对问题的看法

16337341 朱志儒

1. 怎么理解机器学习模型的泛化能力？如何提升分类模型的泛化能力？

机器学习模型的泛化能力就是指模型在训练集上学习后适用于新样本的能力。机器学习的目标是使学得的模型能很好的适用于新样本，而不是仅仅适用于训练样本。一个具有较强泛化能力的模型可以很好地适用于整个样本空间。

提升分类模型的泛化能力有许多办法:

(1) 增加训练集的样本数目，训练样本越多，模型学到的关于样本空间中全体样本服从的分布的信息越多，最后得到的模型的泛化能力也就会越强。为了增加训练样本的数目，可通过生成模型或是其他的技巧生成新的训练样本。

(2) 对训练数据进行缩放，采用标准化或区间缩放法将不同规格的数据转换到同一规格，将其缩放到模型激活函数的阈值范围。例如，若使用 sigmoid 激活函数，可将数据缩放到 0~1 之间。若使用 tanh 激活函数，可将数据缩放到-1~1 之间。

(3) 对训练数据进行变换。观察猜测每种属性的分布，若是指数分布，则可进行对数变换；若是高斯分布，则可采用 Box-Cox 变换实现正态化处理。

(4) 对训练数据预处理时进行特征选择。选择特征时主要考虑两个方面:

特征是否发散：如果一个特征不发散，例如方差接近于 0，也就是说样本在这个特征上基本上没有差异，这个特征对于样本的区分并没有什么用。

特征与目标的相关性：优先选择与目标相关性高的特征。

特征选择的方法分以下 3 种:

过滤法：根据发散性或相关性对所有特征进行评分，设定阈值选择特征；

包装法：根据目标函数每次选择若干特征，或是排除若干特征；

嵌入法：先使用某些机器学习的算法和模型进行训练，得到各个特征的权值系数，根据系数从大到小选择特征。

(5) 构建模型时尝试不同的算法，比如线性模型、树模型、SVM、KNN 和神经网络模型等，采纳效果较好的方法，然后精细调参。

(6) 使用 K 折交叉验证评判模型的效果，在模型的损失函数中加入正则项，使用 early stopping 可以在一定程度上减小过拟合。

(7) 由于模型总是处于过拟合或欠拟合的两个状态之间，所以需要分别计算模型在训练集和验证集上的准确率。若训练集准确率较高，则存在过拟合现象，可以尝试增加正则项；若训练集和验证集的准确率都很低，则存在欠拟合现象，需继续提升模型能力，增加模型训练轮数。

(8) 尝试不同的方法初始化模型的权重，一般使用小的随机数初始化权重。

使用无监督预训练，训练网络的第一个隐藏层，再训练第二个，最后用这些训练好的网络参数值作为整个网络参数的初始值。

采用迁移学习，我们可以将已经学到的模型参数通过某种方式来分享给新模型从而加快并优化模型的学习效率，而不用像大多数网络那样从零学习。

(9) 选择合适的学习率。学习率直接影响我们的模型能够以多快的速度收敛到局部最小值，一般来说，学习率越大，神经网络学习速度越快。如果学习率太小，网络很可能会陷入局部最优；但是如果太大，损失就会停止下降，在某一位置反复震荡。

所以，我们可以先设置一个较低的学习率，然后随着训练迭代逐渐增大这个值，记录每次训练迭代，画出相应的学习率和 loss 变化，我们可以发现，随着学习率不断提高，loss 会在一段时间不断下降，并在触及最低点后开始回升。理想的学习率应该是使 loss 曲线到达最低点的那个值。使用该学习率训练模型发现训练时模型的 loss 不再发生变化时，模型可能到达局部极小值，则我们可以采用由 Loshchilov 和 Hutter 提出的加入了热重启的随机梯度下降

法，这种方法把余弦函数作为循环函数 f，并在每一轮迭代开始时重设一个最大学习率。这样可以使模型越过这些局部极小点，有利于提高模型的泛化能力。

（10） 尝试使用不同的激活函数，例如 sigmoid 函数、softmax 函数、tanh 函数和 ReLu 函数等。实践过程中更多还是需要结合实际情况，考虑不同激活函数的优缺点综合使用。

（11） 调整模型的网络拓扑结构，我们可以拓宽每层的节点数目，也可以增加网络的层数。由于这些都是模型的超参数，无法从训练集样本中学习得到最优值，但我们可以使用经验法，也可以反复试验探寻其最优值。

（12） 尝试不同的 batch 和 epoch 组合。batch 的大小决定模型一次训练的样本数目，将影响模型的优化程度和速度，epoch 表示使用训练集的全部数据对模型进行一次完整的训练。机器学习模型一般采用较小的 batch 和较大的 epoch 进行训练。如果 batch 大小为整个训练样本数目，则为批量学习，消除样本顺序的影响，对梯度向量精确估计。如果 batch 大小设为 1，则为在线学习，能够追踪训练数据小的改变，对大规模和困难模式分类问题能提供有效解。

2. 请阐述在聚类任务当中，聚类算法重要还是样本点间的距离（相似度）定义更重要?

我认为在聚类任务中，样本点间的距离（相似度）定义更重要。因为评估聚类结果好坏的内部指标是通过计算簇内的样本距离以及簇间的样本距离来对聚类结果进行评估，并且聚类算法都需要用到样本点间的距离。如果距离的定义不够严谨和细致的话，使用各种聚类算法最终的效果也会不尽人意。

定义样本点间的距离需要满足一些基本性质：

非负性：两点之间距离不可为负；

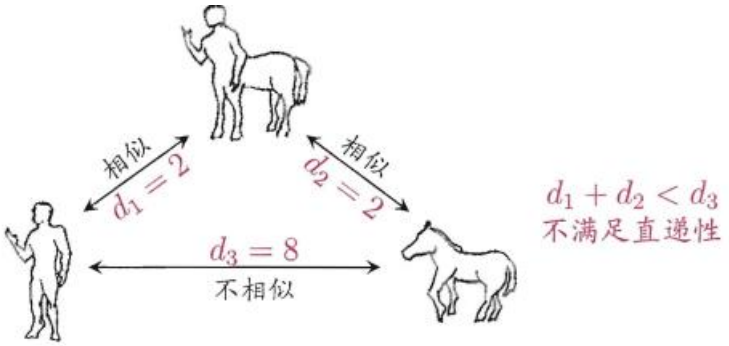
同一性：两个点只有在样本空间上重合才可能距离为零；

对称性：a 到 b 的距离等于 b 到 a 的距离；

直递性：a 到 c 的距离加上 c 到 b 的距离大于等于 a 直接到 b 的距离。

通常我们是基于某种形式的距离来定义相似度度量的，距离越大，则相似度越小。但是用于相似度度

量的距离未必一定满足距离度量的直递性。如下图所示：



我们想让人和人马相似、人马和马相似，但是我们不能认为人和马是相似的，在此情况下，要满足直递性就得要人和马的相似度不大于 4，而直观上来看人和马十分的不相近（距离应该远大于 4），所以在这里直递性就不适合了。这种距离称为“非度量距离”，在不少现实任务中，我们需要基于数据样本来确定合适的距离计算式。

在上述例子中，如果只依据距离度量的性质来定义就会出现问题，采用各种聚类算法最终的效果都不会令人满意。所以我认为在聚类任务当中，样本点间的距离（相似度）定义比聚类算法更重要。

3. 在文本分类、问答和翻译等 NLP 任务中，分析、比较你学过的和已知的文本建模算法。

一篇文档可以看成是一组有序的词的序列，从统计学角度来看，文档的生成可以看成是投掷色子生成的结果，每次投掷色子都会生成一个词汇，投掷 n 次可视为生成一篇 n 词的文档。在统计文本建模中，会涉及到两个最核心的问题：

（1）模型中都有哪些参数，色子的每一个面的概率都对应于模型中的参数；

（2）投掷色子的规则是什么，按照什么规则投掷色子从而产生词序列。

在 Unigram Model 中采用词袋模型，假设了文档之间相互独立，文档中的词汇之间相互独立。假设词典中一共有 v 个词，最简单的 Unigram Model 就是按照以下规则生成文本：

（1） 只有一个骰子，这个骰子有 V 面，每个面对应一个词，各个面的概率不一；

（2） 每抛掷一次骰子，抛出的面就对应的产生一个词；如果一篇文档中 N 个词，就独立的抛掷 n

次骰子产生 n 个词。

在频率派看来，对于一个色子，假设每个面的概率为$\vec{p} = (p_1, p_2, \cdots, p_V)$，每生成词汇可视为多项式分布$\omega \sim \mathrm{Mult}(\omega \mid \vec{p})$。对于一篇文档$\vec{\omega}$，它的生成概率为$p(\vec{\omega}) = p(\omega_1, \omega_2, \cdots, \omega_n) = p(\omega_1)p(\omega_2)\cdots p(\omega_n)$。对于一个语料库$W = (\overrightarrow{\omega_1}, \overrightarrow{\omega_2}, \cdots, \overrightarrow{\omega_m})$，由于文档之间是相互独立的，所以整个语料库的概率为$p(W) = p(\overrightarrow{\omega_1})p(\overrightarrow{\omega_2})\cdots p(\overrightarrow{\omega_m}) = \prod_{k=1}^{V} p_k^{n_k}$，

在贝叶斯派看来，一切参数都是随机变量，以上模型中的骰子$\vec{p}$不是唯一固定的，它也是随机变量。所以，生成文本规则如下：

（1） 现有一个装有无穷多个骰子的坛子，里面装有各式各样的骰子，每个骰子有 V 个面；

（2） 现从坛子中抽取一个骰子出来，然后使用这个骰子不断抛掷，直到产生语料库中的所有词汇。

从概率分布角度看，坛子里面的骰子服从参数$\vec{p}$的先验分布。坛中每个骰子都可能被使用，其概率由先验分布$P(\vec{p})$来决定。对每个具体的骰子，由该骰子产生语料库的概率为$p(W \mid \vec{p})$，则产生语料库的概率就是对每一个骰子上产生语料库进行积分求和$p(W) = \int p(W \mid \vec{p})p(\vec{p})\, d\vec{p}$。

在 Unigram Model 模型中，没有考虑主题词这个概念。人在写文章时都是关于某一个主题的，而不是胡乱写，所以 PLSA 认为生成一篇文档的生成过程如下：

（1） 现有两种类型的骰子，一种是 doc-topic 骰子，每个 doc-topic 骰子有 K 个面，每个面一个 topic 的编号；一种是 topic-word 骰子，每个 topic-word 骰子有 V 个面，每个面对应一个词；

（2） 现有 K 个 topic-word 骰子，每个骰子有一个编号，编号从 1 到 K；

（3） 生成每篇文档之前，先为这篇文章制造一个特定的 doc-topic 骰子，重复如下过程生成文档中的词：投掷这个 doc-topic 骰子，得到一个 topic 编号 z；选择 K 个 topic-word 骰子中编号为 z 的那个，投掷这个骰子，得到一个词。

PLSA 中，也是采用词袋模型，文档和文档之间是独立可交换的，同一个文档内的词也是独立可交换的。

在 PLSA 中生成文档的方式如下：

（1） 按照概率$p(d_i)$选择一篇文档d_i；

（2） 根据选择的文档d_i，从主题分布分布盒中按照概率$p(\zeta_k \mid d_i)$选择一个隐含的主题类别ζ_k；

（3） 根据选择的主题ζk，从词分布中按照概率$p(\omega_j \mid \zeta_k)$选择一个词ω_j。

在 LDA 中，生成文档的过程如下：

（1） 按照先验概率$p(d_i)$选择一篇文档d_i；

（2） 从狄利克雷分布α中取样生成文档d_i的主题分布θ_i，主题分布θ_i由超参数为α的狄利克雷分布生成；

（3） 从主题的多项式分布θ_i中取样生成文档d_i第 j 个词的主题$z_{i,j}$；

（4） 从狄利克雷分布β中取样生成主题$z_{i,j}$对应的词语分布$\phi_{z_{i,j}}$，词语分布$\phi_{z_{i,j}}$由参数β的狄利克雷分布生成；

（5） 从词语的多项式分布$\phi_{z_{i,j}}$中采样最终生成词语$\omega_{i,j}$。

从上面可以看出，LDA 在 PLSA 的基础上为主题分布和词分布分别加了两个狄利克雷先验。

在 PLSA 中，主题分布和词分布都是唯一确定的。但是，在 LDA 中，主题分布和词分布是不确定的，LDA 的作者们采用的是贝叶斯派的思想，认为它们应该服从一个分布，主题分布和词分布都是多项式分布，因为多项式分布和狄利克雷分布是共轭结构，在 LDA 中主题分布和词分布使用狄利克雷分布作为它们的共轭先验分布。

参考文献：

1. 周志华 《机器学习》
2. Leslie N. Smith Cyclical Learning Rates for Training Neural Networks
3. Ilya Loshchilov, Frank Hutter Stochastic Gradient Descent with Warm Restarts
4. Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. Journal of machine Learning research, 3(Jan), 993-1022.
5. Hofmann, T. (1999). Probabilistic latent semantic indexing. In Proceedings of the 22nd annual international

ACM SIGIR conference on Research and development in information retrieval (pp. 50-57). ACM.

6. 通俗理解 LDA 主题模型. https://blog.csdn.net/v_july_v/article/details/41209515