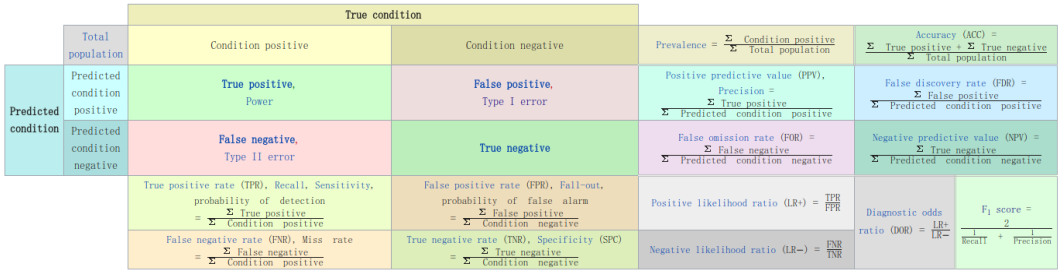

Fig 8.24. Confusion Matrix 示意图

ROC & AUC

ROC 曲线，恒周围假阳性概率 (FP)，真阳性 (TP) 为纵轴组成的坐标图。
真阳性又称为灵敏度。
ROC 曲线越靠近左上角，实验的准确性就越高。
AUC 为 ROC 曲线下的面积，哪一种实验的 AUC 最大，则哪一种实验的诊断判断性最佳。
AUC 越接近于 1, 说明效果越好。

8.28.2 机器学习面试总结

参考文献：机器学习面试总结 -博客园

8.29 花书前两部分总结

8.29.1 第三章

概率论的知识

Sigmoid 函数：

$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$

其导数：

$$\frac{d}{dx}\delta(x) = \sigma(x)\left(1 - \sigma(x)\right)$$

并且有：

$$1 - \sigma(x) = \sigma(-x)$$

信息论

一个事件 x 的自信息为：

$$I(x) = -\log(Pr(x))$$

自信息只处理单个输出。我们可以用香农熵来对整个概率分布中的不确定性总量进行量化：

$$H(x) = \mathbb{E}_{x\sim P}\left[\log Pr(x)\right]$$

对于同一个随机变量 x，有两个单独的概率分布 $P(x)$ 和 $Q(x)$，**可以使用 KL 散度来衡量两个分布的差异**！

$$\begin{aligned} D_{KL}(P \parallel Q) &= \mathbb{E}_{x\sim P}\left[\log \frac{P(x)}{Q(x)}\right] \\ &= \mathbb{E}_{x\sim P}\left[\log P(x) - \log Q(x)\right] \end{aligned}$$

从上面的公式可以看出，假设输入的 x 服从 P 的分布，计算此时的 P 与 Q 之间的差异。

KL 散度是非负的，并且衡量的是两个分部之间的差异，它经常被用作分布之间的某种距离。然而他并不是真正的距离，因为它不是对称的。通过选择 $D_{KL}(P \parallel Q)$ 和 $D_{KL}(Q \parallel P)$ 对结果影响较大。

一种与 KL 散度联系密切的量是交叉熵。即 $H(P,Q) = D_{KL}(P \parallel Q) + H(P)$, 可以得到下面的公式：

$$H(P,Q) = -\mathbb{E}_{x\sim P}\left[\log Q(x)\right]$$

针对分布 Q 最小化交叉熵等价于最小化 KL 散度，因为分布 Q 并不参与被省略的那一项。一种特殊的情况：$\lim_{x\leftarrow 0} x\log x = 0$。

个人补充：

后面我们可以看到，**这个交叉熵的优化等价于最大似然的优化**！如果但看上面交叉熵的计算公式的话，可以这么认为：分布 P 是数据的真实分布，Q 是模型的分布，我们优化的目的就是让这两种分布之间的差距最小；然而一般情况下我们并不知道真是的数据分布，一种直观的方法就是用训练数据的经验分布代替真实的的数据分布。

结构化概率模型

这部分的有向、无向图模型更详细的内容可以参考 << 计算机视觉 -模型、学习和推理 >> 或者 << 统计学习方法 >>。

8.29.2 第四章

数值计算

上溢和下溢

必须对上溢和下溢进行数值稳定的一个例子是 **Softmax 函数**，该函数后面可以看到，经常用于预测和 Multinoulli 分部相关联的概率，定义如下：

$$softmax(x)_i = \frac{\exp(x_i)}{\sum_{j=1}^{n}\exp(x_j)}$$

若所有的 $x_j = c$，那么 softmax 结果为 $\frac{1}{n}$。从数值上来说，当 c 量级很大时，这可能不会发生什么。如果 c 很小的负数，那么 $\exp(c)$ 就会下溢，这就意味着 softmax

的分母变为 0，所以结果是未定义的；另一方面，如果 c 很大的正数，那么 exp(x) 本身就会上溢，同样导致未定义的结果。

解决办法：

构建新的变量：$\mathbf{z} = \mathbf{x} - \max_i x_i$, 然后用 z 代替 x，即：

$$softmax(\mathbf{z})$$

从公式中可以看出，softmax 解析上的函数值不会因为从输入向量减去或加上标量而改变。减去 $\max_i x_i$ 导致 exp 的最大参数为 0，这排除了 exp 函数的上溢的可能性；同样的，分母中至少有一个值为 1 的项，这就排除了因分母下溢而导致的零除的可能性。

病态条件

首先引入条件数的概念，条件数指的是函数相对于输入的微小变化而变化的快慢程度。也就是说，输入微小扰动，输出也可能迅速变化。

考虑函数 $f(x) = \mathbf{A}^{-1}\mathbf{x}$, 当 $A \in R^{n*n}$ 具有特征分解时，其条件数为：

$$\max_{i,j} \left| \frac{\lambda_i}{\lambda_j} \right|$$

这里与通常的条件数定义有所不同，即取最大和最小特征值的模之比。当该数很大时，矩阵求逆对输入的误差特别敏感。后面可以看到，L2 正则项对病态有一些帮助。

基于梯度的优化方法

函数在 **u** 方向的方向导数，是函数 f 在 u 方向的斜率。换句话说，方向导数是函数 $f(x + \alpha u)$ 关于 α 的导数。使用**链式法则**，我们可以看到当 $\alpha = 0$ 时，有：

$$\frac{\delta}{\delta\alpha} f(\mathbf{x} + \alpha\mathbf{u}) = \mathbf{u}^T \nabla_x f(\mathbf{x})$$

为了最小化 f，我们希望找到使 f 下降的最快的方向。通过上面公式可以看出，当 **u** 和 $\nabla_x f(\mathbf{x})$ 的方向相反时，它俩的夹角为 0，cos 值为 1，得到最大。所以这又被称为最速下降法或梯度下降。

梯度下降决定了下降最快的方向，我们还需要设定在该方向的步长，也就是学习率了，一种选择学习率的方法是线搜索。

梯度之上

Jacobian 和 Hessian 矩阵。

Hessian 是实对称矩阵，可以分解成 $H = Q\Lambda Q^T$。在特定方向 **d** 上的二阶导数可以写成 $d^T H d$。当 d 是 H 的一个特征向量时，这个方向的二阶导数就是对应的特征值。对于其他方向的 d，方向的二阶导数是所有特征值的加权平均。最大特征值确定最大二阶导数，最小特征值确定最小二阶导数。

将函数 $f(x)$ 进行二级泰勒级数近似，则可以计算求得在进行 ϵg 步长更新后，其函数值为：

$$f(x^{(0)} - \epsilon g) = f(x^{(0)}) - \epsilon g^T g + \frac{1}{2} g^T H g$$

其中有三项，函数的原始值、函数斜率导致的预期改善、函数曲率导致的矫正。为了使 $f(x^{(0)}-\epsilon g)$, 那么上式的右边应该使 $-\epsilon g^T g+\frac{1}{2}g^T H g$ 最小，为了确定最优的学习率，那么把后面这一项看做是 ϵ 的二阶方程，求解其最小值，应该使得一阶导数为 0。对后一项进行对 ϵ 求导，则得到最优的学习率为：

$$\epsilon^* = \frac{g^T g}{g^T H g}$$

在最坏的情况下，g 与 H 的最大特征值 λ_{max} 对应的特征向量对齐，此时最优步长为 $\frac{1}{\lambda_{max}}$。Hessian 的特征值决定了学习率的量级。

Hessian 的另一个作用就是：二阶导数测试。在临界点 (一阶导数为 0) 时，若 Hessian 是正定的，则该临界点是**全局最小点**，因为方向二阶导数在任意方向都是正的 $d^T H d > 0$，所以该临界点的导数导数都是正的，也就是该点是全局最小点了。当 Hessian 矩阵是负定的时候，该点是**局部极大点**。如果 Hessian 的特征值有大于 0 的值，也有小于 0 的值，则为鞍点。

当 Hessian 的条件数很差时，梯度下降法也会表现的很差。这是因为一个方向上的导数增加的很快，而在另一个方向上增加的很慢。梯度下降不知道导数的这种变化，所以它不知道应该优先搜索导数长期为负的方向，而不是曲率最大的方向。病态条件也导致很难选择合适的补偿。

我们可以使用 Hessian 矩阵的信息来指导搜索。比如牛顿法。

约束优化

包含著名的 KKT 条件 + 广义拉格朗日了。

8.29.3 第五章

机器学习基础

设计矩阵

设计矩阵的每一行包含一个不同的样本，每一列对应不同的特征。

容量、过拟合和欠拟合

模型的容量：通俗来讲，模型的容量是指其拟合各种函数的能力。容量高的模型可能会过拟合，因为记住了不适用于测试集的训练集性质，所以从这个角度来看，增加正则项来防止过拟合本质上是限制模型可以拟合的函数的空间的大小。

VC 维度：定义为该分类器能够分类的训练样本的最大数目。

正则化：是指修改学习算法、使其降低泛化误差而非训练误差。

正则化是机器学习领域的中心问题之一，只有优化能够与其重要性相提并论。这一点从花书的第七章、第八章可以看出来，第七章全面的讲了正则化、第八章全面的讲了优化算法。

估计、偏差和方差

偏差和方差度量着估计量的两个不同误差来源。偏差度量着偏离真实函数或参数的误差期望，而方差度量着数据上任意特定采样可能导致的估计期望的偏差。

均方误差 (MSE):

$$\begin{aligned} MSE &= \mathbb{E}\left[(\hat{\theta}_m - \theta)^2\right] \\ &= Bias(\hat{\theta}_m)^2 + Var(\hat{\theta}_m) \end{aligned}$$

MSE 度量着估计与真实参数 θ 之间的平方误差的总体期望偏差。如上式所示，MSE 包含了偏差和方差。理想的估计具有较小的 MSE 或是在检查中会稍微约束他们的偏差和方差。

偏差和方差的关系与机器学习容量、欠拟合和过拟合的概念紧密相连。用 MSE 度量泛化误差 (偏差和方差对于泛化误差都是有意义的) 时，增加容量会增加方差，降低偏差。

补充一点：

偏差和方差的区别

偏差：描述的是预测值的期望与真实值之间的差距。

方差：描述的是预测值的变化范围，离散程度，也就是距离其期望值的距离。

当偏差较大时，一般发生欠拟合；当方差较大时，泛化能力较差，一般发生过拟合。随着模型容量的增加，偏差下降，方差呈 U 型。

训练程度与偏差、方差的关系：

1. 训练不足时，学习器的拟合能力不够强，训练数据的扰动不足以使学习器发生显著变化，偏差将主导泛化错误率。

2. 当训练程度加深，学习器的能力逐渐增强，训练数据发生的扰动逐渐能够被学习器学到，方差将主导泛化错误率。

3. 训练程度充足后，学习器的拟合能力已经非常强，训练数据发生的轻微扰动都会导致学习器发生显著变化。将发生过拟合。

最大似然估计

这一部分主要推导下前面提到的，最大似然估计与交叉熵之间的关系。

最大似然估计的目标是：

$$\begin{aligned} \theta_{ML} &= \arg\max_{\theta} p_{model}(\mathcal{X}; \theta) \\ &= \arg\max_{\theta} \prod_{i=1}^{m} p_{model}(x^{(i)}; \theta) \end{aligned}$$

其中，m 是样本数量。

推导如下，将上面的最大似然转换成对数形式, 则乘积编程加法:

$$\begin{aligned} \theta_{ML} &= \arg\max_{\theta} \sum_{i=1}^{m} \log p_{model}(x^{(i)}; \theta) \\ &= \arg\max_{\theta} \mathbb{E}_{x \sim \hat{p}_{data}} \log p_{model}(x; \theta) \end{aligned}$$

上面的推导成立的原因：因为当重新缩放代价函数时，argmax 的结果不会改变，我们可以除以 m 得到和训练数据经验分布 $\hat{p}_{data}$ 相关的期望作为准则，为啥会是 $\hat{p}_{data}$ 而不是 p_{model} 呢，因为输入的数据 x 是来自训练数据而不是模型。

一种，解释最大似然估计的观点是将它看做最小化训练集上的经验分布和模型分布之间的差异，两者之间的差异程度可以通过 KL 散度度量：

$$D_{KL}\left(\hat{p}_{data} \parallel p_{model}\right) = \mathbb{E}_{x\sim\hat{p}_{data}}\left[\log \hat{p}_{data}(x) - \log p_{model}(x)\right]$$

在上式中，由于左边一项仅设计数据的生成过程，和模型无关。这意味着最小化 KL 散度时，可以省略第一项，得到：

$$-\mathbb{E}_{x\sim\hat{p}_{data}}\left[\log p_{model}(x)\right]$$

对比上式与前面的最大似然的式子，可以发现，最小化 KL 散度就是最小化分布之间的交叉熵，也就等价于最小化负对数似然，也就是最大似然了。

因为最大似然具有：一致性、统计效率，使其成为机器学习中的首选估计方法。

贝叶斯估计

这里主要说明最大后验 (MAP) 估计与后面正则项的关系！

MAP 的公式为：

$$\begin{aligned}\theta_{MAP} &= \arg\max_{\theta} p(\theta|\mathbf{x}) \\ &= \arg\max_{\theta} \left[\log p(\mathbf{x}|\theta) + \log p(\theta)\right]\end{aligned}$$

公式默认省略了数据的生成分布 $p(\mathbf{x})$。从上面可以看出来，相比于 ML，MAP 多了一项：$p(\theta)$，这一项就是参数 θ 的先验概率分布了，该先验信息有助于减少最大后验点估计的方差，但缺点是增加了偏差。更重要的，看上面的公式：是不是跟带有正则项的目标函数比较相似。事实上，许多正则化估计方法，例如权重衰减正则化的最大似然学习，可以解释为贝叶斯推断的 MAP 近似。这个是英语正则化时加到目标函数的附加项对应着 $\log p(\theta)$。但并非所有的正则化惩罚对应着 MAP 贝叶斯推断。

MAP 贝叶斯推断提供了一个直观的方法来设计复杂但可解释的正则化。

随机梯度下降

随机梯度下降的核心是，梯度是期望 (这一点可以从训练数据集上的梯度计算可以看出来，是一个求平均的过程)。既然是期望，就可以用小规模的样本近似估计。一个非常重要的优势：使用随机梯度下降，当总的样本数 m 不断增加时，我们都可以只使用小批量 m' 来计算梯度估计，而且该模型最终会在随机梯度下降抽样完训练集上的所有样本之前收敛到可能的最优测试误差。继续增加 m 不会延长达到模型可能的最优测试误差的时间，从这个角度来看，SGD 训练模型的渐进代价是关于 m 的函数的 $O(1)$ 级别的！！！

促使深度学习发展的挑战

- 维数灾难
- 局部不变性和平滑正则项
- 流行学习

细节就不说了。

8.29.4 第六章

8.29.5 第七章

8.29.6 第八章

8.30 待续