第六章 逻辑斯蒂回归

刘杰
人工智能学院

南开大学
Nankai University

Generative V.S. Discriminative

Task: to distinguish dog images from cat images.


区分模型：尝试去学习区分类别。所以也许所有的训练数据中的狗都戴着项圈和项圈猫不是。如果这一个特征能很好地将类，模型是满意的。

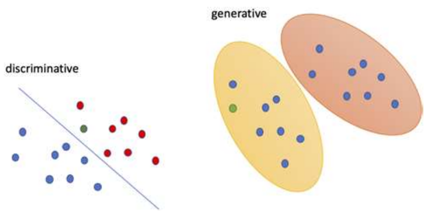
如果你问这样的模型，它对猫的了解，它能说的就是猫不戴项圈。而不能告诉你猫是什么样子。

Generative V.S. Discriminative

Task: to distinguish dog images from cat images.


生成模型目标是了解狗的长相猫长什么样。
“狗”模型能够“生成”狗。“猫”模型能够“生成”猫。
对进行测试然后，系统会询问是猫模型还是狗模型更好能生成样本里的形象，则选择某个模型作为它的标签。

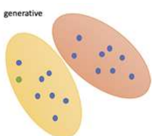
Generative and Discriminative models



生成性模型

Naïve Bayes

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} \overbrace{P(d|c)}^{\text{likelihood}} \overbrace{P(c)}^{\text{prior}}$$



A **generative model** like naive Bayes makes use of this **likelihood** term, which expresses how to generate the features of a document *if we knew it was of class c.*

Naïve Bayes recap

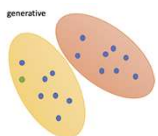
- Define $\mathrm{p}(\boldsymbol{x}, \boldsymbol{y})$ via a *generative model*
- Prediction: $\hat{y} = \arg\max_y \mathrm{p}(\boldsymbol{x}_i, y)$
- Learning:

$$\boldsymbol{\theta} = \arg\max_{\boldsymbol{\theta}} \mathrm{p}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta})$$

$$\mathrm{p}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}) = \prod_i \mathrm{p}(\boldsymbol{x}_i, y_i; \boldsymbol{\theta}) = \prod_i \mathrm{p}(\boldsymbol{x}_i | y_i) \mathrm{p}(y_i)$$

$$\phi_{y,j} = \frac{\sum_{i:Y_i=y} x_{ij}}{\sum_{i:Y_i=y} \sum_j x_{ij}}$$

$$\mu_y = \frac{\text{count}(Y = y)}{N}$$

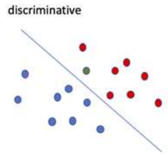


This gives the maximum likelihood estimator (MLE; same as relative frequency estimator)

Discriminative Model

By contrast a **discriminative model** in this text categorization scenario attempts to **directly** compute $P(c/d)$.

Perhaps it will learn to assign a high weight to document features that directly improve its ability to *discriminate* between possible classes, even if it couldn’t generate an example of one of the classes.



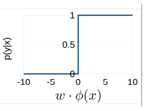
The Perceptron

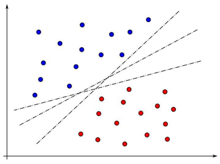
The perceptron algorithm

- finds a separating hyperplane, if it exists;
- but it seems not a probabilistic model of P(y|x)

In other words:

$$P(y=1|x)=1 \text{ if } \boldsymbol{w}\cdot\boldsymbol{\varphi}(x)\geq 0$$
$$P(y=1|x)=0 \text{ if } \boldsymbol{w}\cdot\boldsymbol{\varphi}(x)<0$$





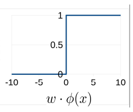
The Perceptron

The perceptron algorithm

- finds a separating hyperplane, if it exists;
- but it seems not a probabilistic model of P(y|x)

In other words:

$$P(y=1|x)=1 \text{ if } \boldsymbol{w}\cdot\boldsymbol{\varphi}(x)\geq 0$$
$$P(y=1|x)=0 \text{ if } \boldsymbol{w}\cdot\boldsymbol{\varphi}(x)<0$$



通过sign()强行对p(y|x)进行赋值
最好能有一个平滑的概率函数进行建模

我们希望 z 可以是一个合法的概率

$$z = w\cdot x+b$$

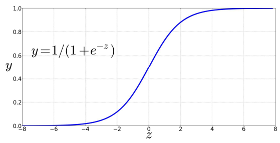
然而z 不在0和1之间，甚至可以为负
z的取值(-∞， ∞)

Perceptron & Probabilities

Sigmoid 函数

$$y=\sigma(z)=\frac{1}{1+e^{-z}}$$

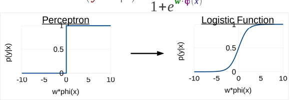
优点：
值域[0,1]，满足概率取值要求
可微可导，对学习算法友好



Perceptron & Probabilities

If we want a probability P(y|x)

$$P(y=1|x)=\frac{e^{\boldsymbol{w}\cdot\boldsymbol{\varphi}(x)}}{1+e^{\boldsymbol{w}\cdot\boldsymbol{\varphi}(x)}}$$



- “Softer” function than in perceptron
- Can account for uncertainty
- Differentiable

逻辑斯蒂分布

Logistic distribution

- 设X是连续随机变量，X服从Logistic distribution，
- 分布函数：$F(x)=P(X\leq x)=\dfrac{1}{1+e^{-\frac{x-\mu}{\gamma}}}$
- 密度函数：$f(x)=F'(x)=\dfrac{e^{-\frac{x-\mu}{\gamma}}}{\gamma(1+e^{-\frac{x-\mu}{\gamma}})^2}$
- μ为位置参数，γ大于0为形状参数，
  (μ,1/2)中心对称

$$F(-x+\mu)-\frac{1}{2}=-F(x-\mu)+\frac{1}{2}$$

二项逻辑斯蒂回归

Binomial logistic regression model

- 由条件概率P(Y|X)表示的分类模型
- 形式为参数化的logistic distribution
- X取实数，Y取值1,0

$$P(Y=1|x)=\frac{e^{w\cdot x+b}}{1+e^{w\cdot x+b}} \qquad P(Y=0|x)=\frac{1}{1+e^{w\cdot x+b}}$$

$$\Rightarrow \quad P(Y=1|x)=\frac{e^{w\cdot x}}{1+e^{w\cdot x}} \qquad P(Y=0|x)=\frac{1}{1+e^{w\cdot x}}$$

$$w=(w^{(1)},w^{(2)},...,w^{(n)},b)^T$$
$$x=(x^{(1)},x^{(2)},...,x^{(n)},1)^T$$

二项逻辑斯蒂回归

事件的几率odds：事件发生与事件不发生的概率之比为

$$\frac{p}{1-p}$$

称为事件的发生比(the odds of experiencing an event),

对数几率：

$$logit(p)=\log\frac{p}{1-p}$$

对数逻辑斯蒂回归：

$$\log\frac{P(Y=1|x)}{1-P(Y=1|x)}=w\cdot x$$

似然函数

- *logistic*分类器是由一组权值系数组成的，最关键的问题就是如何获取这组权值，通过极大似然函数估计获得，并且*Y~f(x;w)*
- 似然函数是统计模型中参数的函数。给定输出x时，关于参数θ的似然函数L(θ|x)（在数值上）等于给定参数θ后变量X的概率：L(θ|x)=P(X=x|θ)
- 似然函数的重要性不是它的取值，而是当参数变化时概率密度函数到底是变大还是变小。
- 极大似然函数：似然函数取得最大值表示相应的参数能够使得统计模型最为合理

似然函数

- 那么对于上述*N*个观测事件，设

$$P(Y=1|x)=\pi(x),P(Y=0|x)=1-\pi(x)$$

- 其联合概率密度函数，即似然函数为：

$$\prod_{i=1}^{N}[\pi(x_i)]^{y_i}[1-\pi(x_i)]^{1-y_i}$$

- 目标：求出使这一似然函数的值最大的参数估，$w_1,w_2,...,w_n$，使得L(w)取得 最大值。
- 对L(w)取对数：

模型参数估计

对数似然函数

$$\begin{aligned}L(w)&=\sum_{i=1}^{N}[y_i\log\pi(x_i)+(1-y_i)\log(1-\pi(x_i))]\\&=\sum_{i=1}^{N}[y_i\log\frac{\pi(x_i)}{1-\pi(x_i)}+\log(1-\pi(x_i))]\\&=\sum_{i=1}^{N}[y_i(w\cdot x_i)-\log(1+e^{w\cdot x_i})]\end{aligned}$$

对L(w)求极大值，得到w的估计值。

通常采用梯度下降法及拟牛顿法，学到的模型：

$$P(Y=1|x)=\frac{e^{\hat{w}\cdot x}}{1+e^{\hat{w}\cdot x}} \qquad P(Y=0|x)=\frac{1}{1+e^{\hat{w}\cdot x}}$$

多项logistic回归

设Y的取值集合为

$$\{1,2,...,K\}$$

多项logistic回归模型

$$P(Y=k|x)=\frac{e^{w_k\cdot x}}{1+\sum_{k=1}^{K-1}e^{w_k\cdot x}},k=1,2,...,K-1$$

$$P(Y=K|x)=\frac{1}{1+\sum_{k=1}^{K-1}e^{w_k\cdot x}}$$

最大熵模型

- 最大熵模型(Maximum Entropy Model)由最大熵原理推导实现。
- 最大熵原理：
- 学习概率模型时，在所有可能的概率模型(分布)中，熵最大的模型是最好的模型，表述为在满足约束条件的模型集合中选取熵最大的模型。
- 假设离散随机变量X的概率分布是P(X)，
- 熵： $H(P) = -\sum_x P(x) \log P(x)$
- 且： $0 \le H(P) \le \log |X|$
- |X|是X的取值个数，X均匀分布时右边等号成立。

例子：

- 假设随机变量X有5个取值{A,B,C,D,E},估计各个值的概率。
- 解：满足 $P(A)+P(B)+P(C)+P(D)+P(E)=1$
- 等概率估计： $P(A)=P(B)=P(C)=P(D)=P(E)=\frac{1}{5}$
- 加入一些先验： $P(A)+P(B)=\frac{3}{10}$

  $P(A)+P(B)+P(C)+P(D)+P(E)=1$
- 于是： $P(A)=P(B)=\frac{3}{20}$

  $P(C)=P(D)=P(E)=\frac{7}{30}$

例子：

- 假设随机变量X有5个取值{A,B,C,D,E},估计各个值的概率。
- 解：满足 $P(A)+P(B)+P(C)+P(D)+P(E)=1$
- 等概率估计： $P(A)=P(B)=P(C)=P(D)=P(E)=\frac{1}{5}$
- 加入一些先验： $P(A)+P(B)=\frac{3}{10}$

  $P(A)+P(B)+P(C)+P(D)+P(E)=1$
- 于是： $P(A)=P(B)=\frac{3}{20}$ 再加入约束： $P(A)+P(C)=\frac{1}{2}$

  $P(C)=P(D)=P(E)=\frac{7}{30}$ $P(A)+P(B)=\frac{3}{10}$

  $P(A)+P(B)+P(C)+P(D)+P(E)=1$

最大熵模型

- X和Y分别是输入和输出的集合，这个模型表示的是对于给定的输入X，以条件概率P(Y|X)输出Y.
- 给定数据集：

$$T = \{(x_1,y_1),(x_2,y_2),...,(x_N,y_N)\}$$

- 联合分布P(Y|X)的经验分布，边缘分布P(X)的经验分布：

$$\tilde{P}(X,Y) \rightarrow \tilde{P}(X=x,Y=y)=\frac{V(X=x,Y=y)}{N}$$

$$\tilde{P}(X) \rightarrow \tilde{P}(X=x)=\frac{V(X=x)}{N}$$

- 特征函数： $f(x,y)=\begin{cases}1, & \text{x与y满足某一事实} \\ 0, & \text{否则}\end{cases}$

最大熵模型

- 特征函数f(x,y)关于经验分布 $\tilde{P}(X,Y)$ 的期望值:

$$E_{\tilde{P}}(f)=\sum_{x,y}\tilde{P}(x,y)f(x,y)$$

- 特征函数f(x,y)关于模型P(Y|X)与经验分布 $\tilde{P}(X)$ 的期望值:

$$E_{\tilde{P}}(f)=\sum_{x,y}\tilde{P}(x)P(y|x)f(x,y)$$

- 如果模型能够获取训练数据中的信息，那么就可以假设这两个期望值相等，即

$$E_P(f)=E_{\tilde{P}}(f) \longrightarrow \sum_{x,y}\tilde{P}(x)P(y|x)f(x,y)=\sum_{x,y}\tilde{P}(x,y)f(x,y)$$

- 假设有n个特征函数:

$$f_i(x,y), i=1,2,...,n$$

最大熵模型

- 定义:
- 假设满足所有约束条件的模型集合为:

$$\mathcal{C} \equiv \{P \in \mathcal{P} | E_P(f_i)=E_{\tilde{P}}(f_i), i=1,2,...,n\}$$

- 定义在条件概率分布P(Y|X)上的条件熵:

$$H(P)=-\sum_{x,y}\tilde{P}(x)P(y|x)\log P(y|x)$$

- 则模型集合C中条件熵H(P)最大的模型称为最大熵模型

最大熵模型的学习

- 最大熵模型的学习可以形式化为约束最优化问题。
- 对于给定的数据集以及特征函数：$f_i(x,y)$
- 最大熵模型的学习等价于约束最优化问题:

$$\max_{P\in\mathcal{C}} H(P) = -\sum_{x,y}\tilde{P}(x)P(y|x)\log P(y|x) \qquad \min_{P\in\mathcal{C}} -H(P) = \sum_{x,y}\tilde{P}(x)P(y|x)\log P(y|x)$$

$$s.t.\ E_P(f_i) = E_{\tilde{P}}(f_i), i=1,2,\dots,n \qquad s.t.\ E_P(f_i) - E_{\tilde{P}}(f_i) = 0, i=1,2,\dots,n$$

$$\sum_y P(y|x) = 1 \qquad \sum_y P(y|x) - 1 = 0$$

最大熵模型的学习

- 这里，将约束最优化的原始问题转换为无约束最优化的对偶问题,通过求解对偶问题求解原始问题:
- 引进拉格朗日乘子，定义拉格朗日函数:

$$\begin{aligned} L(P,w) &= -H(P) + w_0(1-\sum_y P(y|x)) + \sum_{i=1}^n w_i(E_{\tilde{P}}(f_i) - E_P(f_i)) \\ &= \sum_{x,y}\tilde{P}(x)P(y|x)\log P(y|x) + w_0(1-\sum_y P(y|x)) \\ &+ \sum_{i=1}^n w_i(\sum_{x,y}\tilde{P}(x,y)f_i(x,y) - \sum_{x,y}\tilde{P}(x)P(y|x)f_i(x,y)) \end{aligned}$$

- 最优化原始问题可表示为:

$$\min_{P\in\mathcal{C}}\max_w L(P,w)$$

最大熵模型的学习

- 最优化原始问题 到 对偶问题:

$$\min_{P\in\mathcal{C}}\max_w L(P,w) \longrightarrow \max_w \min_{P\in\mathcal{C}} L(P,w)$$

- L(P,w)是P的凸函数，解的等价性（证明部分在SVM部分介绍）
- 先求极小化问题：$\min_{P\in\mathcal{C}} L(P,w)$ 是w的函数,

$$\Psi(w) = \min_{P\in\mathcal{C}} L(P,w) = L(P_w, w)$$

$$P_w = \arg\min_{P\in\mathcal{C}} L(P,w) = P_w(y|x)$$

最大熵模型的学习

- 求L(P,w)对P(y|x)的偏导数:

$$\begin{aligned} \frac{\partial L(P,w)}{\partial P(y|x)} &= \sum_{x,y}\tilde{P}(x)(\log P(y|x)+1) - \sum_y w_0 - \sum_{x,y}(\tilde{P}(x)\sum_{i=1}^n w_i f_i(x,y)) \\ &= \sum_{x,y}\tilde{P}(x)(\log P(y|x)+1-w_0-\sum_{i=1}^n w_i f_i(x,y)) \end{aligned}$$

- 令偏导数为零，得:

$$P(y|x) = \exp(\sum_{i=1}^n w_i f_i(x,y) + w_0 - 1) = \frac{\exp(\sum_{i=1}^n w_i f_i(x,y))}{\exp(1-w_0)}$$

最大熵模型的学习

- 由：$\sum_y P(y|x) = 1$
- 得：$P_w(y|x) = \frac{1}{Z_w(x)}\exp(\sum_{i=1}^n w_i f_i(x,y))$ (6.22)
- 规范化因子：$Z_w(x) = \sum_y \exp\sum_{i=1}^n w_i f_i(x,y)$ (6.23)
- 模型$P_w = P_w(y|x)$ 就是最大熵模型
- 求解对偶问题外部的极大化问题:

$$\max_w \Psi(w) \qquad w^* = \arg\max_w \Psi(w) \qquad P^* = P_{w^*} = P_{w^*}(y|x)$$

例子：

原例子中的最大熵模型:

$$\min\ -H(P) = \sum_{i=1}^5 P(y_i)\log P(y_i)$$

$$s.t.\ P(y_1)+P(y_2) = \tilde{P}(y_1)+\tilde{P}(y_2) = \frac{3}{10}$$

$$\sum_{i=1}^5 P(y_i) = \sum_{i=1}^5 \tilde{P}(y_i) = 1$$

$$L(P,w) = \sum_{i=1}^5 P(y_i)\log P(y_i) + w_1(P(y_1)+P(y_2)-\frac{3}{10}) + w_0(\sum_{i=1}^5 P(y_i)-1)$$

$$\max_w \min_P L(P,w)$$

例子：

$$\frac{\partial L(P,w)}{\partial P(y_1)} = 1 + \log P(y_1) + w_1 + w_0$$
$$\frac{\partial L(P,w)}{\partial P(y_2)} = 1 + \log P(y_2) + w_1 + w_0$$
$$\frac{\partial L(P,w)}{\partial P(y_3)} = 1 + \log P(y_3) + w_0$$
$$\frac{\partial L(P,w)}{\partial P(y_4)} = 1 + \log P(y_4) + w_0$$
$$\frac{\partial L(P,w)}{\partial P(y_5)} = 1 + \log P(y_5) + w_0$$

解得：
$$P(y_1) = P(y_2) = e^{-w_1 - w_0 - 1}$$
$$P(y_3) = P(y_4) = P(y_5) = e^{-w_0 - 1}$$

31

例子：

$$\min_P L(P,w) = L(P_w,w) = -2e^{-w_1-w_0-1} - 3e^{-w_0-1} - \frac{3}{10}w_1 - w_0$$

得:

$$\max_w L(P_w,w) = -2e^{-w_1-w_0-1} - 3e^{-w_0-1} - \frac{3}{10}w_1 - w_0$$

对w_i求偏导并令为0：

$$e^{-w_1-w_0-1} = \frac{3}{20} \qquad P(y_1) = P(y_2) = \frac{3}{20}$$
$$e^{-w_0-1} = \frac{7}{20} \qquad P(y_3) = P(y_4) = P(y_5) = \frac{7}{20}$$

32

极大似然估计

最大熵模型就是(6.22),(6.23)表示的条件概率分布，

证明：对偶函数的极大化等价于最大熵模型的极大似然估计.

已知训练数据的经验概率分布$\tilde{P}(X,Y)$，条件概率分布P(Y|X)的对数似然函数表示为：

$$\begin{aligned} L_{\tilde{P}}(P_w) &= \log \prod_{x,y} P(y|x)^{\tilde{P}(x,y)} = \sum_{x,y} \tilde{P}(x,y) \log P(y|x) \\ &= \sum_{x,y} \tilde{P}(x,y) \sum_{i=1}^n w_i f_i(x,y) - \sum_{x,y} \tilde{P}(x,y) \log Z_w(x) \\ &= \sum_{x,y} \tilde{P}(x,y) \sum_{i=1}^n w_i f_i(x,y) - \sum_{x} \tilde{P}(x) \log Z_w(x) \end{aligned}$$

33

极大似然估计

而：
$$\begin{aligned} \Psi(w) &= \sum_{x,y} \tilde{P}(x,y) P_w(y|x) \log P_w(y|x) \\ &\quad + \sum_{i=1}^n w_i \Big(\sum_{x,y} \tilde{P}(x,y) f_i(x,y) - \sum_{x,y} \tilde{P}(x) P_w(y|x) f_i(x,y)\Big) \\ &= \sum_{x,y} \tilde{P}(x,y) \sum_{i=1}^n w_i f_i(x,y) + \sum_{x,y} \tilde{P}(x) P_w(y|x) \Big(\log P_w(y|x) - \sum_{i=1}^n w_i f_i(x,y)\Big) \\ &= \sum_{x,y} \tilde{P}(x,y) \sum_{i=1}^n w_i f_i(x,y) - \sum_{x,y} \tilde{P}(x) P_w(y|x) \log Z_w(x) \\ &= \sum_{x,y} \tilde{P}(x,y) \sum_{i=1}^n w_i f_i(x,y) - \sum_{x,y} \tilde{P}(x) \log Z_w(x) \end{aligned}$$

34

极大似然估计

最大熵模型与逻辑斯谛回归模型有类似的形式，它们又称为对数线性模型(log linear model)。

模型学习就是在给定的训练数据条件下对模型进行极大似然估计或正则化的极大似然估计。

35

模型学习的最优化算法

逻辑斯谛回归模型、最大熵模型学习归结为以似然函数为目标函数的最优化问题，通常通过迭代算法求解，它是光滑的凸函数，因此多种最优化的方法都适用。

常用的方法有：
- 改进的迭代尺度法
- 梯度下降法
- 牛顿法
- 拟牛顿法

36