计算机科学188：人工智能导论
2024年春季 笔记19

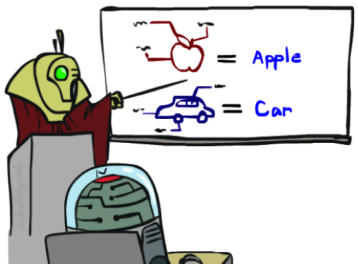
作者（其他所有笔记）：尼基尔·夏尔马
作者（贝叶斯网络笔记）：乔希·胡格和杰基·梁，由王瑞佳编辑
作者（逻辑笔记）：亨利·朱，由考佩林编辑
致谢（机器学习与逻辑笔记）：部分章节改编自教材《人工智能：一种现代方法》。

最后更新时间：2023年8月26日

机器学习

在本课程之前的几篇笔记中，我们学习了各种有助于我们在不确定性下进行推理的模型。到目前为止，我们一直假定我们所使用的概率模型是理所当然的，并且我们所使用的基础概率表的生成方法已被抽象化。当我们深入探讨机器学习时，我们将开始打破这一抽象障碍，机器学习是计算机科学的一个广泛领域，它涉及根据一些数据构建和/或学习指定模型的参数。

有许多机器学习算法，它们处理许多不同类型的问题和不同类型的数据，根据它们希望完成的任务和所处理的数据类型进行分类。机器学习算法的两个主要子类别是监督学习算法和无监督学习算法。监督学习算法推断输入数据和相应输出数据之间的关系，以便为新的、以前未见过的输入数据预测输出。另一方面，无监督学习算法的输入数据没有任何相应的输出数据，因此处理识别数据点之间或数据点内部的固有结构，并相应地对它们进行分组和/或处理。在本课程中，我们将讨论的算法将限于监督学习任务。得分。



(a) Training

(b) Validation

(c) Testing

一旦你有了准备好用于学习的数据集，机器学习过程通常包括将你的数据集分成三个不同的子集。第一个是训练数据，用于实际生成将输入映射到输出的模型。然后，验证数据（也称为留出或开发数据）用于通过对输入进行预测并生成准确率来衡量你的模型的性能

如果你的模型表现不如你期望的那样好，那么回去重新训练总是可以的，要么通过调整称为超参数的特定于模型的特殊值，要么完全使用不同的学习算法，直到你对结果满意为止。最后，使用你的模型对你的数据的第三个也是最后一个子集（测试集）进行预测。测试集是你的数据中直到开发结束才被你的智能体看到的部分，相当于一场“期末考试”，用于评估在真实世界数据上的性能。

在接下来的内容中，我们将介绍一些基础的机器学习算法，比如朴素贝叶斯、线性回归、逻辑回归和感知机算法。

朴素贝叶斯

我们将通过一个机器学习算法的具体例子来推动我们对机器学习的讨论。让我们考虑构建一个电子邮件垃圾邮件过滤器的常见问题，该过滤器将邮件分类为垃圾邮件（不需要的邮件）或正常邮件（需要的邮件）。这样的问题被称为分类问题——给定各种数据点（在这种情况下，每封电子邮件都是一个数据点），我们的目标是将它们分组到两个或更多类别中的一个。对于分类问题，我们会得到一组带有相应标签的数据点训练集，这些标签通常是几个离散值之一。

正如我们所讨论的，我们的目标是使用这个训练数据（电子邮件，以及每封邮件的垃圾邮件/正常邮件标签）来学习某种关系，以便我们可以对以前未见过的电子邮件进行预测。在本节中，我们将描述如何构建一种用于解决分类问题的模型，称为朴素贝叶斯分类器。

为了训练一个模型来将电子邮件分类为垃圾邮件或正常邮件，我们需要一些由预先分类的电子邮件组成的训练数据，以便从中学习。然而，电子邮件只是文本字符串，为了学到有用的东西，我们需要从每封邮件中提取某些被称为特征的属性。特征可以是从特定的单词计数到文本模式（例如单词是否全大写），再到你能想象到的数据的几乎任何其他属性。

为训练而提取的特定特征通常取决于你试图解决的具体问题，并且你决定选择哪些特征通常会极大地影响模型的性能。决定使用哪些特征被称为特征工程，这是机器学习的基础，但就本课程而言，你可以假设对于任何给定的数据集，你总是会得到提取的特征。在本笔记中，$\mathbf{f}\left(\mathbf{x}\right)$ 指的是在将所有输入 $\mathbf{x}$ 放入模型之前应用于它们的特征函数。

现在假设你有一个包含 n 个单词的词典，并且从每封电子邮件中提取一个特征向量 $F \in \mathbb{R}^n$，其中 F 中的 i^{th} 项是一个随机变量 F_i，它可以取值为 0 或 1，这取决于词典中的 i^{th} 个单词是否出现在正在考虑的电子邮件中。

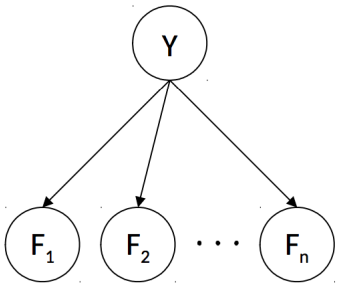
例如，如果 F_{200} 是单词“free”的特征，那么当“free”出现在邮件中时，我们将得到 $F_{200} = 1$ ，否则为0。基于这些定义，我们可以更具体地定义如何预测一封邮件是垃圾邮件还是正常邮件——如果我们能够生成每个 F_i 与标签 Y 之间的联合概率表，那么我们就可以根据邮件的特征向量计算出任何一封被考虑的邮件是垃圾邮件还是正常邮件的概率。具体来说，我们可以计算

$$P(Y = spam|F_1 = f_1, \ldots, F_n = f_n)$$

和

$$P(Y = ham|F_1 = f_1, \ldots, F_n = f_n)$$

然后根据两个概率中较高的那个来简单地标记电子邮件。不幸的是，由于我们有 n 个特征和1个标签，每个特征都可以取2个不同的值，对应于这种分布的联合概率表要求一个大小为 n 的指数级且有 2^{n+1} 个条目的表——这非常不切实际！通过用贝叶斯网络对联合概率表进行建模来解决这个问题，做出关键的简化假设：给定类别标签，每个特征 F_i 与所有其他特征相互独立。这是一个非常强的建模假设（也是朴素贝叶斯被称为朴素的原因），但它简化了推理，并且在实践中通常效果良好。它导致了以下贝叶斯网络来表示我们期望的联合概率分布。



请注意，课程前面所描述的 d 分离规则立刻表明，在这个贝叶斯网络中，给定 Y 时，每个 F_i 都与所有其他的 F_i 条件独立。现在，我们有一个用于 $P\left(Y\right)$ 的表，有2个条目，以及为每个 $P\left(F_i \mid Y\right)$ 准备的 n 个表，每个表有 $2^2 = 4$ 个条目，总共 $4n + 2$ 个条目——与 n 成线性关系！这个简化假设突出了统计效率概念所带来的权衡；为了保持在计算资源的限制范围内，我们有时需要在模型复杂性上做出妥协。

确实，在特征数量足够少的情况下，通常会对特征之间的关系做出更多假设，以生成更好的模型（这相当于给你的贝叶斯网络添加边）。对于我们采用的这个模型，对未知数据点进行预测相当于在我们的贝叶斯网络上进行推理。我们已经观察到了 $F_1, \cdots F_n$ 的值，并且想要选择在这些特征条件下具有最高概率的 Y 的值：

$$\begin{aligned} prediction(f_1, \cdots f_n) &= \operatorname*{argmax}_y P(Y = y|F_1 = f_1, \ldots F_N = f_n) \\ &= \operatorname*{argmax}_y P(Y = y, F_1 = f_1, \ldots F_N = f_n) \\ &= \operatorname*{argmax}_y P(Y = y) \prod_{i=1}^{n} P(F_i = f_i|Y = y) \end{aligned}$$

其中第一步是因为在归一化或未归一化的分布中，最高概率的类别是相同的，第二步直接来自朴素贝叶斯的独立性假设，即在给定类别标签的情况下特征是独立的（如在图形模型结构中所见）。

从垃圾邮件过滤器进行推广，现在假设存在 k 个类别标签（Y 的可能值）。此外，在注意到我们期望的概率——给定我们的特征 $P\left(Y = y_i \mid F_1 = f_1, \ldots, F_n = f_n\right)$ 时每个标签 y_i 的概率——与联合概率 $P\left(Y = y_i, F_1 = f_1, \ldots, F_n = f_n\right)$ 成比例后，我们可以计算：

$$P(Y, F_1 = f_1, \ldots, F_n = f_n) = \begin{bmatrix} P(Y = y_1, F_1 = f_1, \ldots, F_n = f_n) \\ P(Y = y_2, F_1 = f_1, \ldots, F_n = f_n) \\ \vdots \\ P(Y = y_k, F_1 = f_1, \ldots, F_n = f_n) \end{bmatrix} = \begin{bmatrix} P(Y = y_1) \prod_i P(F_i = f_i | Y = y_1) \\ P(Y = y_2) \prod_i P(F_i = f_i | Y = y_2) \\ \vdots \\ P(Y = y_k) \prod_i P(F_i = f_i | Y = y_k) \end{bmatrix}$$

对于与特征向量 F 对应的类别标签的预测，简单来说就是上述计算向量中最大值对应的标签：

$$prediction(F) = \underset{y_i}{\text{argmax}} \; P(Y = y_i) \prod_j P(F_j = f_j | Y = y_i)$$

我们现在已经了解了朴素贝叶斯分类器建模假设背后的基本理论以及如何使用它进行预测，但尚未涉及如何从输入数据中准确学习我们的贝叶斯网络中使用的条件概率表。这将不得不等待我们下一个讨论主题，参数估计。

参数估计

假设你有一组 N 样本点或观测值，$x_1, \ldots, x_N$，并且你认为这些数据是从一个由未知值 θ 参数化的分布中抽取的。换句话说，你认为每个观测值的概率 $P_\theta\left(x_i\right)$ 是 θ 的函数。例如，我们可能在抛一枚正面朝上的概率为 θ 的硬币。

根据你的样本，你如何"学习" θ 的最可能值？例如，如果我们抛了10次硬币，其中7次正面朝上，我们应该为 θ 选择什么值？这个问题的一个答案是推断 θ 等于从你假设的概率分布中选择你的样本 $x_1, \ldots, x_N$ 的概率最大化的值。机器学习中一种常用的基本方法，称为最大似然估计（MLE），正是这样做的。

最大似然估计通常做出以下简化假设：

- 每个样本都从相同的分布中抽取。换句话说，每个 x_i 都是同分布的。在我们抛硬币的例子中，每次抛硬币出现正面的概率相同，为 θ。
- 给定分布的参数，每个样本 x_i 与其他样本条件独立。这是一个很强的假设，但正如我们将看到的，它极大地有助于简化最大似然估计问题，并且在实践中通常效果良好。在抛硬币的例子中，一次抛硬币的结果不会影响其他任何一次的结果。
- 在我们看到任何数据之前，θ 的所有可能值的可能性相同（这被称为均匀先验）。

上述前两个假设通常被称为独立同分布（i.i.d.）。上述第三个假设使最大似然估计方法成为最大后验（MAP）方法的一种特殊情况，最大后验方法允许使用非均匀先验。

现在让我们定义样本的似然函数 $\mathcal{L}(\theta)$，它是一个表示从我们的分布中抽取该样本的概率的函数。对于固定样本 x_1, x_N，似然函数仅仅是 θ 的函数：

$$\mathscr{L}(\theta) = P_\theta(x_1, \ldots, x_N)$$

利用我们的简化假设，即样本 x_i 是独立同分布的，似然函数可以重新表示为
如下：

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

我们如何找到使这个函数最大化的 θ 值呢？这将是最能解释我们所观察到的数据的 θ 值。回想一下微积分知识，在函数取得最大值和最小值的点处，它关于每个输入的一阶导数（也称为函数的梯度）必须等于零。因此， θ 的最大似然估计是满足以下方程的值：

$$\frac{\partial}{\partial \theta} \mathscr{L}(\theta) = 0$$

让我们通过一个例子来使这个概念更具体。假设你有一个装满红色和蓝色球的袋子，并且不知道每种颜色的球各有多少个。你通过从袋子中取出一个球，记录颜色，然后再把球放回袋子（有放回抽样）来抽取样本。从这个袋子中抽取三个球的样本得到红红蓝。这似乎意味着我们应该推断袋子中 $\frac{2}{3}$ 的球是红色的，$\frac{1}{3}$ 的球是蓝色的。我们假设从袋子中取出的每个球为红色的概率是 θ，为蓝色的概率是 $1-\theta$，对于某个我们想要估计的值 θ （这被称为伯努利分布）：

$$P_\theta(x_i) = \begin{cases} \theta & x_i = red \\ (1-\theta) & x_i = blue \end{cases}$$

那么我们样本的似然性为：

$$\mathscr{L}(\theta) = \prod_{i=1}^{3} P_\theta(x_i) = P_\theta(x_1 = red) P_\theta(x_2 = red) P_\theta(x_2 = blue) = \theta^2 \cdot (1-\theta)$$

最后一步是将似然性的导数设为0并求解 θ：

$$\frac{\partial}{\partial \theta} \mathscr{L}(\theta) = \frac{\partial}{\partial \theta} \theta^2 \cdot (1-\theta) = \theta(2-3\theta) = 0$$

针对 θ 求解此方程可得 $\theta = \frac{2}{3}$，从直观上看这是有意义的！（还有第二个解，即 $\theta = 0$，但这对应于似然函数的一个最小值，如 $\mathcal{L}(0) = 0 < \mathcal{L}\left(\frac{2}{3}\right) = \frac{4}{27}$ 所述。）

朴素贝叶斯的最大似然估计

现在让我们回到为垃圾邮件分类器推断条件概率表的问题，首先回顾一下我们已知的变量：

- n - 我们字典中的单词数量。
- N - 用于训练的观测值（电子邮件）数量。在我们即将进行的讨论中，我们还将 N_h 定义为标记为正常邮件的训练样本数量，将 N_s 定义为标记为垃圾邮件的训练样本数量。注意 $N_h + N_s = N$。

• F_i - 一个随机变量，若正在考虑的电子邮件中出现了 i^{th} 字典中的单词，则该变量为1，否则为0。

• Y - 一个随机变量，根据相应电子邮件的标签，它要么是垃圾邮件，要么是正常邮件。

• $f_i^{(j)}$ - 这引用了训练集中 j^{th} 项中随机变量 F_i 的解析值。换句话说，如果单词 i 出现在正在考虑的 j^{th} 电子邮件中，则每个 $f_i^{(j)}$ 为1，否则为0。这是我们第一次见到这种表示法，但在即将进行的推导中它会很有用。

现在在每个条件概率表 $P\left(F_i \mid Y\right)$ 中，请注意我们有两个不同的伯努利分布： $P\left(F_i \mid Y=\text{ham}\right)$ 和 $P\left(F_i \mid Y=\text{spam}\right)$。为了简单起见，让我们具体考虑 $P\left(F_i \mid Y=\text{ham}\right)$，并尝试找到参数 $\theta = P\left(F_i = 1 \mid Y=\text{ham}\right)$ 的最大似然估计，即我们字典中的 i^{th} 词出现在垃圾邮件中的概率。由于我们的训练集中有 N_h 封垃圾邮件，我们有 N_h 个关于词 i 是否出现在垃圾邮件中的观测值。因为我们的模型假设给定其标签时每个词的出现服从伯努利分布，所以我们可以将似然函数表述为

$$\mathscr{L}(\theta) = \prod_{j=1}^{N_h} P(F_i = f_i^{(j)} | Y = ham) = \prod_{j=1}^{N_h} \theta^{f_i^{(j)}} (1-\theta)^{1-f_i^{(j)}}$$

第二步源于一个小数学技巧：若 $f_i^{(j)} = 1$，那么

$$P(F_i = f_i^{(j)} | Y = ham) = \theta^1 (1-\theta)^0 = \theta$$

类似地，若 $f_i^{(j)} = 0$，那么

$$P(F_i = f_i^{(j)} | Y = ham) = \theta^0 (1-\theta)^1 = (1-\theta)$$

为了计算 θ 的最大似然估计，回想一下，下一步是计算 $\mathcal{L}\left(\theta\right)$ 的导数并将其设为 0。尝试这样做会发现相当困难，因为隔离并求解 θ 并非易事。相反，我们将采用一种在最大似然推导中非常常见的技巧，即改为找到使似然函数的 $\log$ 最大化的 θ 值。由于 $\log\left(x\right)$ 是一个严格递增函数（有时称为单调变换），找到使 $\log \mathcal{L}\left(\theta\right)$ 最大化的值也将使 $\mathcal{L}\left(\theta\right)$ 最大化。 $\log \mathcal{L}\left(\theta\right)$ 的展开式如下：

$$\begin{aligned}
\log \mathscr{L}(\theta) &= \log\left(\prod_{j=1}^{N_h} \theta^{f_i^{(j)}} (1-\theta)^{1-f_i^{(j)}}\right) \\
&= \sum_{j=1}^{N_h} \log\left(\theta^{f_i^{(j)}} (1-\theta)^{1-f_i^{(j)}}\right) \\
&= \sum_{j=1}^{N_h} \log\left(\theta^{f_i^{(j)}}\right) + \sum_{j=1}^{N_h} \log\left((1-\theta)^{1-f_i^{(j)}}\right) \\
&= \log(\theta) \sum_{j=1}^{N_h} f_i^{(j)} + \log(1-\theta) \sum_{j=1}^{N_h} (1-f_i^{(j)})
\end{aligned}$$

请注意，在上述推导过程中，我们使用了 $\log$ 函数的性质，即 $\log\left(a^c\right) = c \cdot \log\left(a\right)$ 和 $\log\left(ab\right) = \log\left(a\right) + \log\left(b\right)$。现在我们将似然函数的 $\log$ 的导数设为 0，并求解 θ：

$$
\begin{aligned}
\frac{\partial}{\partial\theta}\left(\log(\theta)\sum_{j=1}^{N_h} f_i^{(j)} + \log(1-\theta)\sum_{j=1}^{N_h}(1-f_i^{(j)})\right) &= 0 \\
\frac{1}{\theta}\sum_{j=1}^{N_h} f_i^{(j)} - \frac{1}{(1-\theta)}\sum_{j=1}^{N_h}(1-f_i^{(j)}) &= 0 \\
\frac{1}{\theta}\sum_{j=1}^{N_h} f_i^{(j)} &= \frac{1}{(1-\theta)}\sum_{j=1}^{N_h}(1-f_i^{(j)}) \\
(1-\theta)\sum_{j=1}^{N_h} f_i^{(j)} &= \theta\sum_{j=1}^{N_h}(1-f_i^{(j)}) \\
\sum_{j=1}^{N_h} f_i^{(j)} - \theta\sum_{j=1}^{N_h} f_i^{(j)} &= \theta\sum_{j=1}^{N_h} 1 - \theta\sum_{j=1}^{N_h} f_i^{(j)} \\
\sum_{j=1}^{N_h} f_i^{(j)} &= \theta\cdot N_h \\
\theta &= \frac{1}{N_h}\sum_{j=1}^{N_h} f_i^{(j)}
\end{aligned}
$$

我们得到了一个非常简单的最终结果！根据我们上面的公式，θ 的最大似然估计（记住，它是 $P\left(F_i = 1 \mid Y = \text{ham}\right)$ 对应于计算出现单词 i 的火腿邮件数量并将其除以火腿邮件总数的概率）。你可能认为对于一个直观的结果来说这工作量很大（确实如此），但这里的推导和技术对于比我们在此为每个特征所使用的简单伯努利分布更复杂的分布将是有用的。总之，在这个具有伯努利特征分布的朴素贝叶斯模型中，在任何给定类别内，任何结果概率的最大似然估计对应于该结果的计数除以给定类别的样本总数。上述推导可以推广到我们有两个以上类别且每个特征有两个以上结果的情况，不过这里不给出此推导。

平滑

尽管最大似然估计是一种非常强大的参数估计方法，但糟糕的训练数据往往会导致不良后果。例如，如果在我们的训练集中，每次“minute”这个词出现在一封电子邮件中，那封电子邮件就被归类为垃圾邮件，我们训练好的模型就会学到

$$P(F_{minute} = 1|Y = ham) = 0$$

因此，在一封未见过的电子邮件中，如果单词“minute”出现了，$P\left(Y = \text{ham}\right)\prod_i P\left(F_i \mid Y = \text{ham}\right) = 0$，这样你的模型就永远不会将任何包含单词“minute”的电子邮件分类为垃圾邮件。这是一个过拟合的经典例子，即构建一个不能很好地推广到以前未见过的数据的模型。仅仅因为某个特定的单词没有出现在你的训练数据中的电子邮件中，并不意味着它不会出现在你的测试数据中的电子邮件或现实世界中的电子邮件中。朴素贝叶斯分类器的过拟合可以通过拉普拉斯平滑来缓解。从概念上讲，强度为 k 的拉普拉斯平滑假设每个结果都额外出现了 k 次。因此，如果对于给定的样本，你对一个可以取的结果 x 的最大似然估计

$|X|$ 来自大小为 N 的样本的不同值是

$$P_{MLE}(x) = \frac{count(x)}{N}$$

那么强度为 k 的拉普拉斯估计是

$$P_{LAP,k}(x) = \frac{count(x) + k}{N + k|X|}$$

这个方程说明了什么？我们假设每个结果都额外出现了 k 个实例，所以就好像我们看到的是 $\text{count}\,(x) + k$ 个而不是 $\text{count}\,(x)$ 个 x 的实例。类似地，如果我们看到 $|X|$ 类中的每一类都额外出现了 k 个实例，那么我们必须在原始样本数量 N 的基础上加上 $k\,|X|$ 。这两个陈述共同得出了上述公式。对于计算条件概率的拉普拉斯估计（这对于计算不同类别的结果的拉普拉斯估计很有用），也有类似的结果：

$$P_{LAP,k}(x|y) = \frac{count(x, y) + k}{count(y) + k|X|}$$

拉普拉斯平滑有两个特别值得注意的情况。第一种情况是当 $k = 0$ 时，那么 $P_{LAP,0}\,(x) = P_{MLE}\,(x)$ 。第二种情况是 $k = \infty$ 的情形。观察到每个结果出现的次数非常多甚至无穷多，会使你实际样本的结果变得无关紧要，因此你的拉普拉斯估计意味着每个
结果出现的可能性相等。实际上：如下所示：

$$P_{LAP,\infty}(x) = \frac{1}{|X|}$$

在你的模型中适合使用的 k 的具体值通常通过反复试验来确定。 k 是你模型中的一个超参数，这意味着你可以将它设置为任何你想要的值，然后看看哪个值在你的验证数据上产生最佳的预测准确性/性能。