几种线性分类器的异同

Yu-Zhe Shi

2020 年 5 月 9 日

摘要

本文介绍了统计学习中线性分类器的共性，并从概率模型类型、损失函数和目标函数三个角度比较了它们特性的不同之处。

1 线性分类器的共性

感知机 (Perceptron)，朴素贝叶斯分类器 (Naive Bayes Classifier)，线性回归 (Linear Regression)，对数几率回归 (Logistic Regression)，支持向量机 (Support Vector Machine) 等都是常见的线性分类器。首先介绍它们的共性。

给定大小为 m 的带标签数据集 $D = \{(\mathbf{x}_1, y_1,), \ldots, (\mathbf{x}_m, y_m)\}$，$\mathbf{x}_i \in \mathbb{R}^{1\times d}$ 是样本的特征向量，其中 d 为样本特征维度，$y_i \in \mathcal{Y}$ 是样本的标签，对于二分类问题，$\mathcal{Y} = \{+1, -1\}$，即分为正类和负类；对于多分类问题，$\mathcal{Y} = \{l_1, \ldots, l_C\}$，其中 C 为数据中包含的标签类别数目。为了讨论方便，后面的举例都以二分类问题说明。分类问题就是以一系列测试样本作为输入，以预测它们对应的标签作为输出。通常，这里的输入和输出对应一组评分函数 $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$ 的输入和输出，当评分函数高于某个预设阈值时，样本被判定为正类；反之，样本被判定为负类。

对于线性分类器，我们的目标是学得一组权重 $\mathbf{w}$，它能对样本特征向量的不同维度赋权，使得评分函数能够“知道”哪些维度的特征对于这个分类任务是较重要的，哪些是较不重要的。由此，评分函数的输出能够尽可能地接近样本对应的真实标签，即使分类器的分类效果尽可能好，即最小化泛化误差。事实上，这一系列权重就是样本不同分量的线性组合，它们构成了一个超平面，正是这个超平面试图将样本空间中不同类别的样本分开。如果我们将所有的样本和对应的超平面映射到二维空间，便可以发现不同类别的样本聚集于平面内不同的区域，区域之间有直线将它们分开。

与线性分类器不同的是，非线性分类器可以学得更加复杂的边界，有可能是某个曲面。尽管工业界中线性可分的数据很少，线性分类器依然有广泛的用途。而且由线性分类器可以发展或组合出（如深度神经网络）非线性分类器。

2 线性分类器的特性

虽然同属线性分类器，感知机、朴素贝叶斯分类器、线性回归、对数回归和支持向量机之间也有着显著差异。

2.1 从模型类别角度

首要的区别是模型类别不同。朴素贝叶斯分类器是生成式模型 (Generative Model)，对联合分布 $P(\mathbf{x}, y)$ 建模。欲求得后验概率 $P(y|\mathbf{x})$，必须基于训练数据得到类别的分布 $P(y)$ 和先验概率 $P(\mathbf{x}|y)$。其对类别的判定准则为

$$h_{gen}(\mathbf{x}) = \arg_{y\in\mathcal{Y}} \max\ P(y) \prod_{i=1}^{d} P(x_i|c) \tag{1}$$

即利用贝叶斯公式和来判定。

而基于线性回归的分类器都是判别式模型 (Discriminative Model)，直接对后验概率 $P(y|\mathbf{x})$ 进行建模，最大化后验概率

$$h_{dis}(\mathbf{x}) = \arg_{y\in\mathcal{Y}} \max \prod_{i=1}^{d} P(y_i|x_i; w, b) \tag{2}$$

即最大似然估计 (MLE) 或最大化后验概率 (MAP) 来判定。前者适用于无正则化项的线性回归模型，后者适用于带 L2 正则化项（或）的线性回归模型（岭回归）。尽管支持向量机的目标函数与岭回归的目标函数本质相近，但由于支持向量机的超平面仅由最接近决策边界的样本确定，它并不能估计某样本是属于某一类的概率，即 $P(y|\mathbf{x})$，这是 SVM 与线性回归模型的区别。

	Discriminative	Generative	$P(y\|\mathbf{x})$	Multinomial	Gaussian
Perceptron	√		√	√	
Linear Regression	√		√	√	
Naive Bayes Classifier		√	√		√
Logistic Regression	√		√		√
Support Vector Machine	√			√	√

表 1: 从概率模型角度的比较

2.2 从损失函数角度

除了本质区别外，本文中探讨的这五种分类器在使用的损失函数方面也有区别。为了突出“线性分类器”的主题，这里所有函数全展开为线性组合形式：

- 感知机的损失函数是单位阶跃函数 (0-1 Loss)

$$y(\mathbf{x}) = sign(\mathbf{w}^T\mathbf{x} + b)$$
$$\mathcal{L}(\mathbf{x}, y) = \begin{cases} 1, & y \neq \mathbf{w}^T\mathbf{x} + b \\ 0, & y = \mathbf{w}^T\mathbf{x} + b) \end{cases} \tag{3}$$

有些感知机使用松弛的 0-1 损失函数，即

$$\mathcal{L}(\mathbf{x}, y) = \begin{cases} 1, & |y - (\mathbf{w}^T\mathbf{x} + b)| \geq M \\ 0, & |y - (\mathbf{w}^T\mathbf{x} + b)| < M \end{cases} \tag{4}$$

尽管 0-1 损失能够清晰展示被归类错误的样本数目，但它是非凸函数，不利于优化求解。

- 线性回归分类器的损失函数是均方误差 (Square Error)

$$\mathcal{L}(\mathbf{x}, y) = \big(y - (\mathbf{w}^T\mathbf{x} + b)\big)^2 \tag{5}$$

这种损失函数具有良好的几何意义：它衡量了输出标记与真实标记的欧氏距离。

- 朴素贝叶斯分类器本质上的损失函数和对率回归一致，都是对率回归损失 (Logistic Loss)

$$y(\mathbf{x}) = \frac{1}{1 + e^{-(\mathbf{w}^T\mathbf{x}+b)}} \tag{6}$$

以损失函数，即真实标签 y 也作为函数输入的形式重写得

$$\mathcal{L}(\mathbf{x}, y) = \ln \frac{P(y=1|\mathbf{x})}{P(y=0|\mathbf{x})} = \ln \frac{e^{y(\mathbf{w}^T\mathbf{x}+b)}}{1 + e^{\mathbf{w}^T\mathbf{x}+b}} \tag{7}$$

对数似然函数可以看作“平滑”版的阶跃函数。实际上，它利用了高斯分布表征 $P(x_i|y) = \mathcal{N}(\mu_{i,C}, \sigma^2_{i,C})$，因此朴素贝叶斯分类器和对率回归函数适用于处理基于高斯分布连续特征的样本。为了简化这个形式，令 $\beta = (\mathbf{w}; b)$，$\hat{\mathbf{x}} = (\mathbf{x}; 1)$，同时计算输出标签与真实标签之间的欧氏距离，我们可以将损失函数重写为

$$\begin{aligned}\mathcal{L}(\mathbf{x}, y) &= y \ln P(y=1|\hat{\mathbf{x}}) + (1-y)\ln(1 - P(y=0|\hat{\mathbf{x}})) \\ &= y \ln \frac{e^{\beta\hat{\mathbf{x}}}}{1 + e^{\beta\hat{\mathbf{x}}}} + (1-y)\ln \frac{1}{1 + e^{\beta\hat{\mathbf{x}}}} \\ &= (y-1)\ln(1 + e^{\beta\hat{\mathbf{x}}})\end{aligned} \tag{8}$$

这种损失函数称为交叉信息熵损失 (Cross-Entropy Loss)。

- 支持向量机的损失函数是合页损失 (Hinge Loss)

$$\mathcal{L}(\mathbf{x}, y) = \max\big(0, 1 - y(\mathbf{w}^T\mathbf{x} + b)\big) \tag{9}$$

这对应了 SVM 只考虑“支持样本”的策略：任何距离决策边界太远的样本不对计算损失作贡献。这种损失是上述四种损失函数中鲁棒性最好，即对噪声最不敏感的。然而，正如前文所述，它不能估计支持样本外的样本属于该类的概率，因此没有很好的统计学习可解释性。

2.3 从目标函数角度

损失函数刻画的是模型的经验风险，而目标函数则关注的是模型的结构风险。目标函数包括经验风险和一衡量模型自身复杂性相关的项，也称为正则项 (Regularization Term)。线性分类器算法的目标就是最小化模型的结构风险。本文中探讨的这五种分类器具有不同的目标函数

- 感知机的目标函数可以写成简单

$$(\mathbf{w}^*) = \arg_{\mathbf{w}} \min \|\mathbf{w}^T\mathbf{x}\|_1 \tag{10}$$

且没有任何限制条件。

- 线性回归分类器的目标函数可以写成 L2 项

$$(\mathbf{w}^*, b) = \arg_{\mathbf{w},b} \min \|y - (\mathbf{w}^T\mathbf{x} + b)\|_2^2 \tag{11}$$

这个问题可以用凸优化求解。欲避免优化问题陷入局部最优，应加入正则化项。常用的正则化项包括 L2 正则化，这使得线性回归成为岭回归 (Tikhonov Regression)

$$(\mathbf{w}^*, b) = \arg_{\mathbf{w},b} \min \|y - (\mathbf{w}^T\mathbf{x} + b)\|_2^2 + \|\mathbf{w}\|_2^2 \tag{12}$$

以及 L1 正则化。尽管 L1 正则化项可以保证我们获得更稀疏的解，但优化过程较麻烦，需要构造 L-Libschitz 条件和 LASSO 方法求解。

- 朴素贝叶斯分类器和对率回归的目标函数都是

$$(\beta^*) = \arg_{\beta^*} \min(y-1)\ln(1 + e^{\beta\hat{\mathbf{x}}}) \tag{13}$$

这个问题可以用凸优化求解。

- 支持向量机的目标函数是

$$\begin{aligned}(\mathbf{w}^*, b) = \arg_{\mathbf{w},b} \min\ & \frac{1}{2}\|\mathbf{w}\|_2^2 \\ subject\ to\ & y(\mathbf{w}^T\mathbf{x} + b) \geq 1\end{aligned} \tag{14}$$

这个问题可以用凸优化求解。

	Loss Function	Objective Function	Constrained?	Convex?
Perceptron	0-1 Loss	LP=L1		
Linear Regression	Square Error	LP=L2		√
LR: Ridge Regression	Square Error	LP=L2+L2		√
LR: Sparse Regression	Square Error	LP=L2+L1		
Naive Bayes Classifier	Cross-Entropy Loss	LP=L2		√
Logistic Regression	Cross-Entropy Loss	LP=L2		√
Support Vector Machine	Hinge Loss	LP=L2	√	√

表 2: 损失函数及目标函数对比