Probabilistic Models

Maximum Likelihood Estimation

假如我们有一个硬币（可能不公平），掷了 N=100 次，得到了结果 $\{x_1, ... x_N\}$，其中 $x_i \in \{0,1\}$（设 1 为正面向上），并且正面向上的次数 $N_H = 55$，$N_T = 45$。

接下来我们希望建立模型来预测下一次掷硬币的结果。

由于硬币可能不公平，我们设结果 x 是 Bernoulli random variable，设得到 1 的概率为 θ，$\theta \in \{0,1\}$。

$$p(x=1|\theta) = \theta \quad \text{and} \quad p(x=0|\theta) = 1-\theta$$
$$\text{or more succinctly} \quad p(x|\theta) = \theta^x (1-\theta)^{1-x}$$

Thus the joint probability of the outcome $\{x_1, \ldots, x_N\}$ is

$$p(x_1, ..., x_N|\theta) = \prod_{i=1}^{N} \theta^{x_i}(1-\theta)^{1-x_i}$$

注：上面是进行连乘。θ 是 0 和 1 之间的值。看到类似 (x|y) 的结构，就是知道 y，求 x，至于求 x 的什么，根据情况而定。

- We call the probability mass (or density for continuous) of the observed data the likelihood function (as a function of the parameters θ):

$$L(\theta) = \prod_{i=1}^{N} \theta^{x_i}(1-\theta)^{1-x_i}$$

- We usually work with log-likelihoods:

$$\ell(\theta) = \sum_{i=1}^{N} x_i \log\theta + (1-x_i)\log(1-\theta)$$

注：似然和概率类似。不过概率描述了已知参数时的随机变量的输出结果；似然则用来描述已知随机变量输出结果时，未知参数的可能取值。上面的似然方程就是关于未知参数 θ 的。

之后我们要选择 θ。根据 observed data，在我们取得似然函数的最大值时，对应的概率密度最大（最合理），因此我们需要 maximize likelihood：

$$\hat{\theta}_{\mathrm{ML}} = \max_{\theta \in [0,1]} \ell(\theta)$$

接下来得到 θ 的最大值：

- Remember how we found the optimal solution to linear regression by setting derivatives to zero? We can do that again for the coin example.

$$\begin{aligned}
\frac{\mathrm{d}\ell}{\mathrm{d}\theta} &= \frac{\mathrm{d}}{\mathrm{d}\theta}\left(\sum_{i=1}^{N} x_i \log\theta + (1-x_i)\log(1-\theta)\right) \\
&= \frac{\mathrm{d}}{\mathrm{d}\theta}\left(N_H \log\theta + N_T \log(1-\theta)\right) \\
&= \frac{N_H}{\theta} - \frac{N_T}{1-\theta}
\end{aligned}$$

where $N_H = \sum_i x_i$ and $N_T = N - \sum_i x_i$.

- Setting this to zero gives the maximum likelihood estimate:

$$\hat{\theta}_{\mathrm{ML}} = \frac{N_H}{N_H + N_T}.$$

同时，我们还最小化了交叉熵：

- Notice, in the coin example we are actually minimizing cross-entropies!

$$\begin{aligned}
\hat{\theta}_{\mathrm{ML}} &= \max_{\theta \in [0,1]} \ell(\theta) \\
&= \min_{\theta \in [0,1]} -\ell(\theta) \\
&= \min_{\theta \in [0,1]} \sum_{i=1}^{N} -x_i \log\theta - (1-x_i)\log(1-\theta)
\end{aligned}$$

- This is an example of maximum likelihood estimation.
  - define a model that assigns a probability (or has a probability density at) to a dataset
  - maximize the likelihood (or minimize the neg. log-likelihood).

Discriminative VS Generative

Two approaches to classification:

- Discriminative approach: estimate parameters of decision boundary/class separator directly from labeled examples.
  - Model $p(t|\mathbf{x})$ directly (logistic regression models)
  - Learn mappings from inputs to classes (linear/logistic regression, decision trees etc)
  - Tries to solve: How do I separate the classes?
- Generative approach: model the distribution of inputs characteristic of the class (Bayes classifier).
  - Model $p(\mathbf{x}|t)$
  - Apply Bayes Rule to derive $p(t|\mathbf{x})$.
  - Tries to solve: What does each class "look" like?
- Key difference: is there a distributional assumption over inputs?

两种方法：一种是直接建立决策边界的函数，不考虑分布；另一种是考虑分布进行分类。

A Generative Model: Bayes Classifier

假如我们要对邮件分类：spam = 1，not spam = 0。

- Example: "You are one of the very few who have been selected as a winners for the free $1000 Gift Card."
- Use bag-of-words features, get binary vector $\mathbf{x}$ for each email
- Vocabulary:
  - "a": 1
  - ...
  - "car": 0
  - "card": 1
  - ...
  - "win": 0
  - "winner": 1
  - "winter": 0
  - ...
  - "you": 1

注：bag-of-words 中的词如果出现在了邮件中，则值为 1，反之为 0。这些数据组成了向量 x。

- Given features $\mathbf{x} = [x_1, x_2, \cdots, x_D]^T$ we want to compute class probabilities using Bayes Rule:

$$\underbrace{p(c|\mathbf{x})}_{\text{Pr. class given words}} = \frac{p(\mathbf{x}, c)}{p(\mathbf{x})} = \frac{\overbrace{p(\mathbf{x}|c)}^{\text{Pr. words given class}} \quad p(c)}{p(\mathbf{x})}$$

上面是根据邮件中出现的词来得到邮件属于哪个类的概率。$p(c|x)$ 表示根据词求类别的概率，$p(x|c)$ 代表词出现在不同类中的概率。

- How can we compute $p(\mathbf{x})$ for the two class case? (Do we need to?)

$$p(\mathbf{x}) = p(\mathbf{x}|c=0)p(c=0) + p(\mathbf{x}|c=1)p(c=1)$$

- To compute $p(c|\mathbf{x})$ we need: $p(\mathbf{x}|c)$ and $p(c)$

Naive Bayes

贝叶斯定理假设一个属性值对给定类的影响独立于其它属性的值，而此假设在实际情况中经常是不成立的，因此我们使用朴素贝叶斯，即假设给定目标值时属性之间相互条件独立。

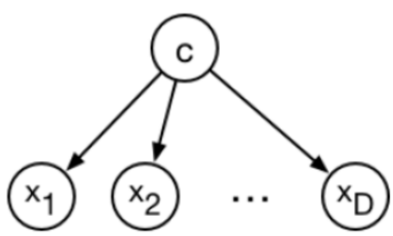
- Naïve assumption: Naïve Bayes assumes that the word features x_i are conditionally independent given the class c.
  - This means x_i and x_j are independent under the conditional distribution $p(\mathbf{x}|c)$.
  - Note: this doesn't mean they're independent.
  - Mathematically,

$$p(c, x_1, \ldots, x_D) = p(c)p(x_1|c) \cdots p(x_D|c).$$

这样我们构建了一个 joint distribution (把类 c 加入了分布)，可以由此得到 $p(c)$ 和 $p(x|c)$ 。

Bayes Nets

- We can represent this model using an directed graphical model, or Bayesian network:



这种形式可以看作给定 (父) 变量的每个变量的条件分布的乘积。

Learning

- The parameters can be learned efficiently because the log-likelihood decomposes into independent terms for each feature.

$$
\begin{aligned}
\ell(\boldsymbol{\theta}) &= \sum_{i=1}^{N} \log p(c^{(i)}, \mathbf{x}^{(i)}) = \sum_{i=1}^{N} \log \left\{ p(\mathbf{x}^{(i)} | c^{(i)}) p(c^{(i)}) \right\} \\
&= \sum_{i=1}^{N} \log \left\{ p(c^{(i)}) \prod_{j=1}^{D} p(x_j^{(i)} \mid c^{(i)}) \right\} \\
&= \sum_{i=1}^{N} \left[\log p(c^{(i)}) + \sum_{j=1}^{D} \log p(x_j^{(i)} \mid c^{(i)}) \right] \\
&= \underbrace{\sum_{i=1}^{N} \log p(c^{(i)})}_{\substack{\text{Bernoulli log-likelihood} \\ \text{of labels}}} + \underbrace{\sum_{j=1}^{D} \sum_{i=1}^{N} \log p(x_j^{(i)} \mid c^{(i)})}_{\substack{\text{Bernoulli log-likelihood} \\ \text{for feature } x_j}}
\end{aligned}
$$

- Each of these log-likelihood terms depends on different sets of parameters, so they can be optimized independently.

现在可以得到 $p(c)$ 和 $p(x|c)$。我们可以分开处理它们。

首先最大化 $\sum_{i=1}^{N} \log p\left(c^{(i)}\right)$：

假设 $p\left(c^{(i)}=1\right)=\pi$，那么 $p\left(c^{(i)}\right)=\pi^{c^{(i)}}(1-\pi)^{1-c^{(i)}}$。

- Log-likelihood:

$$
\sum_{i=1}^{N} \log p(c^{(i)}) = \sum_{i=1}^{N} c^{(i)} \log \pi + \sum_{i=1}^{N} (1 - c^{(i)}) \log(1 - \pi)
$$

- Obtain MLEs by setting derivatives to zero:

$$
\hat{\pi} = \frac{\sum_i \mathbb{I}[c^{(i)} = 1]}{N} = \frac{\#\text{ spams in dataset}}{\text{total }\#\text{ samples}}
$$

注：MLE 是“最大似然估计”。

接下来是最大化 $\sum_{i=1}^{N} \log p\left(x_j^{(i)} \mid c^{(i)}\right)$。

我们假设 $\theta_{jc} = p\left(x_j^{(i)} = 1 \mid c\right)$，$\theta_{jc}$ 是第 j 个 x (词) 在类别 c 出现的似然。
$p\left(x_j^{(i)} \mid c\right) = \theta_{jc}^{x_j^{(i)}} (1-\theta_{jc})^{1-x_j^{(i)}}$。

- Log-likelihood:

$$\begin{aligned}\sum_{i=1}^{N} \log p(x_j^{(i)} \mid c^{(i)}) = & \sum_{i=1}^{N} c^{(i)} \left\{ x_j^{(i)} \log \theta_{j1} + (1 - x_j^{(i)}) \log(1-\theta_{j1}) \right\} \\ & + \sum_{i=1}^{N} (1-c^{(i)}) \left\{ x_j^{(i)} \log \theta_{j0} + (1 - x_j^{(i)}) \log(1-\theta_{j0}) \right\}\end{aligned}$$

- Obtain MLEs by setting derivatives to zero:

$$\hat{\theta}_{jc} = \frac{\sum_i \mathbb{I}[x_j^{(i)} = 1 \ \& \ c^{(i)} = c]}{\sum_i \mathbb{I}[c^{(i)} = c]} \overset{\text{for } c=1}{=} \frac{\#\text{word } j \text{ appears in spams}}{\#\text{ spams in dataset}}$$

Inference

- We predict the category by performing inference in the model.
- Apply Bayes' Rule:

$$p(c \mid \mathbf{x}) = \frac{p(c)p(\mathbf{x} \mid c)}{\sum_{c'} p(c')p(\mathbf{x} \mid c')} = \frac{p(c) \prod_{j=1}^{D} p(x_j \mid c)}{\sum_{c'} p(c') \prod_{j=1}^{D} p(x_j \mid c')}$$

如果我们只想得到最可能的类别，就不需要计算分母。

- For input $\mathbf{x}$, predict by comparing the values of $p(c) \prod_{j=1}^{D} p(x_j \mid c)$ for different c (e.g. choose the largest).

注：计算每个类的大小，取结构最大的那个类。

MLE issue: Data Sparsity

MLE 有一个问题，就是如果数据太少，它会过拟合。这叫做数据稀疏性。

Bayesian Parameter Estimation

前面介绍了贝叶斯分类器的一种参数估计方法“最大似然估计”，现在介绍另一种。它们的作用都是找到 θ。

在最大似然中，我们把 observation 看作随机变量，但参数 θ 不是。但在贝叶斯方法中，我们把参数也看作随机变量。

为了建立 Bayesian model，我们需要指定两个分布：

- 先验分布 (prior distribution) $p(\theta)$，就是先随便给参数赋值。
- 似然 $p(\mathcal{D} \mid \boldsymbol{\theta})$，就是最大似然，$\mathcal{D}$ 是类别

之后，我们根据 observations 更新后验分布 (posterior distribution)：

$$p(\boldsymbol{\theta} \mid \mathcal{D}) = \frac{p(\boldsymbol{\theta})p(\mathcal{D} \mid \boldsymbol{\theta})}{\int p(\boldsymbol{\theta}')p(\mathcal{D} \mid \boldsymbol{\theta}')\,\mathrm{d}\boldsymbol{\theta}'}$$

我们一般不计算分母。

现在让我们回到最开始的问题-掷硬币，我们已经知道了它的似然：

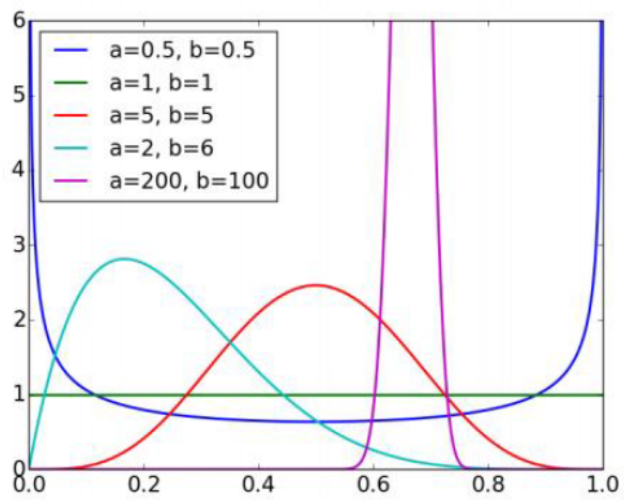
$$L(\theta) = p(\mathcal{D}|\theta) = \theta^{N_H}(1-\theta)^{N_T}$$

接下来需要指定它的先验分布 $p(\theta)$，这里我们设置了 beta distribution：

$$p(\theta; a, b) = \theta^{a-1}(1-\theta)^{b-1}$$

注：a 和 b 是参数。

- Beta distribution for various values of a, b:



- Some observations:
  - The expectation $\mathbb{E}[\theta] = a/(a+b)$ (easy to derive).
  - The distribution gets more peaked when a and b are large.
  - The uniform distribution is the special case where $a = b = 1$.

接下来计算出后验分布：

- Computing the posterior distribution:

$$\begin{aligned} p(\boldsymbol{\theta} \mid \mathcal{D}) &\propto p(\boldsymbol{\theta}) p(\mathcal{D} \mid \boldsymbol{\theta}) \\ &\propto \left[\theta^{a-1}(1-\theta)^{b-1}\right]\left[\theta^{N_H}(1-\theta)^{N_T}\right] \\ &= \theta^{a-1+N_H}(1-\theta)^{b-1+N_T}. \end{aligned}$$

- This is just a beta distribution with parameters $N_H + a$ and $N_T + b$.
- The posterior expectation of θ is:

$$\mathbb{E}[\theta \mid \mathcal{D}] = \frac{N_H + a}{N_H + N_T + a + b}$$

注：上面我们最后用期望得到了后验分布中的参数 θ。

Maximum A-Posteriori Estimation

现在我们希望用另一种方式找到后验分布中最大的参数 θ。

- This converts the Bayesian parameter estimation problem into a maximization problem

$$\begin{aligned} \hat{\boldsymbol{\theta}}_{\mathrm{MAP}} &= \arg\max_{\boldsymbol{\theta}} \; p(\boldsymbol{\theta} \mid \mathcal{D}) \\ &= \arg\max_{\boldsymbol{\theta}} \; p(\boldsymbol{\theta}, \mathcal{D}) \\ &= \arg\max_{\boldsymbol{\theta}} \; p(\boldsymbol{\theta})\, p(\mathcal{D} \mid \boldsymbol{\theta}) \\ &= \arg\max_{\boldsymbol{\theta}} \; \log p(\boldsymbol{\theta}) + \log p(\mathcal{D} \mid \boldsymbol{\theta}) \end{aligned}$$

- Joint probability in the coin flip example:

$$\begin{aligned} \log p(\theta, \mathcal{D}) &= \log p(\theta) + \log p(\mathcal{D} \mid \theta) \\ &= \text{Const} + (a-1)\log\theta + (b-1)\log(1-\theta) + N_H \log\theta + N_T \log(1-\theta) \\ &= \text{Const} + (N_H + a - 1)\log\theta + (N_T + b - 1)\log(1-\theta) \end{aligned}$$

- Maximize by finding a critical point

$$0 = \frac{\mathrm{d}}{\mathrm{d}\theta} \log p(\theta, \mathcal{D}) = \frac{N_H + a - 1}{\theta} - \frac{N_T + b - 1}{1 - \theta}$$

- Solving for θ,

$$\hat{\theta}_{\mathrm{MAP}} = \frac{N_H + a - 1}{N_H + N_T + a + b - 2}$$