

一. BPR推导过程

1. 前言

\succ_u 表示 item-item 的二元关系，针对每个用户都有一个用 \succ 的右下标区分。若 item i 和 item j 中，一个观察到，另一个未观察到，则对该用户 u 来说， i, j 之间有偏好关系，但 (u, i, j) 到 D_{obs} 中与以往行的区别的。

以前：观察到的认为 positive，该 item 值 1，未观察到则认为 negative，item 值 0。

BPR：不对单个 item 值，因为未观察到的可能是 negative，也可能 missing values。

而是使用 item pair，一个观察到和一个未观察到的 item 之间有 prefer 的关系。

2. 公式推导

$$\begin{aligned} P(\theta | \succ_u) &= \frac{P(\succ_u | \theta) P(\theta)}{P(\succ_u)} \propto P(\succ_u | \theta) P(\theta) \\ &= \prod_{u \in U} P(\succ_u | \theta) P(\theta) \\ &= \prod_{(u, i, j) \in D_{\text{obs}}} P(i \succ_u j | \theta) P(\theta) \\ &= \prod_{(u, i, j) \in D_{\text{obs}}} G(\hat{\alpha}_{uij}(\theta)) P(\theta) \end{aligned}$$

其中 $P(\theta) \sim N(0, \Sigma_\theta) = N(0, \lambda \theta I)$

$$\therefore P(\theta) = \frac{1}{\sqrt{2\pi\lambda\theta I}} e^{-\frac{\theta^2}{2\lambda\theta I}}, \quad \ln P(\theta) = \frac{\ln(\sqrt{2\pi\lambda\theta I})}{2\lambda\theta I} - \frac{1}{2}\|\theta\|^2$$

$$\therefore \ln P(\theta | \succ_u) = \sum_{(u, i, j) \in D_{\text{obs}}} \ln G(\hat{\alpha}_{uij}) + \underline{\ln P(\theta)} \text{ 如何得 } -\frac{1}{2}\|\theta\|^2 ?$$

二. AUC学习

二分类的混淆矩阵中，TP, FP, TN, FN

false positive rate : $FPR = \frac{FP}{FP + TN}$, 所有的负样本中被预测为正样本比例，越小越好

true positive rate : $TPR = \frac{TP}{TP + FN}$, 所有的正样本中被预测为正样本比例，越大越好

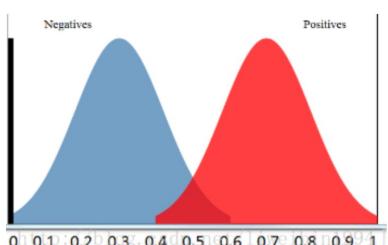
绘制 ROC 曲线：FPR 为横轴，TPR 为纵轴

我们知道，在二分类 (0, 1) 的模型中，一般我们最后的输出是一个概率值，表示结果是1的概率。那么我们最后怎么决定输入的x是属于0或1呢？

我们需要一个阈值，超过这个阈值则归类为1，低于这个阈值就归类为0。所以，不同的阈值会导致分类的结果不同，也就是混淆矩阵不一样了，FPR和TPR也就不一样了。所以当阈值从0开始慢慢移动到1的过程，就会形成很多对(FPR, TPR)的值，将它们画在坐标系上，就是所谓的ROC曲线

AUC的意义

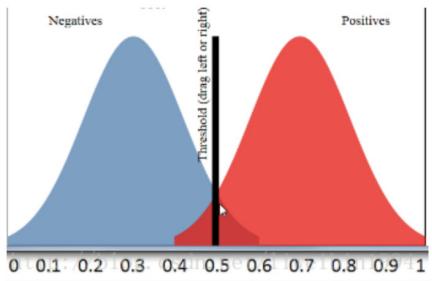
假设我们有一个分类器，输出是样本输入正例的概率，所有的样本都会有一个相应的概率，这样我们可以得到下面这个图：



其中，横轴表示预测为正例的概率，纵轴表示样本数。

所以，蓝色区域表示所有负例样本的概率分布，红色区域表示所有正例样本的概率分布。显然，如果我们希望分类效果最好的话，那么红色区域接近1越好，蓝色区域越接近0越好。

为了验证你的分类器的效果。你需要选择一个阈值，比这个阈值大的预测为正例，比这个阈值小的预测为负例。如下图：

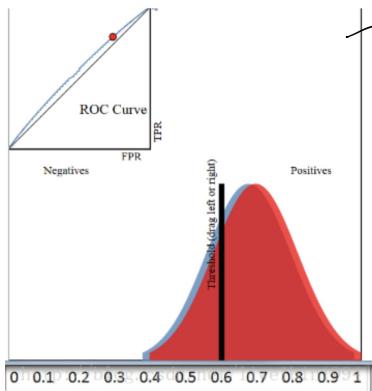


选择一下阈值，计算出一下 (FPR, TPR)

在这个图中，阈值选择了0.5于是左边的样本都被认为是负例，右边的样本都被认为是正例。可以看到，红色区域与蓝色区域是有重叠的，所以当阈值为0.5的时候，我们可以计算出准确率为90%。



上图中，蓝色区域与红色区域的重叠部分不多，所以可以看到ROC曲线距离左上角很近。



→ 与前三张图不同，红蓝区域反演化

前三张图：第一个分类器，讲述画图思路与过程

这一张图：另一种分类器，为了与上一个分类器对比

→ 说明分类器效果不好

但是，当蓝色区域与红色区域基本重叠时，ROC曲线就和接近y=x这条线了。

AUC总结：area under curve

综上两个图，如果我们想要用ROC来评估分类器的分类质量，我们就可以通过计算AUC (ROC曲线下的面积) 来评估了，这就是AUC的目的。

其实，AUC表示的是正例排在负例前面的概率。

三. 学习算法 Learn BPR

- 采用随机梯度下降而非完全梯度下降



针对每个样本都更新 θ

遍历所有样本，才更新一次参数

- 采用有放回的 bootstrap 样本，减少样本顺序的干扰，收敛更快

推导

$$\text{优化目标 } \max \sum_{(u, i, j) \in S} \ln b(\hat{x}_{uij}) - \lambda \|\theta\|^2$$

$$b(x) = \frac{1}{1 + e^{-x}}$$

$$b'(\theta) = \frac{e^{-\theta}}{(1 + e^{-\theta})^2}$$

$$\frac{\partial \ln b(\hat{x}_{uij})}{\partial \theta} = \frac{1}{b(\hat{x}_{uij})} \cdot \frac{\partial b}{\partial \hat{x}_{uij}} \cdot \frac{\partial \hat{x}_{uij}}{\partial \theta}$$

$$= (1 + e^{-\hat{x}_{uij}}) \cdot \frac{e^{-\hat{x}_{uij}}}{(1 + e^{-\hat{x}_{uij}})^2} \cdot \frac{\partial \hat{x}_{uij}}{\partial \theta}$$

$$= \frac{e^{-\hat{x}_{uij}}}{1 + e^{-\hat{x}_{uij}}} \cdot \frac{\partial \hat{x}_{uij}}{\partial \theta}$$

$$\therefore \text{偏导为 } \sum_{(u, i, j) \in S} \frac{e^{-\hat{x}_{uij}}}{1 + e^{-\hat{x}_{uij}}} \cdot \frac{\partial \hat{x}_{uij}}{\partial \theta} - \lambda \|\theta\|^2 \quad (\text{比BPR少一个项?})$$

$$\therefore \theta := \theta - \alpha \cdot \text{偏导}$$

四. BPR应用至MF

详见网课小结

前面算法的过程很完善了，应用到MF中只是确定了 $\frac{\partial \hat{x}_{uij}}{\partial \theta}$ 这项的具体表达式

五. 理解

- BPR 的全称是贝叶斯个性化排序 (Bayesian Personalized Ranking)，它是一种排序算法，做的是TopN任务。
- 适用于隐性反馈数据，当然显性反馈数据把评分变成隐性的也行。
- 以往做TopN任务的做法是利用用户的行为数据，得到用户对所有物品的预测评分，按照分数的高低进行排序推荐，比如FunkSVD。
- 以往那些方法的建模目标是对于要进行推荐的物品尽可能的预测为1，对不应该进行推荐的物品尽可能的预测为0。这样的建模目标并不能够学习到物品之间的先后顺序（虽然我们一直假想的把预测值作为排序的依据）。
- BPR的提出旨在解决物品之间的顺序问题，提出了一个可以应用于各个已有的推荐算法的算法框架，这个算法框架的建模目标是优化物品之间的顺序，解决推荐的物品之间的排序问题。
- 当然这里还有一个建模目标是解决隐性反馈数据不能显示出用户偏好的问题。
- 其实BPR-MF和FunkSVD都是优化P和Q矩阵，得到训练好的这两个矩阵之后，可以计算用户对所有物品的预测评分，然后选择分数排序最高的商品输出。
- 两者不同之处在于优化的目标不一样，BPR-MF优化的目标是加大用户喜欢i物品和喜欢j物品的一个概率差值，而FunkSVD的优化目标是为了降低预测分数与本来的分数之间的差值。

