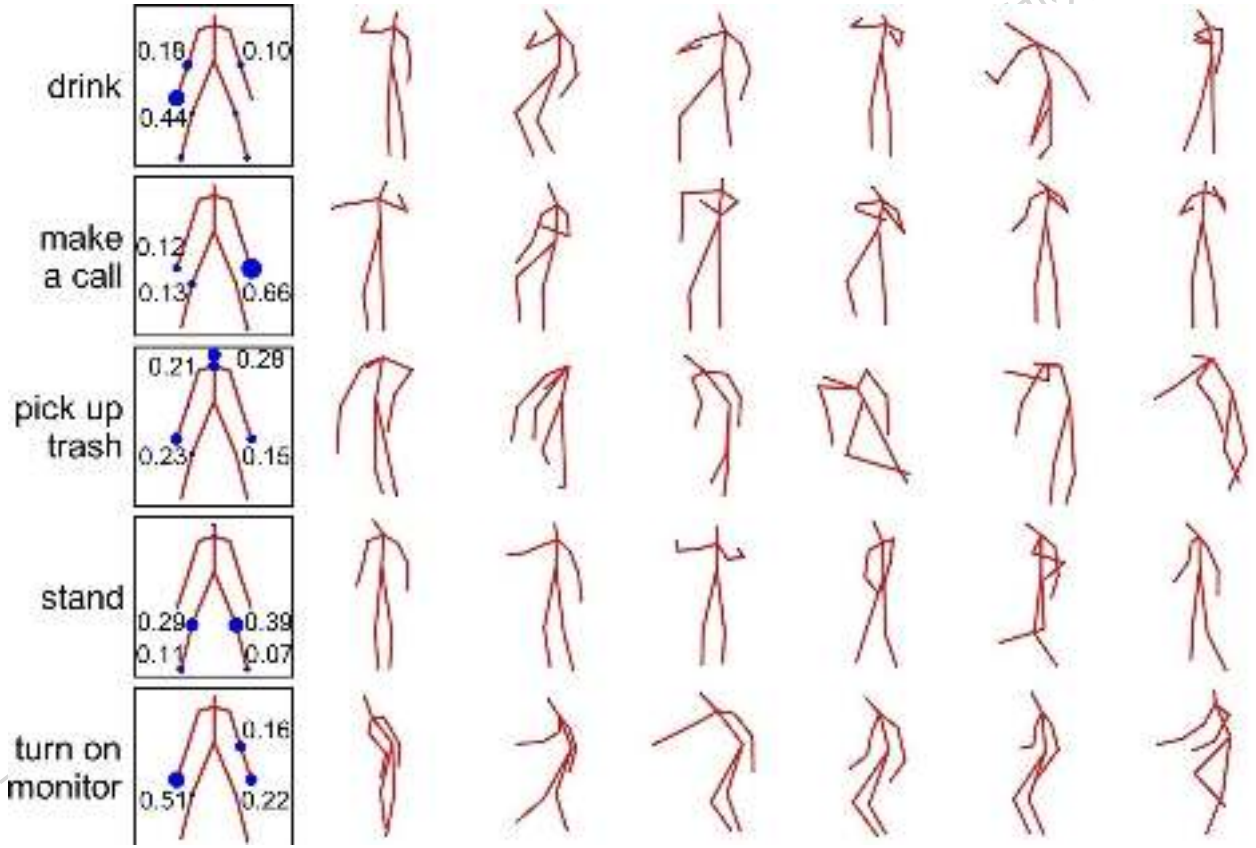
实验结果

- 并发行为检测



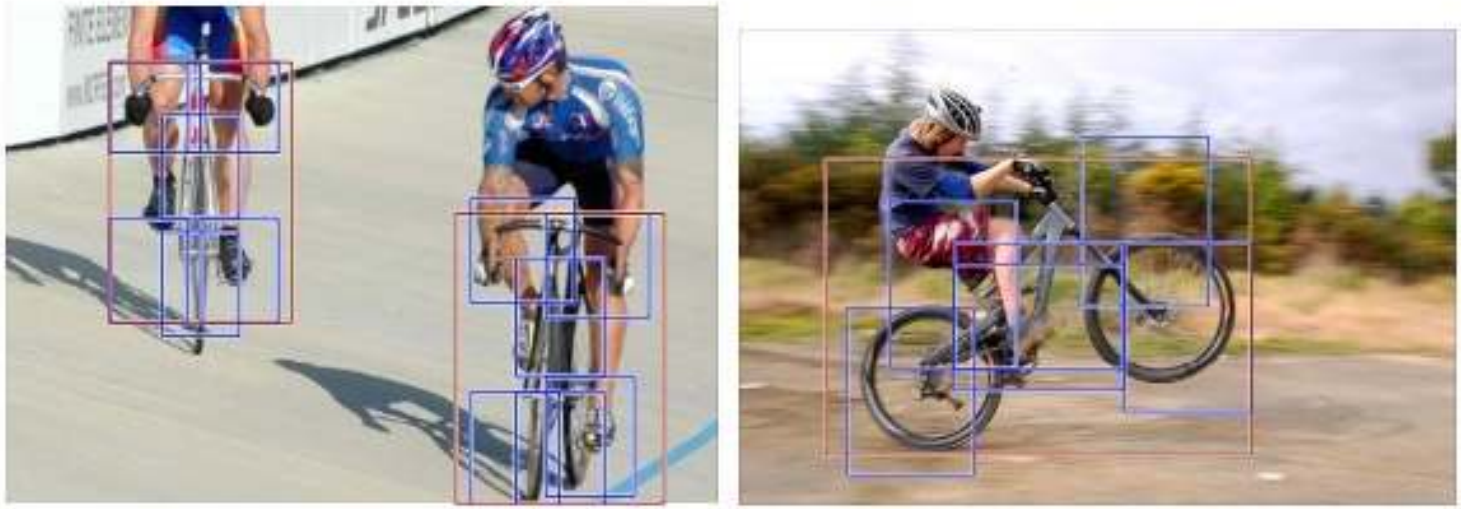
实验结果

- 人体行为信息部



目标检测的隐结构模型

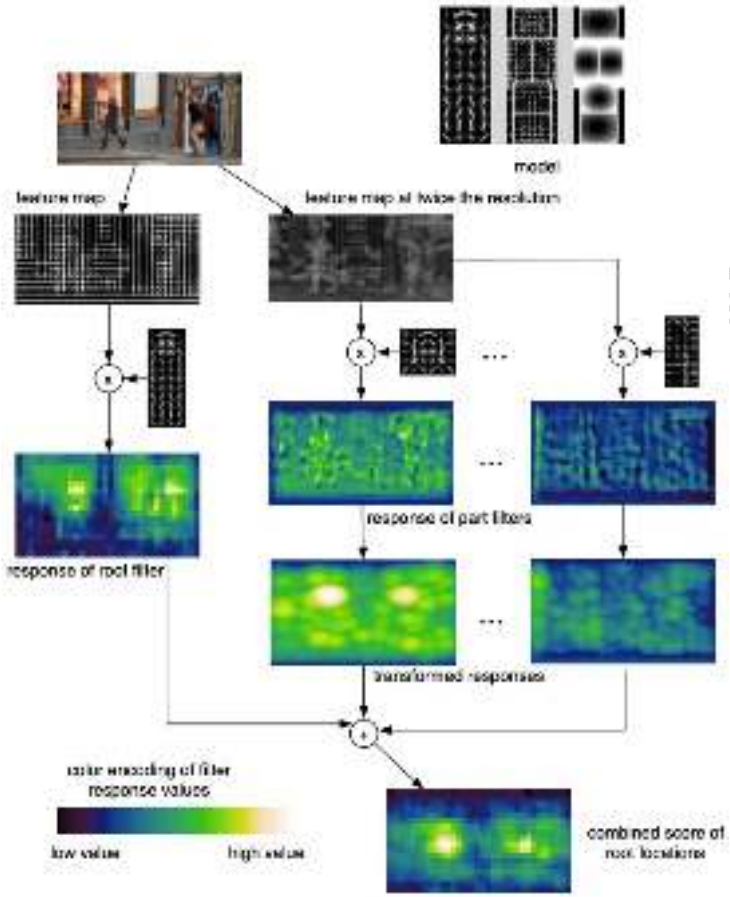
目标检测



Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester and Deva Ramanan, Object Detection with Discriminatively Trained Part Based Models, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 32(9), 2010

目标检测

- Latent SVM



$$\sum_{x',y'} F[x',y'] \cdot G[x+x', y+y']$$

$$f_\beta(x) = \max_{z \in Z(x)} \beta \cdot \Phi(x,z).$$

$$L_D(\beta) = \frac{1}{2}||\beta||^2 + C\sum_{i=1}^{n} \max(0, 1 - y_i f_\beta(x_i))$$

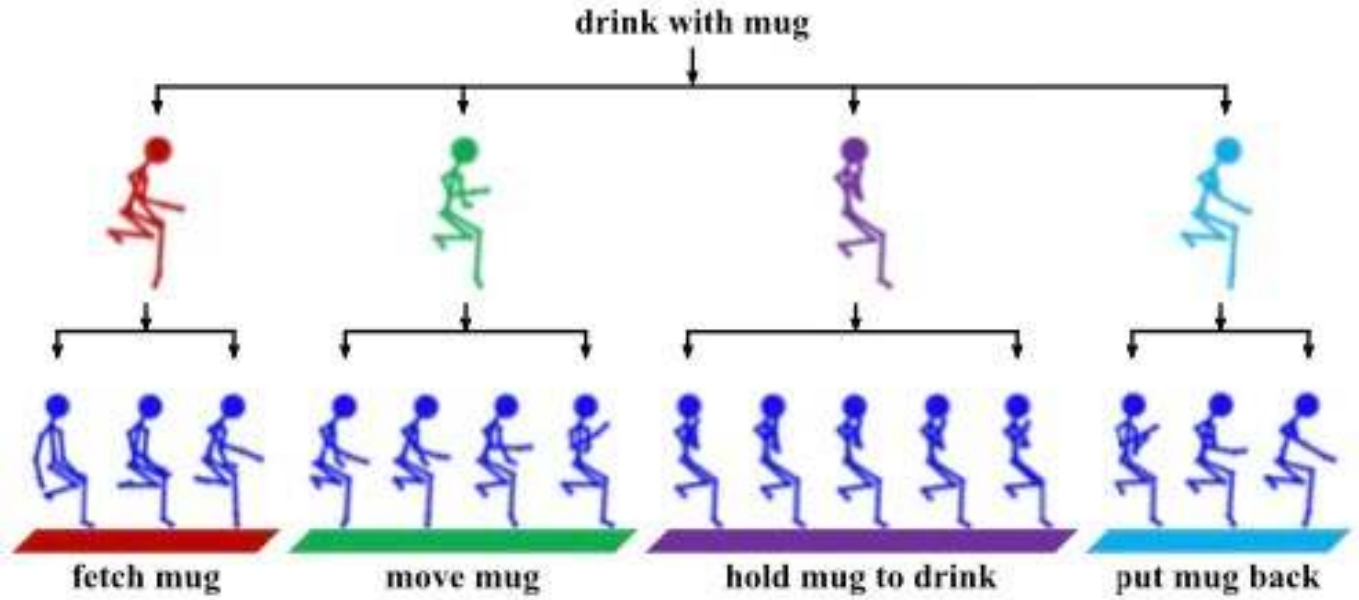
目标检测

行为识别的组合隐结构模型

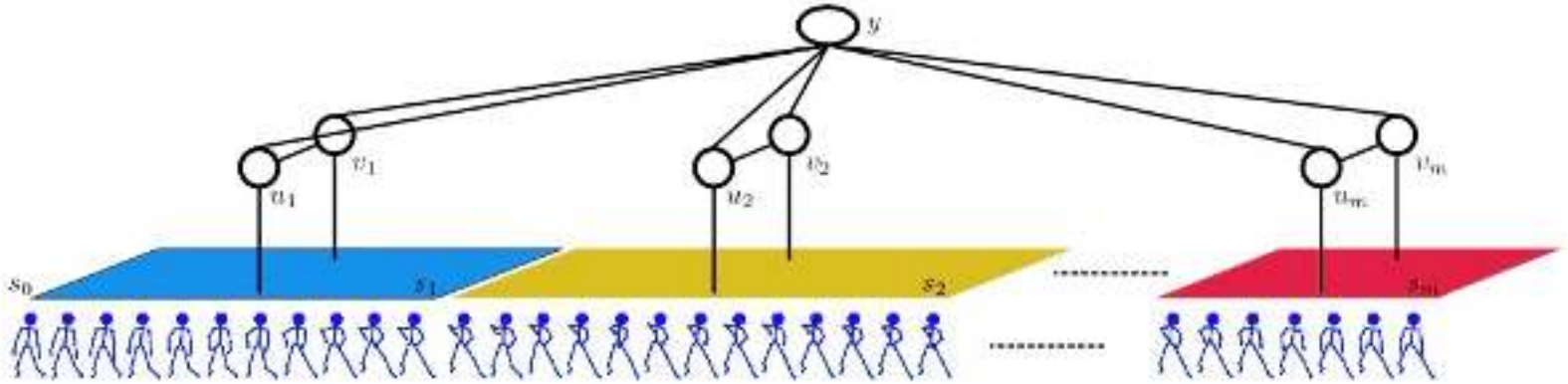
- Ping Wei, Hongbin Sun, Nanning Zheng, Composite Latent Structures for Action Representation and Recognition, IEEE Transactions On Multimedia, Vol. 21, No. 9, September 2019

行为识别

- 行为具有多隐结构特点



行为识别的组合隐结构模型



$$S(\mathbf{x},\mathbf{u},\mathbf{v},y)=\sum_{i=1}^{m}\sum_{t\in v_i}\phi(x_t,u_i,\boldsymbol{\omega})$$

$$\phi(x_t,u_i,\boldsymbol{\omega})=\log\frac{1/(1+e^{-\omega_{u_i}^{\mathrm{T}}\cdot x_t})}{\sum_{k=1}^{K}1/(1+e^{-\omega_k^{\mathrm{T}}\cdot x_t})}$$

$$f(\mathbf{x},y)=S(\mathbf{x},\mathbf{u}^*,\mathbf{v}^*,y)$$

$$(\mathbf{u}^*,\mathbf{v}^*)=\arg\max_{\mathbf{u},\mathbf{v}}S(\mathbf{x},\mathbf{u},\mathbf{v},y)$$

$$y^*=\arg\max_{y\in\mathcal{Y}}f(\mathbf{x},y)$$

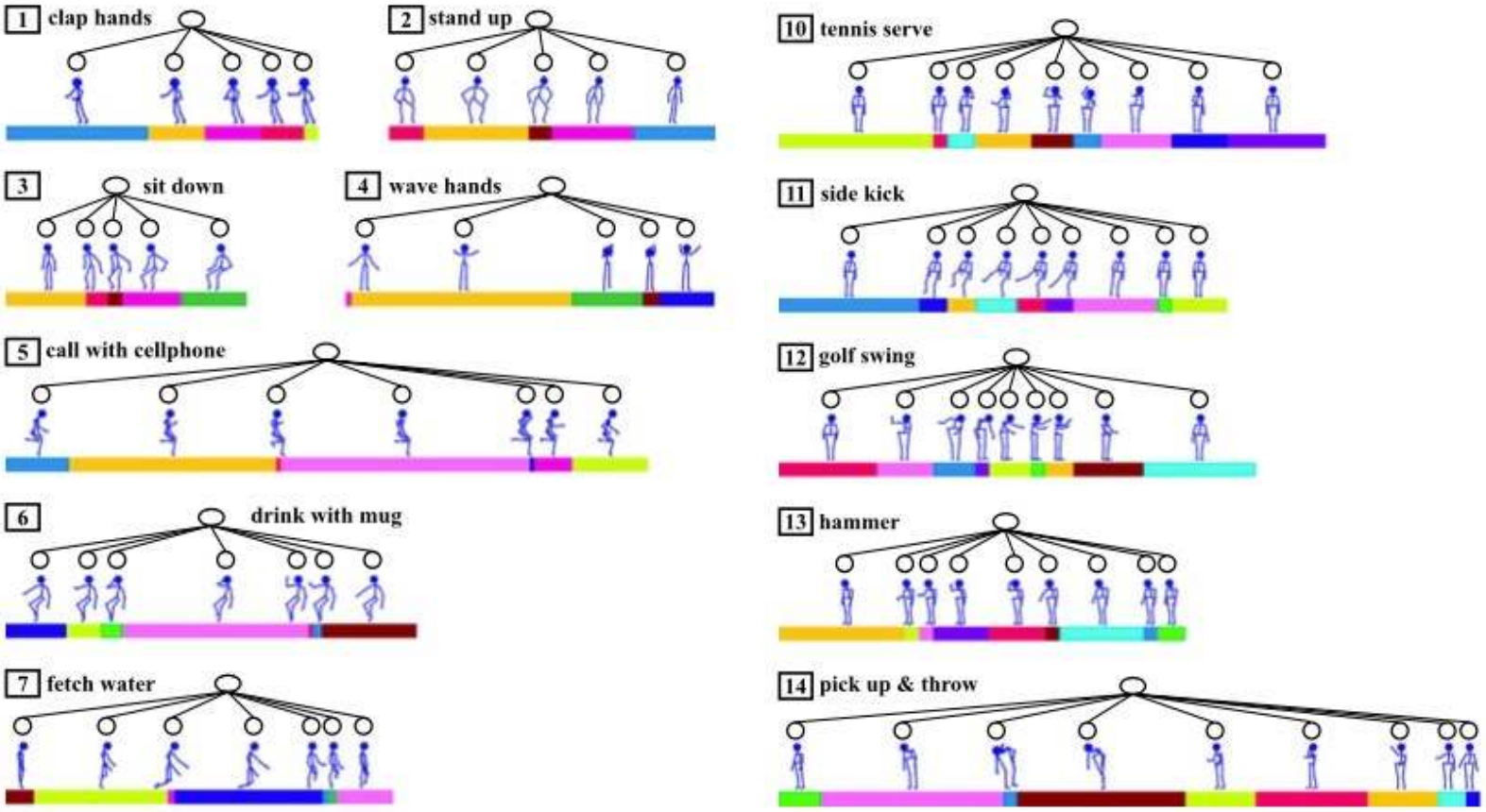
$\mathbf{x}=(x_1,x_2,\ldots,x_T)$ 输入序列数据

$\mathbf{u}=(u_1,u_2,\ldots,u_m)$ 隐类别序列

$\mathbf{v}=(v_1,v_2,\ldots,v_m)$ 隐位置序列

y 行为类别

行为识别的组合隐结构模型



学习资源

论文

- A Tutorial on Support Vector Machines for Pattern Recognition
- knowledge-Based Support Vector Machine Classifiers
- A Training Algorithm for Optimal Margin Classifiers

工具包

- libsvm: www.csie.ntu.edu.tw/~cjlin/libsvm/
- Svmlight: www.cs.cornell.edu/people/tj/svm_light/old/svm_light_v5.00.html
- Liblinear: www.csie.ntu.edu.tw/~cjlin/liblinear/
- Matlab: SVM Toolkit

西安交通大学
XI'AN JIAOTONG UNIVERSITY

IAIR Est. 1986
Institute of Artificial Intelligence and Robotics

人工智能学院
College of Artificial Intelligence, XJTU

The End