机器学习基础（二）

主讲：王皓 副研究员| 硕士导师

邮箱：wanghao3@ustc.edu.cn

主页：http://staff.ustc.edu.cn/~wanghao3

本章内容

- 线性模型
  - 分类问题示例
  - 线性分类模型

- 无监督学习
  - 原型聚类
  - 无监督特征学习
  - 概率密度估计

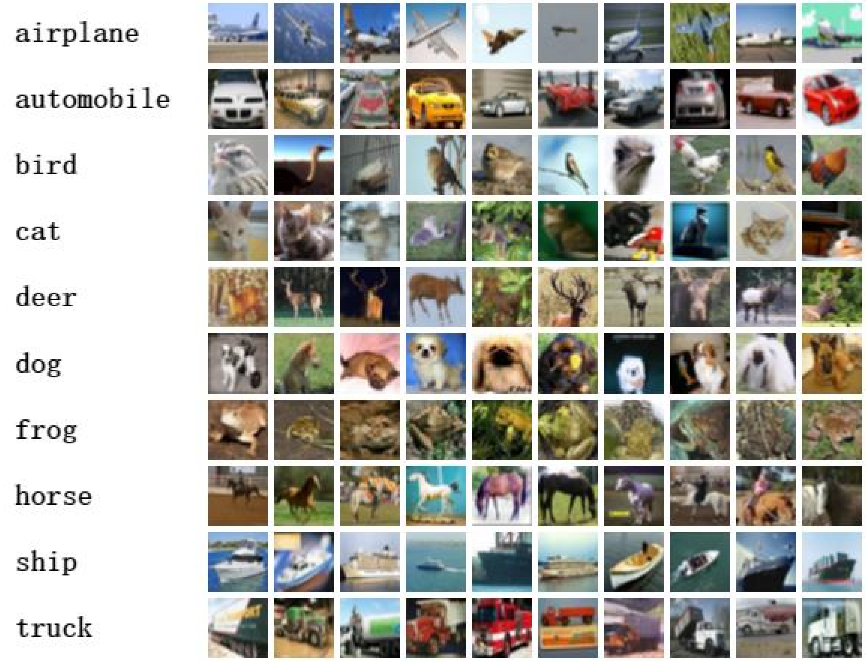
线性模型

示例：图像分类

➢任务描述：其目标是根据图像信息中所反映的不同特征，把不同类别的图像区分开来，并从已知的类别标签集合中为给定的输入图片赋予一个类别标签。

➢数据集：CIFAR-10

- 60000张32x32色彩图像，共10类
- 每类6000张图像



示例：图像分类

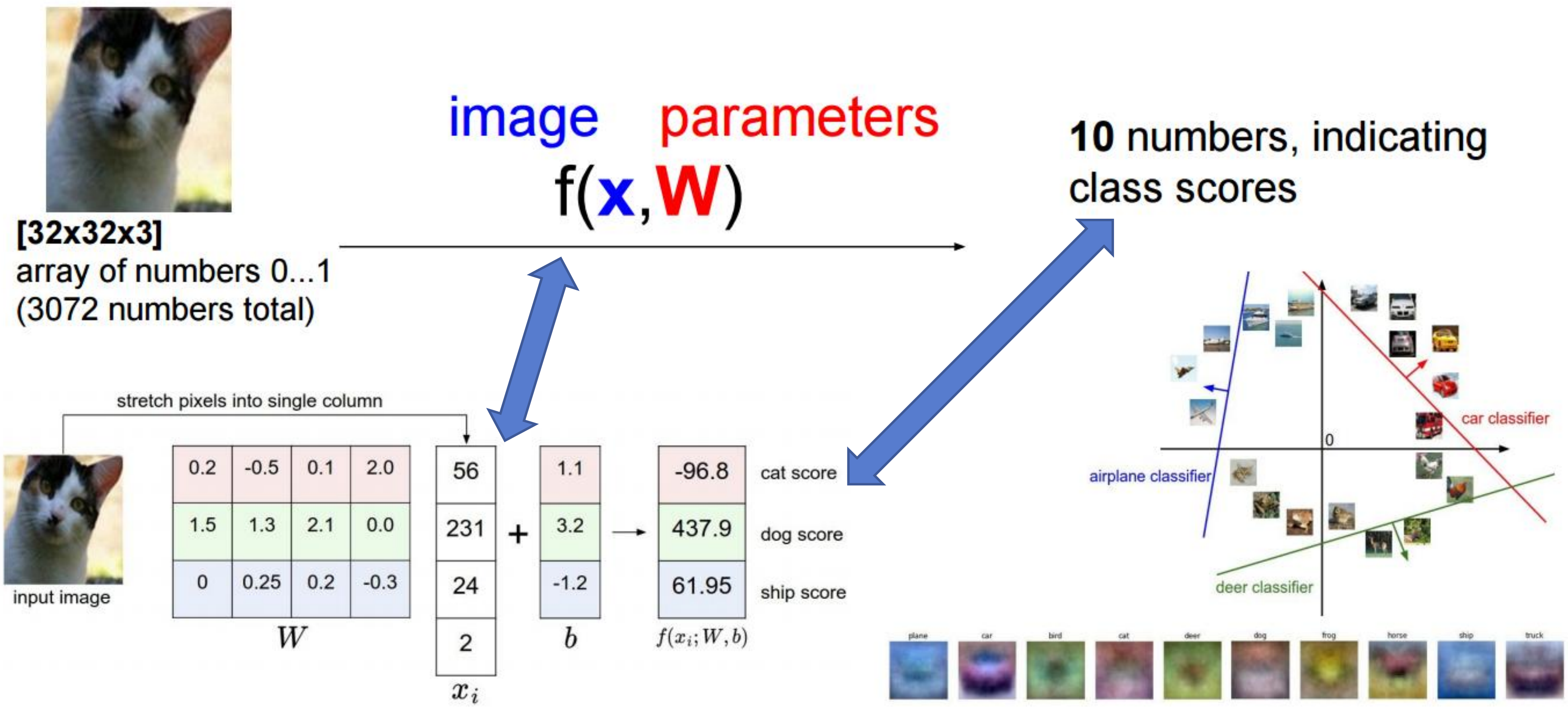
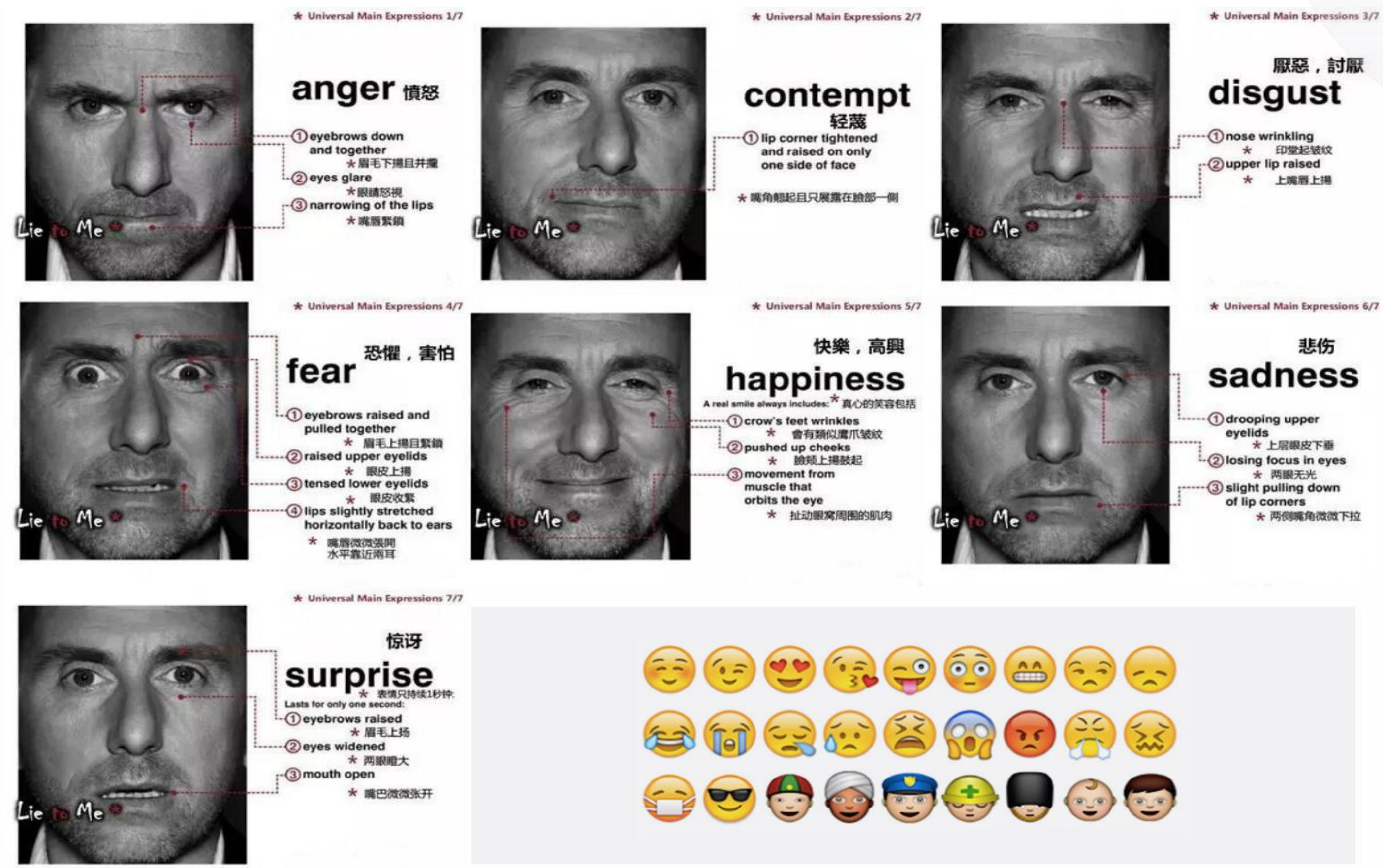


image
parameters
f(x,W)
[32x32x3]
array of numbers 0...1
(3072 numbers total)
10 numbers, indicating
class scores
stretch pixels into single column
input image
0.2 -0.5 0.1 2.0
1.5 1.3 2.1 0.0
0 0.25 0.2 -0.3
W
56
231
24
2
x_i
+
1.1
3.2
-1.2
b
-96.8
437.9
61.95
$f(x_i; W, b)$
cat score
dog score
ship score
airplane classifier
car classifier
deer classifier
0
plane
car
bird
cat
deer
dog
frog
horse
ship
truck

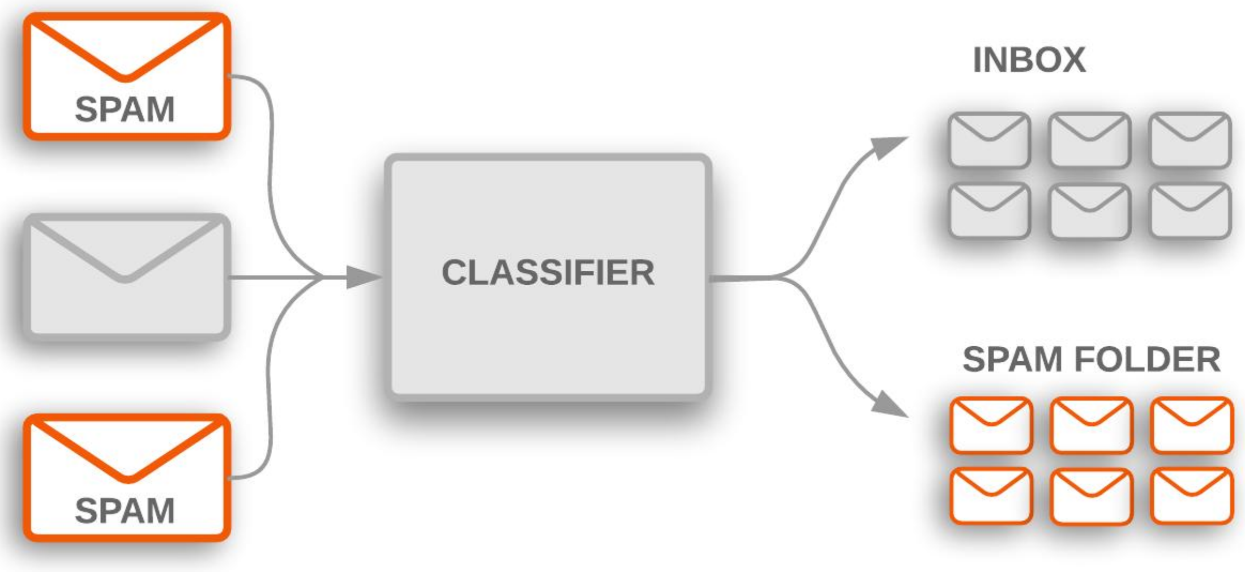
示例：情感分类

➤通过对面部特征的提取，判定人的表情对应的情感



示例：垃圾邮件过滤

➤根据邮件标题、发件人等内容进行垃圾邮件过滤



示例：文档归类

➤为新闻、论文等文本进行归类

https://towardsdatascience.com/automated-text-classification-using-machine-learning-3df4f4f9570b

示例：文本分类

- ➢将样本x从文本形式转为向量形式
  - 词袋模型（Bag-of-Words，BoW）模型

the dog is on the table

0	0	1	1	0	1	1	1
are	cat	dog	is	now	on	table	the

比如两个文本“我 喜欢 读书”和“我 讨厌 读书”中共有“我”、“喜欢”、“讨厌”、“读书”四个词，它们的BoW表示分别为

$$\mathbf{v}_1 = [1\ 1\ 0\ 1\ ]^{\mathrm{T}},$$

$$\mathbf{v}_2 = [1\ 0\ 1\ 1\ ]^{\mathrm{T}}.$$

示例：文本情感分类

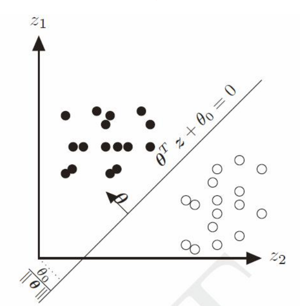
➢ 根据文本内容来判断文本的相应类别

Review (X)	Rating (Y)
"This movie is fantastic! I really like it because it is so good!"	★★★★☆
"Not to my taste, will skip and watch another movie"	★★☆☆☆
"This movie really sucks! Can I get my money back please?"	★☆☆☆☆

D_1：“我喜欢读书”

D_2：“我讨厌读书”

	我	喜欢	讨厌	读书
D_1	1	1	0	1
D_2	1	0	1	1



+

-

线性模型

- 线性模型（Linear Model）是机器学习中应用最广泛的模型，指通过样本特征的线性组合来进行预测的模型。给定一个 D 维样本 $\boldsymbol{x}=[x_1,x_2,\dots,x_D]^{\mathrm{T}}$，其线性组合函数为

$$
\begin{aligned}
f(\boldsymbol{x};\boldsymbol{w}) &= w_1x_1+w_2x_2+\cdots+w_Dx_D+b \\
&= \boxed{\boldsymbol{w}^{\mathrm{T}}}\boldsymbol{x}+\boxed{b},
\end{aligned}
$$

偏置

权重向量$\boldsymbol{w}=[\omega_1,\omega_2,\dots,\omega_D]^{\mathrm{T}}$

- 回归问题：$y=f(\boldsymbol{x};\boldsymbol{w})$，值域为实数
- 分类问题：$y=g(f(\boldsymbol{x};\boldsymbol{w}))$，$f(\boldsymbol{x};\boldsymbol{w})$也称判别函数

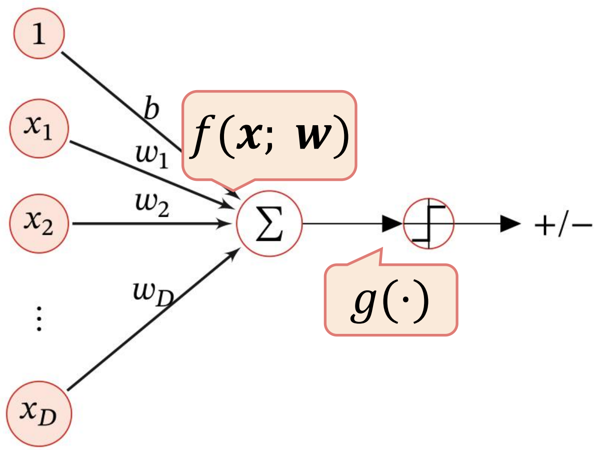
引入一个**非线性的决策函数（Decision Function）**$g(\cdot)$来预测输出目标

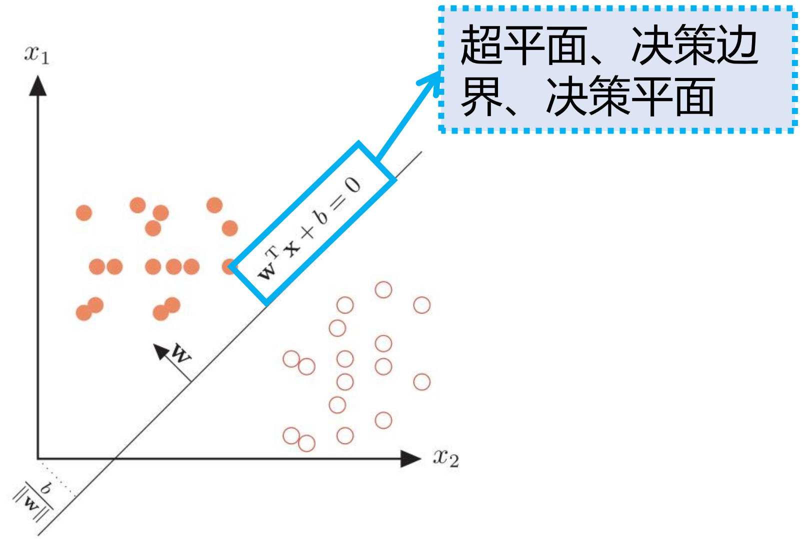
二分类问题

$g(\cdot)$可以是符号函数（Sign Function），定义为

$$g(f(\boldsymbol{x};\boldsymbol{w})) = \operatorname{sgn}(f(\boldsymbol{x};\boldsymbol{w}))$$

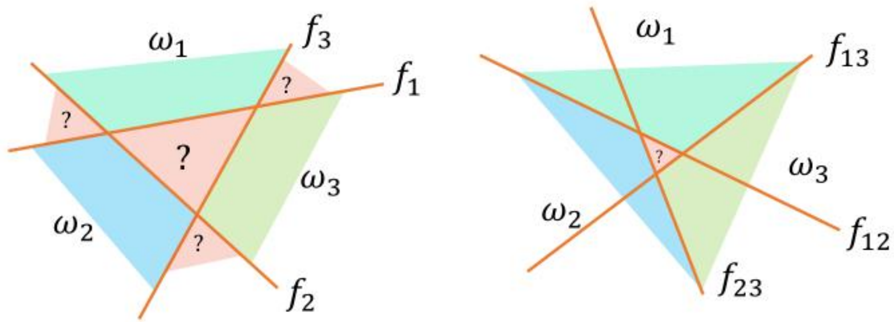
$$\triangleq \begin{cases} +1 & \text{if } f(\boldsymbol{x};\boldsymbol{w}) > 0, \\ -1 & \text{if } f(\boldsymbol{x};\boldsymbol{w}) < 0. \end{cases}$$





多分类问题

➢ 多分类（Multi-class Classification）问题是指分类的类别数 C 大于2，因此一般需要多个线性判别函数，但设计这些判别函数有很多种方式，常用的有以下三种：



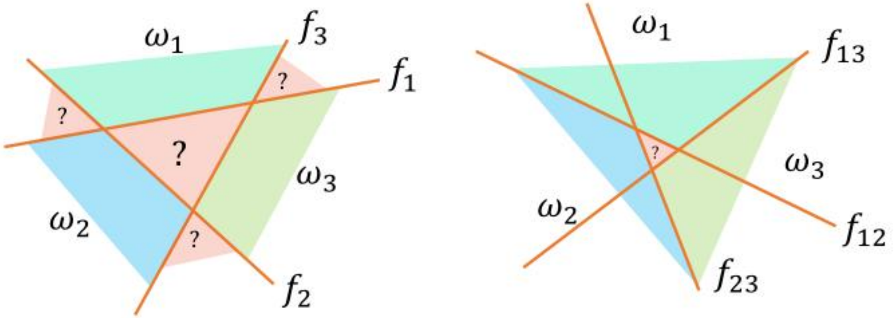
(a) “一对其余”方式　(b) “一对一”方式　(c) “argmax”方式

- “一对其余”：把多分类问题转换为 C 个“一对其余”的二分类问题。这种方式共需要 C 个判别函数，每个判别函数的目的是将属于类别 c 和不属于类别 c 的样本区分开。
- “一对一”：把多分类问题转换为 $C(C-1)/2$ 个“一对一”的二分类问题。这种方式共需要 $C(C-1)/2$ 个判别函数，每个判别函数的目的是将属于类别 i 和属于类别 j 的样本区分开。

✓ 上述两种方式都存在一个缺陷：特征空间中存在会存在一些难以确定类别的区域

多分类问题

➢ 多分类（Multi-class Classification）问题是指分类的类别数 C 大于2，因此一般需要多个线性判别函数，但设计这些判别函数有很多种方式，常用的有以下三种：



(a) “一对其余”方式　　(b) “一对一”方式　　(c) “argmax”方式

- “argmax”：一种改进的“一对其余”方式，共需要 C 个判别函数。

$$f_c(\boldsymbol{x};\ \boldsymbol{w}_c)\ =\ \boldsymbol{w}_c^{\mathrm{T}}\boldsymbol{x}\ +\ b_c,\qquad c\ \in\ \{1,\cdots,C\}$$

对于样本 $\boldsymbol{x}$，如果存在一个类别 c，相对于所有的其他类别 $\tilde{c}(\tilde{c}\neq c)$ 有 $f_c(\boldsymbol{x};\ \boldsymbol{w}_c)\ >f_{\tilde{c}}(\boldsymbol{x};\ \boldsymbol{w}_{\tilde{c}})$，那么 $\boldsymbol{x}$ 属于类别 c。则“argmax”方式的预测函数定义为

$$\boxed{y=\underset{c=1}{\overset{C}{\arg\max}}\, f_c(\boldsymbol{x};\boldsymbol{w}_c).}$$

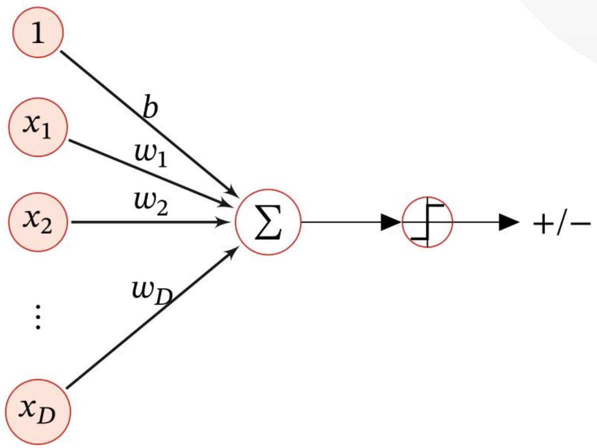
线性分类模型

- Logistic回归
- Softmax回归
- 支持向量机
- ……

Logistic Regression

➤模型（本节中采用 $y \in \{0, 1\}$ 以符合Logistic回归的描述习惯）

$$g(f(\boldsymbol{x};\boldsymbol{w})) = \begin{cases} 1 & \text{if} \quad f(\boldsymbol{x};\boldsymbol{w}) > 0 \\ 0 & \text{if} \quad f(\boldsymbol{x};\boldsymbol{w}) < 0 \end{cases}$$



分类问题

- 将分类问题看作条件概率估计问题
  - 引入非线性函数 $g: \mathbb{R}^D \to (0,1)$ 来预测类别标签的条件概率 $p(y=c|\mathbf{x})$。
  - 以二分类为例：

$$p(y=1|\mathbf{x}) = g(f(\mathbf{x};\mathbf{w}))$$

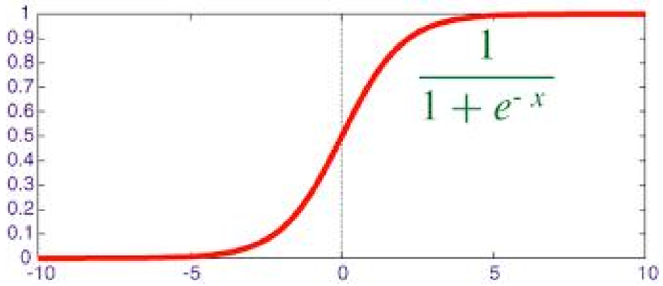
  - 函数f： 线性函数
  - 函数g： 称为激活函数，把线性函数的值域从实数区间“挤压”到了(0,1)之间，可以用来表示概率。

如何构建函数g?

Logistic函数与回归

- Logistic函数

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



- Logistic回归

$$p(y=1|\boldsymbol{x}) = \sigma(\boxed{\boldsymbol{w}^\top \boldsymbol{x}})$$

$$\triangleq \frac{1}{1+\exp(-\boldsymbol{w}^\top \boldsymbol{x})}$$

增广特征向量 $\boldsymbol{x} = [x_1, x_2, \ldots, x_D, 1]^{\mathrm{T}}$

增广权重向量 $\boldsymbol{w} = [\omega_1, \omega_2, \ldots, \omega_D, b]^{\mathrm{T}}$

学习准则

➢模型预测条件概率 $p_\theta(y|\boldsymbol{x})$

$$p_\theta(y=1|\boldsymbol{x}) = \sigma(\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x})$$

➢真实条件概率 $p_r(y|\boldsymbol{x})$

- 对于一个样本(x, y^*)，$y^* \in \{0, 1\}$，其真实条件概率为

$$p_r(y=1|\boldsymbol{x}) = y^*$$

$$p_r(y=0|\boldsymbol{x}) = 1 - y^*$$

如何衡量两个条件分布的差异?

熵（Entropy）

➢在信息论中，熵用来衡量一个随机事件的不确定性。

- 自信息（Self Information）

$$I(x) = -\log(p(x))$$

- 熵

$$H(X) = \mathbb{E}_X[I(x)] = \mathbb{E}_X[-\log p(x)] = -\sum_{x \in \mathcal{X}} p(x) \log p(x)$$

✓熵越高，则事件的不确定性越大，随机变量的信息越多；

✓熵越低，则事件的不确定性越小，随机变量的信息越少。

- 在对某一特定概率分布 $q(y)$ 进行编码时，熵 $H(q)$ 也是理论上最优的平均编码长度，这种编码方式称为熵编码（Entropy Encoding）

交叉熵（Cross Entropy）

- 交叉熵是按照概率分布 q 的最优编码对真实分布为 p 的信息进行编码的长度。

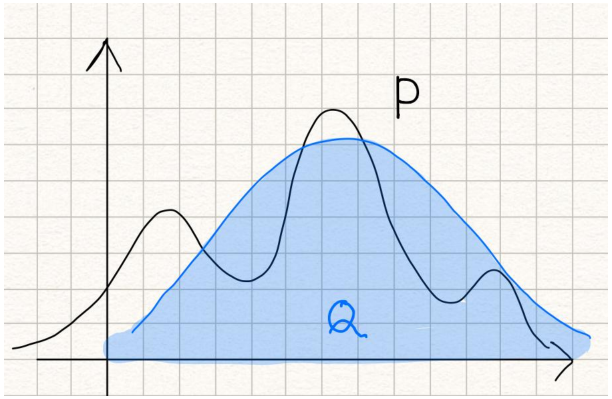
$$\begin{aligned} H(p,q) &= \mathbb{E}_p[-\log q(x)] \\ &= -\sum_x p(x) \log q(x) \end{aligned}$$

在给定分布 q 的情况下

- 如果 p 和 q 越接近，交叉熵越小；
- 如果 p 和 q 越远，交叉熵就越大。

KL散度（Kullback-Leibler Divergence）

- KL散度是用概率分布 q 来近似 p 时所造成的信息损失量。
  - KL散度是按照概率分布 q 的最优编码对真实分布为 p 的信息进行编码，其平均编码长度（即交叉熵）$H(p,q)$ 和 p 的最优平均编码长度（即熵）$H(p)$ 之间的差异。



交叉熵 — 熵

$$\mathrm{KL}(p,q) = H(p,q) - H(p)$$

$$= \sum_x p(x) \log \frac{p(x)}{q(x)}$$

越小效果越好

交叉熵损失

$$\mathrm{D}_{KL}(p_r(y|x)\|p_\theta(y|x)) = \sum_{y=0}^{1} p_r(y|x)\log\frac{p_r(y|x)}{p_\theta(y|x)}$$ KL散度

$$\propto -\sum_{y=0}^{1} p_r(y|x)\log p_\theta(y|x)$$ 交叉熵损失

$y^* \in \{0,1\}$为x的真实标签

$$= -I(y^*=1)\log p_\theta(y=1|x) - I(y^*=0)\log p_\theta(y=0|x)$$

$$= -y^*\log p_\theta(y=1|x) - (1-y^*)\log p_\theta(y=0|x)$$

$$= -\log p_\theta(y^*|x)$$ 负对数似然

梯度下降法

- 给定 N 个训练样本$\{(x^{(n)}, y^{(n)})\}_{n=1}^{N}$，用Logistic回归模型对每个样本 $x^{(n)}$ 进行预测，输出其标签为1的后验概率，记为$\hat{y}^{(n)}$。

$$\hat{y}^{(n)} = \sigma(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}^{(n)}),\ 1 \le n \le N$$

- 由于 $y^{(n)} \in \{0, 1\}$，样本 $(\boldsymbol{x}^{(n)}, y^{(n)})$ 的真实条件概率可以表示为

$$p_r(y^{(n)} = 1|\boldsymbol{x}^{(n)}) = y^{(n)}$$

$$p_r(y^{(n)} = 0|\boldsymbol{x}^{(n)}) = 1 - y^{(n)}$$

- 故使用交叉熵损失函数，模型在训练集的风险函数为

$$\mathcal{R}(\boldsymbol{w}) = -\frac{1}{N}\sum_{n=1}^{N}(p_r(y^{(n)} = 1|\boldsymbol{x}^{(n)})log\hat{y}^{(n)} + p_r(y^{(n)} = 0|\boldsymbol{x}^{(n)})log(1 - \hat{y}^{(n)}))$$

化简后得：$\mathcal{R}(\boldsymbol{w}) = -\frac{1}{N}\sum_{n=1}^{N}(y^{(n)}log\hat{y}^{(n)} + (1 - y^{(n)})log(1 - \hat{y}^{(n)}))$

梯度下降法

- 风险函数$\mathcal{R}(\boldsymbol{w})$关于参数 $\boldsymbol{w}$ 的偏导数为

$$\frac{\partial \mathcal{R}(\mathbf{w})}{\partial \mathbf{w}} = -\frac{1}{N}\sum_{n=1}^{N}\left(y^{(n)}\frac{\hat{y}^{(n)}(1-\hat{y}^{(n)})}{\hat{y}^{(n)}}\mathbf{x}^{(n)} - (1-y^{(n)})\frac{\hat{y}^{(n)}(1-\hat{y}^{(n)})}{1-\hat{y}^{(n)}}\mathbf{x}^{(n)}\right)$$

$$= -\frac{1}{N}\sum_{n=1}^{N}\left(y^{(n)}(1-\hat{y}^{(n)})\mathbf{x}^{(n)} - (1-y^{(n)})\hat{y}^{(n)}\mathbf{x}^{(n)}\right)$$

$$= -\frac{1}{N}\sum_{n=1}^{N}\mathbf{x}^{(n)}\left(y^{(n)} - \hat{y}^{(n)}\right).$$

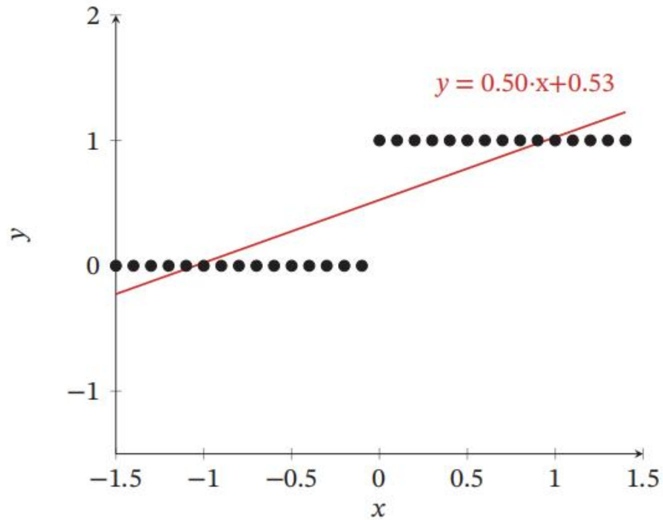
- 采用梯度下降法，Logistic回归的训练过程为：
  - ✓ 初始化 $\boldsymbol{w}_0 \leftarrow 0$
  - ✓ 迭代更新参数： $\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t + \alpha\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}^{(n)}(y^{(n)} - \hat{y}_{\boldsymbol{w}_t}^{(n)})$

Logistic回归

本质上是线性判别y=wx+b>=0?1:0

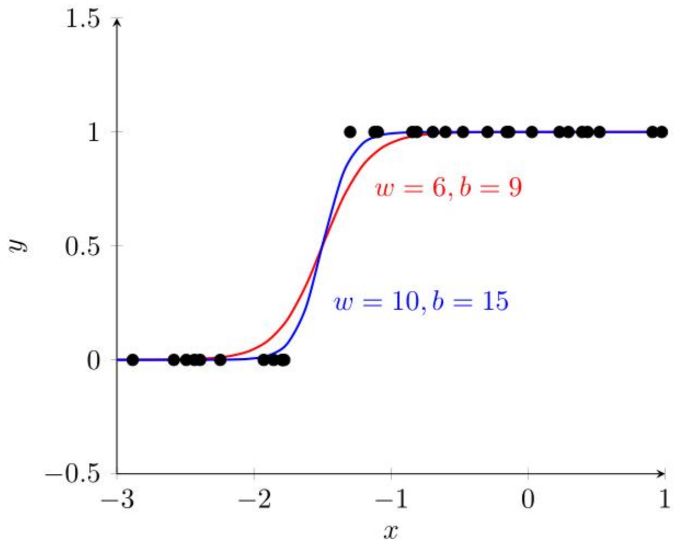
- 一维数据的二分类问题示例

输出连续



线性回归

输出$\in${0,1}



Logistic 回归

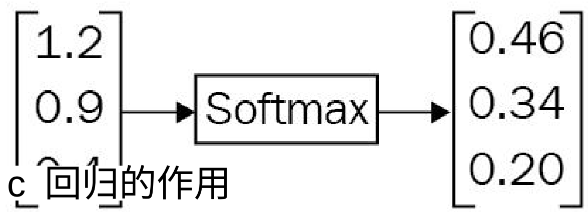
Softmax回归

没有3元线性

➢Softmax 回归，也称为多项或多类的Logistic回归，是Logistic回归在多分类问题上的推广。

✓Softmax函数

$$\mathrm{softmax}(x_k) = \frac{\exp(x_k)}{\sum_{i=1}^{K} \exp(x_i)}$$



logiistic 回归的作用

✓对于多类问题，类别标签 $y \in \{1, 2, \cdots, C\}$ 可以有 C 个取值。给定一个样本 $\boldsymbol{x}$，Softmax回归预测的属于类别 c 的条件概率为：

$$\begin{aligned} p(y = c|\boldsymbol{x}) &= \mathrm{softmax}(\boldsymbol{w}_c^{\mathsf{T}}\boldsymbol{x}) \\ &= \frac{\exp(\boldsymbol{w}_c^{\mathsf{T}}\boldsymbol{x})}{\sum_{c'=1}^{C} \exp(\boldsymbol{w}_{c'}^{\mathsf{T}}\boldsymbol{x})}, \end{aligned}$$

Softmax回归

➢因此Softmax回归的决策函数可以表示为：

$$
\begin{aligned}
\hat{y} &= \arg\max_{c=1}^{C} p(y=c \mid x) \\
&= \arg\max_{c=1}^{C} \frac{\exp(\omega_c^T x)}{\sum_{c'}^{C} \exp(\omega_{c'}^T x)} \\
&= \arg\max_{c=1}^{C} \omega_c^T x
\end{aligned}
$$

参数学习

- 给定 N 个训练样本$\{(\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)})\}_{n=1}^{N}$， $\boldsymbol{y}^{(n)}$用C维one-hot向量表示
  - 学习准则：使用交叉熵损失函数，Softmax回归模型的风险函数为

$$\mathcal{R}(\boldsymbol{W}) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} \boldsymbol{y}_c^{(n)} \log \hat{\boldsymbol{y}}_c^{(n)}$$

$$= -\frac{1}{N}\sum_{n=1}^{N} (\boldsymbol{y}^{(n)})^{\mathrm{T}} \log \boxed{\hat{\boldsymbol{y}}^{(n)}},$$

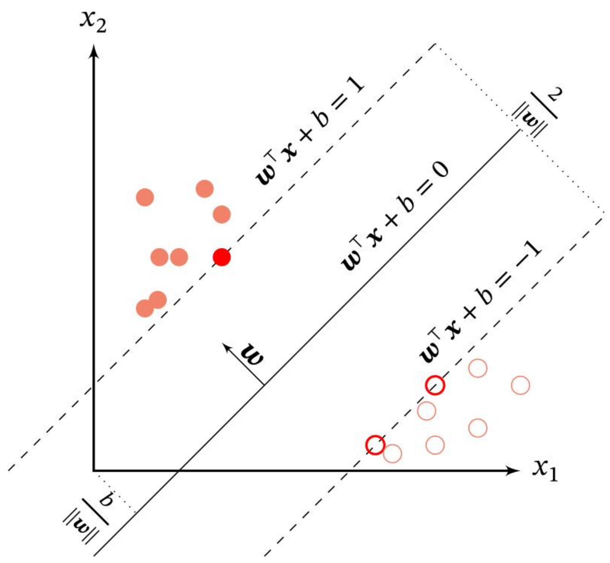
$\hat{\boldsymbol{y}}^{(n)} = softmax(\boldsymbol{W}^T\boldsymbol{x}^{(n)})$
为样本 $\boldsymbol{x}^{(n)}$ 在每个类别的后验概率

  - 优化：使用梯度下降法

$$\frac{\partial \mathcal{R}(\boldsymbol{W})}{\partial \boldsymbol{W}} = -\frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}^{(n)}\left(\boldsymbol{y}^{(n)} - \hat{\boldsymbol{y}}^{(n)}\right)^{\mathrm{T}}.$$

- 参数更新：$\boldsymbol{W}_{t+1} \leftarrow \boldsymbol{W}_t + \alpha\left(\frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}^{(n)}\left(\boldsymbol{y}^{(n)} - \hat{\boldsymbol{y}}^{(n)}\boldsymbol{W}_t\right)^{\mathrm{T}}\right)$

支持向量机（Support Vector Machine，SVM）



数据集中所有满足 $y^{(n)}(\boldsymbol{w}^\mathrm{T}\boldsymbol{x}^{(n)} + b) = 1$ 的样本点，都称为“支持向量”

- 数据集 $\mathcal{D}$ 中每个样本 $\boldsymbol{x}^{(n)}$ 到分割超平面 $\boldsymbol{w}^\mathrm{T}\boldsymbol{x} + b = 0$的距离为：

$$\gamma^{(n)} = \frac{|\boldsymbol{w}^\top \boldsymbol{x}^{(n)} + b|}{\|\boldsymbol{w}\|} = \frac{y^{(n)}(\boldsymbol{w}^\top \boldsymbol{x}^{(n)} + b)}{\|\boldsymbol{w}\|}.$$

- 支持向量机的目标是寻找一个超平面$(\boldsymbol{w}^*, b^*)$使得 γ 最大，即

$$\max_{\boldsymbol{w},b} \quad \gamma$$

$$\text{s.t.} \quad \frac{y^{(n)}(\boldsymbol{w}^\top \boldsymbol{x}^{(n)} + b)}{\|\boldsymbol{w}\|} \geq \gamma, \forall n \in \{1, \cdots, N\}.$$

$$\max_{\boldsymbol{w},b} \quad \frac{1}{\|\boldsymbol{w}\|^2}$$

$$\text{s.t.} \quad y^{(n)}(\boldsymbol{w}^\top \boldsymbol{x}^{(n)} + b) \geq 1, \forall n \in \{1, \cdots, N\}.$$

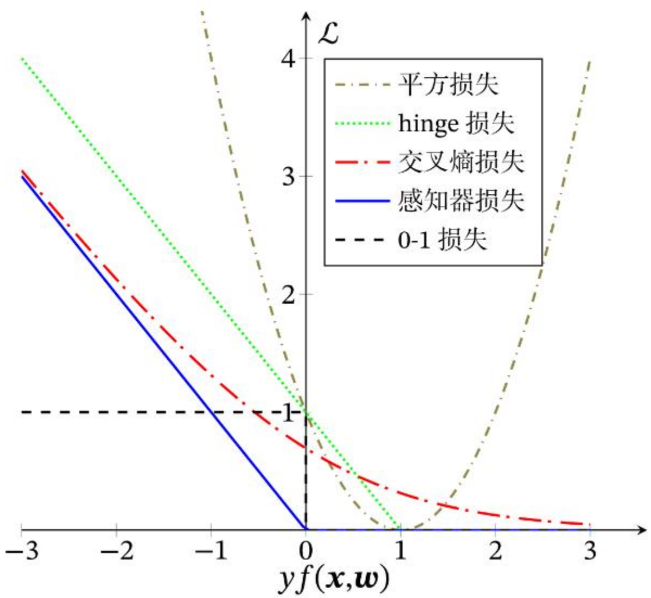
线性分类模型小结

线性模型	激活函数	损失函数	优化方法
线性回归	-	$(y - \boldsymbol{w}^{\top}\boldsymbol{x})^2$	最小二乘、梯度下降
Logistic 回归	$\sigma(\boldsymbol{w}^{\top}\boldsymbol{x})$	$\boldsymbol{y} \log \sigma(\boldsymbol{w}^{\top}\boldsymbol{x})$	梯度下降
Softmax 回归	$\mathrm{softmax}(\boldsymbol{W}^{\top}\boldsymbol{x})$	$\boldsymbol{y} \log \mathrm{softmax}(\boldsymbol{W}^{\top}\boldsymbol{x})$	梯度下降
感知器	$\mathrm{sgn}(\boldsymbol{w}^{\top}\boldsymbol{x})$	$\max(0, -y\boldsymbol{w}^{\top}\boldsymbol{x})$	随机梯度下降
支持向量机	$\mathrm{sgn}(\boldsymbol{w}^{\top}\boldsymbol{x})$	$\max(0, 1 - y\boldsymbol{w}^{\top}\boldsymbol{x})$	二次规划、SMO 等

不同损失函数的对比

损失函数measure样本的输出与实际tag的误差
perception中为max(0,-yf(x;w))

- 为方便比较，定义类别标签 $y \in \{+1, -1\}$，并定义$f(\boldsymbol{x};\ \boldsymbol{w}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b$。
- 对于样本 $(\boldsymbol{x}, y)$，若 $yf(\boldsymbol{x};\ \boldsymbol{w}) > 0$，则分类正确，并且 $yf(\boldsymbol{x};\ \boldsymbol{w})$ 越大，模型预测越正确；反之若 $yf(\boldsymbol{x};\ \boldsymbol{w}) < 0$，则分类错误，且越小越错误。
- **一个好的损失函数应该随着$yf(\boldsymbol{x};\ \boldsymbol{w})$ 的增大而减小。**



$$\mathcal{L}_{LR} = \log\left(1 + \exp\left(-yf(\boldsymbol{x};\boldsymbol{w})\right)\right).$$ LR回归

$$\mathcal{L}_{p} = \max\left(0, -yf(\boldsymbol{x};\boldsymbol{w})\right).$$ 感知机

$$\mathcal{L}_{hinge} = \max\left(0, 1 - yf(\boldsymbol{x};\boldsymbol{w})\right).$$ SVM

$$\mathcal{L}_{squared} = \left(1 - yf(\boldsymbol{x};\boldsymbol{w})\right)^2.$$ 平方损失

无监督学习

无监督学习（Unsupervised Learning）

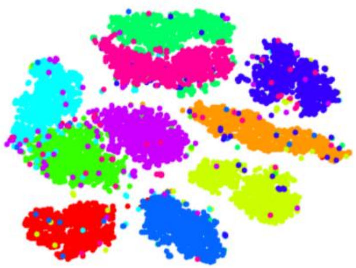
- 监督学习
  - 建立输入与输出之间的映射关系 $f: x \rightarrow y$
- 无监督学习
  - 指从无标签的数据中学习出一些有用的模式。一般直接从原始数据中学习，不借助于任何人工给出标签或者反馈等指导信息。
  - 发现隐藏的数据中的有价值信息，包括有效的特征、类别、结构以及概率分布等

为什么要无监督学习?

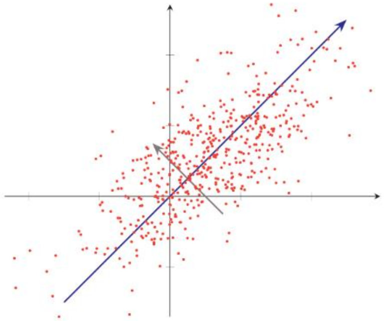
大脑有大约 10^{14} 个突触，我们只能活大约 10^9 秒. 所以我们有比数据更多的参数. 这启发了我们必须进行大量无监督学习的想法，因为感知输入（包括本体感受）是我们可以获得每秒 10^5 维约束的唯一途径.

——杰弗里·辛顿（Geoffrey Hinton）
2018年图灵奖获得者

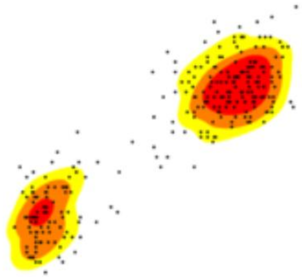
典型的无监督学习问题



聚类



无监督特征学习



概率密度估计

聚类（Clustering）是将一组样本根据一定的准则划分到不同的组（也称为簇Cluster）。一个比较通用的准则是组内样本的相似性要高于组间样本的相似性

无监督特征学习（Unsupervised Feature Learning）是从无标签训练数据中挖掘有效的特征或表示。无监督特征学习一般用来进行降维、数据可视化或监督学习前期的数据预处理

概率密度估计（Probabilistic Density Estimation）简称密度估计，是根据一组训练样本来估计样本空间的概率密度。密度估计可以分为参数密度估计和非参数密度估计

原型聚类

- 原型聚类
  - 也称为“基于原型的聚类”（prototype-based clustering)，此类算法假设聚类结构能通过一组原型刻画。
- 算法过程：
  - 通常情况下，算法先对原型进行初始化，再对原型进行迭代更新求解。
  - 采用不同的原型表示、不同的求解方式，将产生不同的算法。

原型聚类：K均值

➢ 优化问题：$\min_{\mathbf{T},\boldsymbol{\mu}} E(\boldsymbol{T},\boldsymbol{\mu}) = \|\boldsymbol{X}-\boldsymbol{T}\boldsymbol{\mu}\|_F^2$

算法流程（迭代优化）：

初始化每个簇的均值向量（共k个簇）

repeat

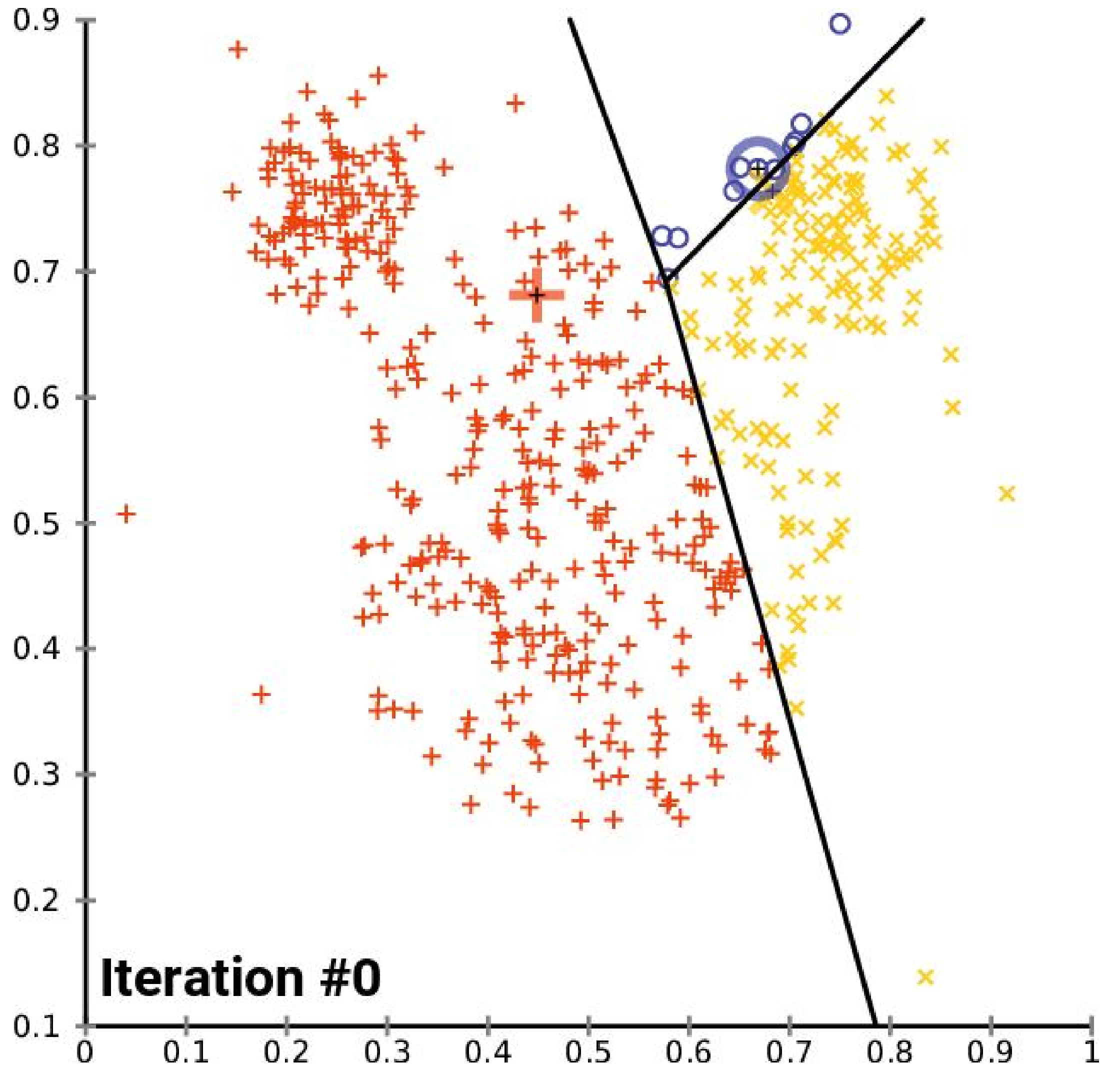
1. 将每个样本分配给最近的簇；
2. 计算每个簇的均值向量

until 当前均值向量均未更新

$$\boldsymbol{T}^{(t)} \leftarrow \min_{\boldsymbol{T}} E\left(\boldsymbol{T},\boldsymbol{\mu}^{(t-1)}\right)$$

$$\boldsymbol{\mu}^{(t)} \leftarrow \min_{\boldsymbol{\mu}} E\left(\boldsymbol{T}^{(t)},\boldsymbol{\mu}\right)$$

原型聚类：K均值



主成份分析PCA

➢一种最常用的数据降维方法，使得在转换后的空间中数据的方差最大。

- 样本点 $\boldsymbol{x}^{(n)}$投影之后的表示为

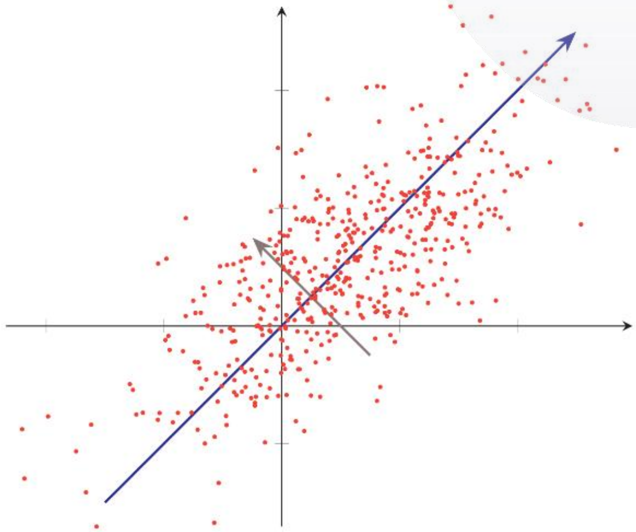
$$z^{(n)} = \boldsymbol{w}^{\top}\boldsymbol{x}^{(n)}$$

- 所有样本投影后的方差为

$$\begin{aligned}\sigma(\boldsymbol{X};\boldsymbol{w}) &= \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{w}^{\top}\boldsymbol{x}^{(n)} - \boldsymbol{w}^{\top}\bar{\boldsymbol{x}})^2 \\ &= \frac{1}{N}(\boldsymbol{w}^{\top}\boldsymbol{X} - \boldsymbol{w}^{\top}\bar{\boldsymbol{X}})(\boldsymbol{w}^{\top}\boldsymbol{X} - \boldsymbol{w}^{\top}\bar{\boldsymbol{X}})^{\top} \\ &= \boldsymbol{w}^{\top}\boldsymbol{\Sigma}\boldsymbol{w},\end{aligned}$$

- 目标函数

$$\max_{\boldsymbol{w}} \boldsymbol{w}^{\top}\boldsymbol{\Sigma}\boldsymbol{w} + \lambda(1 - \boldsymbol{w}^{\top}\boldsymbol{w})$$



- 对目标函数求导并令导数等于 0 ，可得

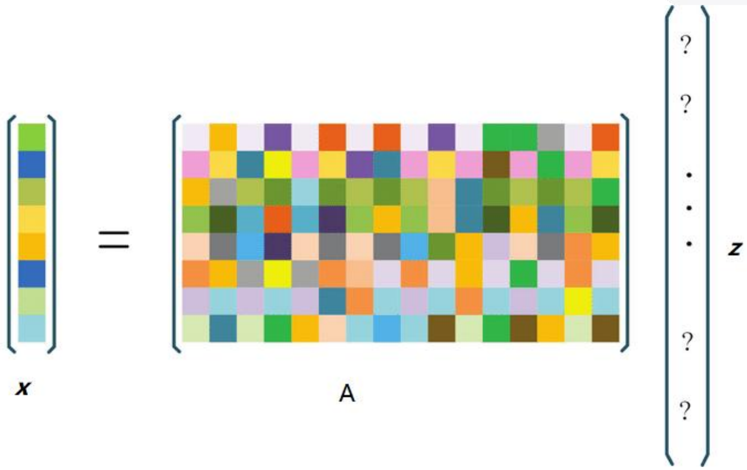
$$\boldsymbol{\Sigma}\boldsymbol{w} = \lambda\boldsymbol{w}$$

（线性）编码

- 给定一组基向量$A = [\boldsymbol{a}_1, \cdots, \boldsymbol{a}_M]$，将输入样本$\boldsymbol{x}$ 表示为这些基向量的线性组合

$$
\begin{aligned}
\boldsymbol{x} &= \sum_{m=1}^{M} z_m \boldsymbol{a}_m \\
&= A\boldsymbol{z},
\end{aligned}
$$

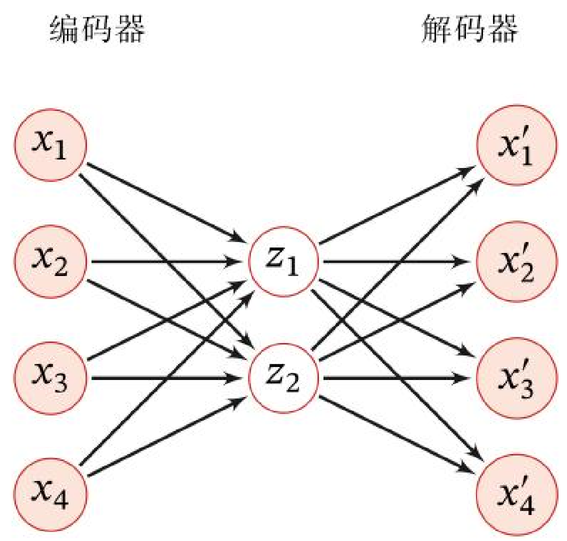
字典（dictionary）　　编码（encoding）



自编码器（Auto-Encoder）

- 编码器（Encoder） $f : \mathbb{R}^D \to \mathbb{R}^M$
- 解码器（Decoder） $g : \mathbb{R}^M \to \mathbb{R}^D$
- 学习目标：最小化重构错误

$$
\begin{aligned}
\mathcal{L} &= \sum_{n=1}^{N} \| \boldsymbol{x}^{(n)} - g(f(\boldsymbol{x}^{(n)})) \|^2 \\
&= \sum_{n=1}^{N} \| \boldsymbol{x}^{(n)} - f \circ g(\boldsymbol{x}^{(n)}) \|^2 .
\end{aligned}
$$



两层网络结构的自编码器

概率密度估计

- 参数密度估计（Parametric Density Estimation）
  - 根据先验知识假设随机变量服从某种分布，然后通过训练样本来估计分布的参数.
  - 估计方法：最大似然估计

$$\log p(\mathcal{D};\theta) = \sum_{n=1}^{N} \log p(\boldsymbol{x}^{(n)};\theta).$$

对数似然函数

$$\theta^{ML} = \arg\max_{\theta} \sum_{n=1}^{N} \log p(\boldsymbol{x}^{(n)};\theta).$$

参数估计问题转化为最优化问题

- 非参数密度估计（Nonparametric Density Estimation）
  - 不假设数据服从某种分布，通过将样本空间划分为不同的区域并估计每个区域的概率来近似数据的概率密度函数。

参数密度估计

- 正态分布

假设样本 $\boldsymbol{x} \in \mathbb{R}^D$ 服从正态分布

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\top}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right),$$

其中 $\boldsymbol{\mu}$ 和 $\boldsymbol{\Sigma}$ 分别为正态分布的均值和方差.

数据集 $\mathcal{D} = \{\boldsymbol{x}^{(n)}\}_{n=1}^{N}$ 的对数似然函数为

$$\log p(\mathcal{D}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\frac{N}{2}\log\left((2\pi)^D|\boldsymbol{\Sigma}|\right) - \frac{1}{2}\sum_{n=1}^{N}(\boldsymbol{x}^{(n)}-\boldsymbol{\mu})^{\top}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}^{(n)}-\boldsymbol{\mu}).$$

分别求上式关于 $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ 的偏导数,并令其等于0. 可得,

$$\boldsymbol{\mu}^{ML} = \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}^{(n)},$$

$$\boldsymbol{\Sigma}^{ML} = \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{x}^{(n)}-\boldsymbol{\mu}^{ML})(\boldsymbol{x}^{(n)}-\boldsymbol{\mu}^{ML})^{\top}.$$

参数密度估计

- 多项分布

数据集 $\mathcal{D} = \{\boldsymbol{x}^{(n)}\}_{n=1}^{N}$ 的对数似然函数为

$$\log p(\mathcal{D}|\boldsymbol{\mu}) = \sum_{n=1}^{N}\sum_{k=1}^{K} x_k^{(n)} \log(\mu_k). \tag{9.34}$$

多项分布的参数估计为约束优化问题. 引入拉格朗日乘子 λ, 将原问题转换为无约束优化问题.

speech and language processing

$$\max_{\mu,\lambda} \quad \sum_{n=1}^{N}\sum_{k=1}^{K} x_k^{(n)} \log(\mu_k) + \lambda\Big(\sum_{k=1}^{K} \mu_k - 1\Big). \tag{9.35}$$

分别求上式关于 μ_k, λ 的偏导数, 并令其等于 0. 可得,

$$\mu_k^{ML} = \frac{m_k}{N}, \qquad 1 \le k \le K \tag{9.36}$$

其中 $m_k = \sum_{n=1}^{N} x_k^{(n)}$ 为数据集中取值为第 k 个状态的样本数量.

参数密度估计一般存在以下问题

- 模型选择问题
  - 如何选择数据分布的密度函数?
  - 实际数据的分布往往是非常复杂的，而不是简单的正态分布或多项分布。
- 不可观测变量问题
  - 即我们用来训练的样本只包含部分的可观测变量，还有一些非常关键的变量是无法观测的，这导致我们很难准确估计数据的真实分布。
- 维度灾难问题
  - 高维数据的参数估计十分困难
  - 随着维度的增加，估计参数所需要的样本数量指数增加。在样本不足时会出现过拟合。

非参数密度估计

- 对于高维空间中的一个随机向量$\boldsymbol{x}$，假设其服从一个未知分布$p(\boldsymbol{x})$，则$\boldsymbol{x}$落入空间中的小区域 $\mathcal{R}$ 的概率为

$$P = \int_{\mathcal{R}} p(\boldsymbol{x}) d\boldsymbol{x}.$$

- 给定N个训练样本$D = \{\boldsymbol{x}^{(n)}\}_{n=1}^{N}$,落入区域 $\mathcal{R}$ 的样本数量K服从二项分布

$$P_K = \binom{N}{K} P^K (1-P)^{1-K},$$

- 当N 非常大时，我们可以近似认为

$$P \approx p(\boldsymbol{x}) V$$

- 假设区域R足够小，其内部的概率密度是相同的，则有

$$P \approx \frac{K}{N}$$

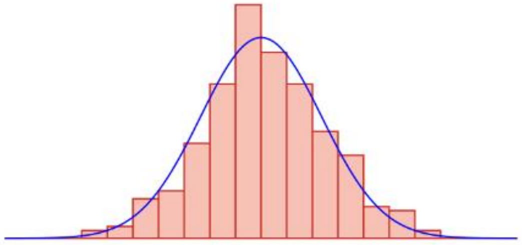
- 结合上述两个公式，得到

$$p(\boldsymbol{x}) \approx \frac{K}{NV}$$

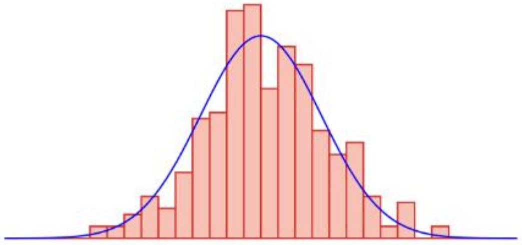
直方图方法（Histogram Method）

- 一种非常直观的估计连续变量密度函数的方法，可以表示为一种柱状图。
  - 以一维随机变量为例，首先将其取值范围分成 M 个连续的、不重叠的区间，每个区间的宽度为 Δ_m．给定 N 个训练样本$\mathcal{D}=\{x^{(n)}\}_{n=1}^{N}$，我们统计这些样本落入每个区间的数量$K_m$，然后将它们归一化为密度函数．

$$p_m = \frac{K_m}{N\Delta_m}, \qquad 1 \leq m \leq M$$



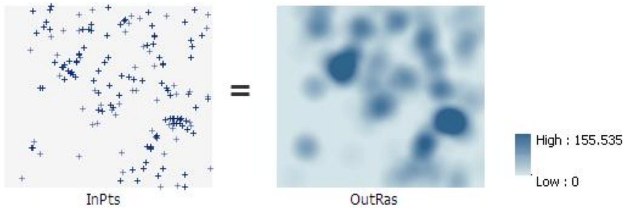
(a) 10个区间（bin）



(b) 30个区间（bin）

核密度估计（Kernel Density Estimation）

- 核密度估计，也叫Parzen窗方法，是一种直方图方法的改进。



- 假设 $\mathcal{R}$ 为d维空间中的一个以点 $\boldsymbol{x}$ 为中心的“超立方体”，并定义核函数来表示一个样本 $\boldsymbol{z}$ 是否落入该超立方体中。其中 H 为超立方体的边长，也称为核函数的宽度

$$\phi\left(\frac{\boldsymbol{z}-\boldsymbol{x}}{H}\right)=\begin{cases}1 & \text{if } |z_i-x_i|<\frac{H}{2}, 1\le i\le D\\ 0 & \text{else}\end{cases}$$

- 点 $\boldsymbol{x}$ 的密度估计为

$$p(\boldsymbol{x})=\frac{K}{NH^D}=\frac{1}{NH^D}\sum_{n=1}^{N}\phi\left(\frac{\boldsymbol{x}^{(n)}-\boldsymbol{x}}{H}\right),$$

K近邻方法

- 核密度估计方法中的核宽度 H 是固定的，因此同一个宽度可能对高密度的区域过大，而对低密度区域过小。
- 一种更灵活的方式是设置一种可变宽度的区域，并使得落入每个区域中样本数量为固定的K。
- 要估计点 $\boldsymbol{x}$ 的密度，首先找到一个以 $\boldsymbol{x}$ 为中心的球体，使得落入球体的样本数量为K，就可以计算出点 $\boldsymbol{x}$ 的密度。
- 因为落入球体的样本也是离 $\boldsymbol{x}$ 最近的K个样本，所以这种方法称为K近邻方法。