MLAPP-C8

8.逻辑回归

8.1 简介

一种建立概率性质分类器的方式是对联合概率模型进行建模，然后以观察到的特征向量x为条件，推导得到类后验分布。这被称为生成式分类器（generative classifier），因为它针对每个类别y指定了生成样本x的方式。更多细节可以参考3.5和4.2节。另一个更加著名的方式，也就是我们在本章将要讨论的，就是训练一个形似的模型，该模型直接将特征x映射到类别y。这被称为判别式分类器（discriminative classifier），因为它在不同的类别之间进行判别，但不能生成每个类别下的样本。我们会在8.6节对两种方式进行对比。

本章，我们将集中讨论与参数呈线性关系的判别式分类器，它是第7章回归模型向分类问题的拓展，这将显著简化模型的训练，这一点我们会在后面看到。在后面的章节中，我们会讨论非线性和非参数判别模型。

8.2 模型描述

正如我们在1.4.6节所讨论的，逻辑回归对应下面的二分类模型：

（8.1）

图1.19（b）展示了在一维空间中的情况。逻辑回归可以很容易地拓展到高维输入的情况。举例来说，图8.1中，展示了在2维空间中不同的权重向量w下的图形。如果我们以0.5为阈值，我们将得到一个线性决策边界，其法线由w决定。

逻辑回归是最著名的分类器，主要基于以下几个原因：

逻辑回归模型比较容易训练，我们在8.3节将讨论这一点。这里说的“容易”是指训练模型的算法很容易实现，而且很快。特别地，存在方法使得算法的时间与非零数据集的数量成线性关系，这是可能使用的最短时间，因为算法需要“接触”每个数据至少一次。相反，更加复杂的分类器（比如SVMs，神经网络，决策树等等）的训练过程往往更慢。

逻辑回归模型更具备可解释性。特别地，让我们定义对数几率（log odds）: 。结果表明，这是一个与参数w呈线性关系的函数，即：。为了说明这一点的重要性，让我们考虑一个简单的例子。假设我们有一个特征向量，其第一个分量为你每天抽香烟的数量，第二个分量为你每天锻炼的分钟数。我们的目标是预测你得肺癌的概率。如果参数的估计值为，这就意味着你每天多抽一支烟，得肺癌风险的增长比例为。

逻辑回归模型很容易拓展到多分类问题，我们将在8.3.7节讨论，以及其他类型的输出数据，比如计数（见9.3节）

通过使用核函数（见14.3节）或者从数据中学习特征（见16章），逻辑回归可以很容易地拓展到解决那些具备非线性决策边界的问题。

8.3 模型训练

本节，我们将讨论训练逻辑回归模型的算法。

8.3.1 MLE

逻辑回归模型的负对数似然函数为：



其中。这被称为交叉熵（cross-entropy）误差函数（见2.8.2节）。

另一种书写的方式如下。假设而非。我们有和。所以：

（8.4）

与线性回归不同，我们无法求得MLE的封闭解。取而代之的是，我们需要使用优化算法去计算它。为了实现这一点，我们需要导出梯度和海森矩阵。

在逻辑回归中，结果表明梯度和海森矩阵为：



其中。结果表明H为一个正定矩阵。所以NLL是一个凸函数，且具有一个唯一的全局最小值。下面我们将讨论找到这个最小值的方式。

8.3.2 最速下降法

对于无约束优化问题，最简单的算法可能就是梯度下降法（gradient descent），又被称为最速下降法（steepest descent）。该算法可表达为：

（8.8）

其中被称为步长（step size）或者学习率（learning rate）。在梯度下降中最主要的问题在于：我们如何设置步长？这是一个十分棘手的问题。如果我们使用一个固定的但十分小的学习率，那收敛速度会十分缓慢，但如果我们设置的值很大，那么最终将无法收敛。图8.2说明了这个问题，在这个图中，我们绘制了如下的函数：



通过观察，全局最小值在点（1,1）处取得。我们任意地决定从点（0,0）处出发。在图8.2（a）中，我们使用一个固定的步长；我们发现它沿着峡谷移动的很慢。在图8.2（b）中，我们使用一个固定的步长。我们发现算法开始在山谷两边振荡且不再收敛。

下面介绍一种选择学习率的更稳定的方法，这样无论我们从哪个点出发，算法都可以保证收敛于局部最优解。（这个性质被称为全局收敛（global convergence），注意不要与收敛于全局最优解混淆！）根据泰勒理论，我们有：

（8.10）

其中d为衰减方向。所以如果足够小，那么。因为梯度是负数。但是如果我们不希望选择太小的步长，否则我们会移动地很慢，可能无法到达最小值。所以我们通过最小化下式实现的选择：

（8.11）

这被称为线性最小化（line minimization）或者线性搜索（line search）。在1维情况下，有不同的方式对上式问题进行求解。

图8.3（a）说明对于我们的简单的案例，线性搜索的确有效。然而，我们发现基于线性搜索的最速下降法的优化路径具有典型的之字型特征。为了说明这一点，注意到精确的线性搜索满足。最优解满足的必要条件是。根据链式法则，我们有，其中为更新终点的梯度。所以要么我们找到一个驻点，使，或者，意味着精确搜索停止时，局部梯度与搜索方向垂直。所以连续的方向是正交的（见图8.3（b））。这就解释了之字型的原因。

一种减小之字形影响的启发式方式是添加动量（momentum）项，如下：

（8.12）

其中控制着动量项的重要程度。在最优化领域，这被称为重球方法（heavy ball method）。

另一个减少“之字形”影响的方式是使用共轭梯度（conjugate gradients）法。在解决线性系统问题是，其目标函数是二次型，此时这种方法被选择。然而，非线性共轭梯度不那么有名。

8.3.3 牛顿法

将空间的曲率（比如海森矩阵）考虑进来，可以得到更快的优化方法。这些被称为二阶（second order）优化方法。主要的例子是牛顿法（Newton’s algorithm）。这是一个迭代的方式，其参数更新的形式为：

（8.13）

算法的伪代码见算法8.1。

这个算法的推导方式如下。将函数在周边进行二阶泰勒展开：

（8.14）

对上式进行重写：



其中：

（8.16）

的最小值所在点为：



所以为了最小化函数*f*在附近的二阶近似，需要增加在上的牛顿步长为。图8.4（a）给出了说明。

在它的最简单的形式中，牛顿法要求是正定矩阵，也就是说要求函数是严格凸函数。如果目标函数是非凸的，那么可能并非正定矩阵，所以可能并不是下降的方向（见图8.4（b））。在这种情况下，一种简单的策略是回归到最速下降法。Levenberg Marquardt算法是一种混合牛顿步长和最速下降步长的自适应算法。在解决非线性最小二乘的问题时，这种方法被广泛使用。一种可选的方案是：相较于直接计算，我们可以使用共轭梯度法求解线性方程组中的。如果不是正定矩阵，只要检测到负曲率，我们就可以简单地截断共轭梯度迭代。这种方式被称为截断牛顿（truncated Newton）。

8.3.4 重复再加权最小二乘

下面使用牛顿法求解逻辑回归模型中参数的MLE。第i+1次的牛顿迭代如下（使用，因为海森矩阵是精确的）：



其中定义工作响应（working response）为：

（8.23）

式8.22是一个加权最小二乘问题（weighted least squares problem）的例子，它是下式最小值对应的解：

（8.24）

因为是一个对角矩阵，我们可以以组分的形式（对每个样本i=1:N）重写目标值：

（8.25）

这个算法被称为重复再加权最小二乘（iteratively reweighted least squares,IRLS），因为在每一次迭代中，我们需要解决一个加权最小二乘问题，其中权重矩阵在每一次迭代中都会改变。算法8.2给出了伪代码。