数学基础

概率基础

概率分布

离散型概率分布

- 伯努利分布
- 二项分布

  表示多次进行伯努利实验的结果分布

$$P(K=k)=\left(n\ k\right)p^k(1-p)^{n-k}$$

- 多项分布

$$P(x_1,x_2,\dots x_k;n,p_1,p_2,\dots p_k)=\frac{n!}{x_1!\dots x_k!}p_1^{x_1}\dots p_1^{x_k}$$

  二项分布扩展到多维的情况，指单次实验中随机变量的取值不再是0-1，而是多种离散的可能型

- 贝叶斯理论

在贝叶斯理论中，存在先验概率和后验概率的概念，其中先验概率指的的是事件未发生之前的概率值

后验概率则是指的根据事件发生的情况确定出的事件概率

$$p(\theta|x)=\frac{p(x|\theta)p(\theta)}{\int p(x|\theta')p(\theta')d\theta'}$$

- 共轭分布

在贝叶斯理论中，根据似然函数和先验分布可以得到后验分布，如果得到的后验分布和先验分布具有相同的形式，那么就称作这两种分布是共轭分布，这种先验分布则称为共轭先验

> 共轭先验的好处在于提供了代数上的方便性，可以直接给出后验分布的封闭形式

- 中心极限定理

当n足够大的时候，独立同分布的随机变量之和服从正态分布

- 极大似然估计

给定一个概率分布 D，已知其概率密度函数（连续分布）或概率质量函数（离散分布）为 f_D，以及一个分布参数 θ ，我们可以从这个分布中抽出一个具有 n个值的采样 $X_1, X_2,\ldots, X_n$，利用 f_D计算出其似然函数：

$$L(\theta \mid x_1, \dots, x_n) = f_\theta(x_1, \dots, x_n).$$

若 D是离散分布， $f_{\theta }$即是在参数为 θ 时观测到这一采样的概率。若其是连续分布，$f_{\theta }$ 则为 $X_1, X_2,\ldots, X_n$联合分布的概率密度函数在观测值处的取值。一旦我们获得 $X_1, X_2,\ldots, X_n$，我们就能求得一个关于 θ 的估计。最大似然估计会寻找关于 θ 的最可能的值（即，在所有可能的 θ θ 取值中，寻找一个值使这个采样的“可能性”最大化）。从数学上来说，我们可以在 θ 的所有可能取值中寻找一个值使得似然函数取到最大值。这个使可能性最大的 $\widehat{\theta}$值即称为 θ 的最大似然估计。由定义，最大似然估计是样本的函数。

$$\theta^* = argmax_\theta \sum_X logL(\theta|x)$$

> 对于每一个样本的值所对应的概率值，能够得到该序列中的整体概率，在该概率式子中，只有一个参数是未知的，所以可以使用导数求极值进而得到最终值

- EM算法

在有些模型当中，存在一些隐含的不确定的变量，但是这些变量会参与到概率模型的效果，为了得到这些隐含变量的值，需要对其值进行估计。

为隐含变量的值设置一个概率分布，然后使用极大似然方法找到目标函数：

$$L(\theta, \pi|X) = \sum p(z_i|\pi)p(x_i|\theta, z_i)$$

根据极大似然估计的理论，可以的到θ的值应当取： $$ \begin{aligned} \theta^* & = argmax_\theta \sum_X logL(\theta|x)\ &= argmax_\theta \sum_X log\sum_Z p(z_i|\pi)p(x_i|\theta,z_i)\

\end{aligned}

由$Jensen$不等式，可以得到如下结论：

\sum_X log\sum_Z p(z_i|\pi)p(x_i|\theta,z_i) \ge \sum_X\sum_Zlog(p(z_i|\pi)p(x_i|\theta,z_i)) $$

EM算法包含两个步骤，分别是E-step和M-step

> `E-step`：找到隐含变量的分布或者期望

$$Q(\Theta|\Theta^t) = E(p_{z_i})LL(\Theta|X, Z) \quad or \quad E_{Z|X,\Theta^t}(LL(\Theta|X, Z))$$

> `M-step`：将这个隐含变量的分布或者期望带入到参数的对数似然当中，根据这个对数似然的式子寻找最大化似然的参数

$$\Theta^{t+1} = argmax_\Theta Q(\Theta|\Theta^t)$$

> 本质上EM算法适用于存在隐含变量的模型估计，解决的思路就是假设隐含变量服从的分布与变量的值以及样本值相关，假设样本的分布进而确定隐含变量的分布参数，根据这个参数选择隐含变量的期望或者采样出隐含变量的值，得到关于参数的似然函数。
>
> 然后对似然函数求最大似然估计，得到新一代的参数值。如是迭代

线性代数

凸优化理论

信息论

机器学习

概率学模型

朴素贝叶斯

关于贝叶斯公式，由条件概率公式：

$$P(c|x) = \frac{P(x,c)}{P(x)}$$

可以得到贝叶斯公式：

$$P(c|x) = \frac{P(c)P(x|c)}{P(x)}$$

在其中$P(x|c)$表示的是在类别c中的各个特征的分布，严格来说是所有特征可能值的联合概率分布，求解比较难。

为了排除这个困难，在朴素贝叶斯估计中，假设所有的属性之间独立对结果产生影响。

$$P(c|x) = \frac{P(c)}{P(x)} \prod_{i=1}^{d} P(x_i|c)$$

其中d表示属性的数目，x_i表示第i个属性上的取值 对于所有类别来说，$P(x)$相同，所以可以得到： $$ h_{nb}(x) = \operatorname{argmax}*{c\in y} P(c)\prod*{i=1}^dP(x_i|c)

其中$P(c)$表示经过统计之后，样本集之中类别的频率：

P(c) = \frac{|D_c|}{|D|}

条件概率$P(x_i|c)$通过样本中的取值为该值的样本计数得到：

P(x_i|c) = \frac{|D_{c,x_i}|}{|D_c|} $$ 如果是连续属性，则使用密度函数的方式表示其分布，常用的方式是正态分布

贝叶斯网络

由于朴素贝叶斯假设所有的属性之间都是条件独立的，这种假设往往比较强，所以在贝叶斯网络中，对各个属性之间的独立性关系进行了建模。

条件随机场

隐马尔可夫模型

经典学习模型

决策树

机器学习中，决策树是一个预测模型；他代表的是对象属性与对象值之间的一种映射关系。树中每个节点表示某个对象，而每个分叉路径则代表某个可能的属性值，而每个叶节点则对应从根节点到该叶节点所经历的路径所表示的对象的值。决策树仅有单一输出，若欲有複数输出，可以建立独立的决策树以处理不同输出。 数据挖掘中决策树是一种经常要用到的技术，可以用于分析数据，同样也可以用来作预测。

从数据产生决策树的机器学习技术叫做决策树学习,通俗说就是决策树。

决策树生成的过程如下： 在每一个循环中，选择出一个最优的特征，对这个特征中的属性进行划分，每一个特征的可选值都会对应数据集中的一部分，

> 如果这个数据集中的数据都
> 1．具有相同的标注结果
> 2．数据集为空
> 3．没有可分的特征了
> 则会被生成叶子节点
> 否则将会对接下来的数据进行迭代，直到无法迭代

决策树的生成方式

设计决策树的重要的一个问题就是，如何在每一步选择最合适的抽取特征，方法有如下几种

- ID3:信息增益

  信息熵的定义

  > p_k表示在数据集D中每一个类别的出现的概率
  >
  > $$Ent(D) = -\sum_{k=1}^{|y|} p_k log_2 p_k$$
  >
  > **信息增益的定义**

  > D^v表示按照特征a中的一种可选值v分割样本数据集得到的子数据集
  >
  > $$Gain(D,a) = Ent(D) - \sum_{v=1}^{V} \frac{|D^v|}{|D|} Ent(D^v)$$

  在ID3方法中，选择最合适的特征的方式就是选择具有最高信息增益的特征

- C4.5:信息增益比

  考虑到有的特征的可选值较多，每一个可选值所分出来的子数据集就相对较少，比较容易得到较纯的数据集和较高的信息增益，所以需要把特征分类的数目也考虑进去。

  $$Gain_radio(D,a) = \frac{Gain(D,a)}{IV(a)}$$

其中$IV()$是用来衡量特征类别数的函数：

$$IV(a) = -\sum_{v=1}^{V} \frac{|D^v|}{|D|} log_2 \frac{|D^v|}{|D|}$$

具有较多可选值的特征，其IV值比较高，信息增益率就比较小。

- CART:分类回归树

  在CART中，借助基尼系数进行最佳属性的判定，对于属性的可能值 **基尼系数**

$$Gini(A) = 1 - \sum_{i=1}^{C} p_i^2$$

  直观上讲，基尼系数反映出了集合中随机抽取两个样本属于不同分类的概率，Gini值越小，数据集纯度越高。 属性a的基尼指数定义为：

$$Gini_index(D, a) = \sum_{1}^{V} \frac{|D_v|}{|D|} Gini(D^v)$$

> 由于CART可以解决分类和回归两类任务，而且使用的是二叉树结构，所以在每一个节点都只会设置为大于某个值和小于某个值，是某个值和不是某个值。
>
> 所以每一步确定属性的时候只是在确定某个唯一的属性，和回归任务中的某个特定的分割点

决策树的剪枝

由于决策树构建完成需要让数据集中的内容为空或者唯一，所以容易引起过拟合，需要对其进行剪枝剪枝的方法分为**预剪枝**和**后剪枝**

- 预剪枝

  预剪枝是在生成决策树的同时，对树的分叉节点进行判断，如果节点分叉之后的树在验证集上的表现比不过未分叉的树的表现，则不对此节点进行分叉。

  > 预剪枝的好处是能减少过拟合，而且训练的时间也会较少。但是有的时候在当前分支表现的不好，可能会在之后的分支有更好的表现，预剪枝无法对这种情况进行判断，有时候会得到欠拟合的结果。

- 后剪枝

  先得到一个完整的决策树，然后从最底层节点开始向上遍历，如果某个节点被剪枝之后的验证集效果高于未被剪枝的效果，那就可以对其进行剪枝。

  > 后剪枝的欠拟合风险小，泛化性能更好，但是需要完全训练决策树，而且针对每一个底层节点都要进行考察，时间成本稍高。

连续值的划分

对于决策树的连续值，可以先得到划分点的集合，然后从集合中选择出使得信息增益最高的划分点，划分点的

Missing or unrecognized delimiter for \left

即取得每两个相邻的值的中间点

信息增益的划分方式为： $$ \begin{aligned} Gain(D,a) &= \max_{t\in T_a}Gain(D,a,t)\ &=\max_{t\in T_a}Ent(D) - \sum_{\lambda\in{-,+}}\frac{|D_t^\lambda|}{|D|}Ent(D_t^\lambda)

\end{aligned}

$$

- 多变量决策树

  与的时候，多个连续值变量存在于决策树之中，如果按照单一变量进行判断和分支，那么得到的结果会比较复杂，尤其是模型中的点是线性可分的时候，这时候要使用多变量的决策树，（使用线性模型让多个变量合成一个变量）

支持向量机

无监督模型

层次聚类

K均值聚类

高斯混合模型

LDA

- Gamma 分布

$$\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}dt$$

  Gamma函数可以看作是阶乘在实数集上的延拓因为其具有如下性质

$$\Gamma(x+1) = x\Gamma(x)$$

- Beta分布 假设函数$B(\alpha,\beta)$表示如下含义：

$$\frac{1}{B(\alpha,\beta)} = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}$$

  β分布的表示如下

$$f(x;\alpha,\beta) = \frac{1}{B(\alpha,\beta)}x^{\aleph-1}(1-x)^{\beta-1}$$

- 狄利克雷(Dirichlet)分布

  Dirichlet分布的概率密度函数为：

$$f(x_1,x_2,\ldots x_k;\alpha_1,\alpha_2,\ldots\alpha_k) = \frac{1}{B(\alpha)}\prod_{i=1}^k x_i^{\alpha^i-1}$$

其中，

$$B(\alpha) = \frac{\prod_{i=1}^{k}\Gamma(\alpha^i)}{\Gamma(\sum_{i=1}^{k}\alpha^i)}, \sum_{i=1}^{k} x_i = 1$$

根据Beta分布，二项分布，Dirichlet分布以及多项式分布的共识可以看出，Beta分布是二项分布的共轭先验分布，Dirichlet分布是多项分布的共轭先验分布

Beta分布以及Dirichlet分布的期望具有如下性质：

$$ \begin{aligned} E(p) & = \int_0^1t*Beta(t|\alpha,\beta)dt \ &=\int_0^1 t\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} t^{\alpha-1}(1-t)^{\beta-1} \ &=\int_0^1 \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} t^{\alpha}(1-t)^{\beta-1}\ &= \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\int_0^1 t^{\alpha}(1-t)^{\beta-1} \end{aligned} $$

对于其中的右边部分，可以看出与$Beta(t|\alpha+1,\beta)$的表达式相似，现在对其概率分布进行积分有如下表达式：

$$\int_0^1 Beta(t|\alpha+1,\beta) = \int_0^1 \frac{\Gamma(\alpha+\beta+1)}{\Gamma(\alpha+1)\Gamma(\beta)} t^{\alpha}(1-t)^{\beta-1} = 1$$

根据Gamma函数的性质，可以将上式改写为：

$$\int_0^1 \frac{\Gamma(\alpha+\beta)(\alpha+\beta)}{\Gamma(\alpha)\alpha\Gamma(\beta)} t^{\alpha}(1-t)^{\beta-1} = 1$$

$$ \int_0^1 \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} t^{\alpha}(1-t)^{\beta-1}=\frac{\alpha}{(\alpha+\beta)}

带入前式，可以得到$E(p)$的表达式为：

$$E(p) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \int_0^1 t^{\alpha}(1-t)^{\beta-1} \qquad = \int_0^1 \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} t^{\alpha}(1-t)^{\beta-1} = \frac{\alpha}{(\alpha+\beta)}$$

这说明$Beta$分布的变量的均值可以用$\frac{\alpha}{\alpha+\beta}$来估计，类似的结论扩展到$Dirichlet$分布上可以得到

E(p) = \left(\frac{\alpha^1}{\sum_{i=1}^K\alpha_i},\frac{\alpha^2}{\sum_{i=1}^K\alpha_i},...,\frac{\alpha^K}{\sum_{i=1}^K\alpha_i}\right)

$$ 所以Dirichlet分布中的每一个类别都有一个估计值。

MCMC 与 Gibbs采样

MCMC即马尔可夫链蒙特卡洛采样(Markov Chain Monte Carlo) 马尔可夫链的假设是，每一个当前状态只依赖于前一个状态，状态之间的变化借助一个状态转移矩阵实现，MH算法是其最基础的形式： 输入：先验概率 $Q(x^{\ast} \mid x^{t-1})$ 过程：

1. 初始化x^0
2. for t = 1, 2, ... do
3. 根据 $Q(x^{\ast} \mid x^{t-1})$采样出候选样本 $x^{\ast}$
4. 根据均匀分布从(0, 1)范围内采样出阈值u;

5. if $u \le A(x^{\ast} \mid x^{t-1})$ then
6. $x^t = x^{\ast}$
7. else
8. $x^t = x^{t-1}$
9. end if
10. enf for
11. return $x^1, x^2, ...$ 输出：采样出的一个样本序列 于是, 为了达到平稳状态，只需将接受率设置为 $$ A(x^|x^{t-1}) = min\left(1,\frac{p(x^Q(X^{t-1}|x^))}{p(x^{t-1}Q(X^{}|x^{t-1}))}\right) $$ 其中先验概率 $Q(x^{\ast} \mid x^{t-1})$表示的就是马尔可夫链中的状态转移矩阵，即在上一步的采样结果x^i的条件下下一步的采样结果的概率分布，本质上是一个根据样本求分布的函数。 而Gibbs采样的方法呢，则是根据xi的现有取值，计算条件概率
12. 随机或以某个次序选取某变量x_i;
13. 根据x中除x_i外的变量的现有取值，计算条件概率$p(x_i \mid X_i)$, 其中$X_i = {x_1, x_2, \cdots, x_{i-1}, x_{i+1}, \cdots, x_N}$;
14. 根$p(x_i \mid X_i)$据对变量x_i采样，用采样值代替原值.

而在LDA当中，需要估计的样本就是文档到主题的分布参数以及主题到单词的分布参数，在计算的时候，目标是使得Gibbs采样收敛，收敛的标志是每一次采样之后，单词的主题信息都不变，所以每一次采样的过程都要进行下面的操作。

1. 计算出每一个单词的主题分布
2. 选出具有最高概率的主题作为单词的主题
3. 如果单词的主题和上一次没有变化则收敛

所以Gibbs采样的要解决的问题就是根据样本标注当前的状态确定下一步的样本标注状态，在这个过程中，随着样本标注的变化，分布也会发生变化。

根据上述步骤，可以确定每一个迭代过程中，每一个样本的主题的概率值为

$$p(z_i = k \mid \vec{z}_{\neg i}, \vec{w}) \propto p(z_i = k, w_i = t \mid \vec{z}_{\neg i}, \vec{w}_{\neg i})$$

为了求得对应的概率值，首先要确定对应的文档到主题，主题到单词的分布，然后使用积分求得对应的概率值，应用文档主题以及主题单词的两个Dirichlet分布，得到如下公式：

$$p(\vec{\theta}_m \mid \vec{z}_{\neg i}, \vec{w}_{\neg i}) = Dir(\vec{\theta}_m \mid \vec{n}_{m,\neg i} + \vec{\alpha})$$

$$p(\vec{\varphi}_k \mid \vec{z}_{\neg i}, \vec{w}_{\neg i}) = Dir(\vec{\varphi}_k \mid \vec{n}_{k,\neg i} + \vec{\beta})$$

然后对每个单词属于的主题概率进行推导： $$ \begin{aligned} &p(z_i = k \mid \vec z_{\lnot i}, \vec w) \propto p(z_i = k, w_i = t \mid \vec z_{\lnot i}, \vec w_{\lnot i}) \ & = \int p(z_i = k, w_i = t, \vec \theta_m, \vec \varphi_k \mid \vec z_{\lnot i}, \vec w_{\lnot i})d\vec \theta_m d \vec \varphi_k \

& = \int p(z_i = k, \vec \theta_m, \mid \vec z_{\lnot i}, \vec w_{\lnot i}) \cdot p(w_i = t, \vec \varphi_k, \mid \vec z_{\lnot i}, \vec w_{\lnot i}) d\vec \theta_m d \vec \varphi_k \

& = \int p(z_i = k \mid \vec \theta_m) p(\vec \theta_m \mid \vec z_{\lnot i}, \vec w_{\lnot i}) \cdotp(w_i = t \mid \vec \varphi_k)p(\vec \varphi_k \mid \vec z_{\lnot i}, \vec w_{\lnot i}) d\vec \theta_m d \vec \varphi_k \

& = \int p(z_i = k \mid \vec \theta_m) Dir(\vec \theta_m \mid \vec n_{m, \lnot i} + \vec \alpha) d \vec \theta_m

) Dir(\vec \varphi_k \mid \vec n_{k, \lnot i} + \vec \beta) d \vec \varphi_k \

& = \int \theta_{mk} Dir(\vec \theta_m \mid \vec n_{m, \lnot i} + \vec \alpha) d \vec \theta_m \cdot \int \varphi_{kt} Dir(\vec \varphi_k \mid \vec n_{k, \lnot i} + \vec \beta) d \vec \varphi_k \

&= E(\theta_{mk}) \cdot E(\varphi_{kt}) \

&= \hat \theta_{mk} \cdot \hat \varphi_{kt}

\end{aligned} $$

可以发现每一个单词属于某一个主题的概率值就等于这两个估计的乘积：

$$

\hat{\theta}***{mk}*** = ***frac{n***{m,\neg i}^{(k)} + \alpha_k}{\sum_{k=1}^K (n_{m,\neg i}^{(k)} + \alpha_k)}

\hat{\varphi}***{kt}*** = ***frac{n***{k,\neg i}^{(t)} + \beta_t}{\sum_{t=1}^V (n_{k,\neg i}^{(t)} + \beta_t)} $$

由此可以看出，要求得每一次的单词的主题分布，只要对上一次的文档-主题，主题-单词的数量进行统计就可以了。

所以Gibbs采样的过程就是一步一步的生成这些概率，最终样本集稳定的时候，就代表模型收敛

LDA 使用两个狄利克雷模型来生成文档到主题，主题到词语的分布，借助这两个分布来生成主题到

采样

蒙特卡洛采样

蒙特卡洛采样的思路是按照某一个变量或者概率分布进行随机化采样，通过结果计算某一个特定的值，如均值，如果采样的区间是均匀分布的，直接使用均匀分布采样即可，如果不是均匀分布的，那么就要计算在$q(x)$分布下目标值，使用积分或者求和的方式。

集成学习

概念

在机器学习的有监督学习算法中，我们的目标是学习出一个稳定的且在各个方面表现都较好的模型，但实际情况往往不这么理想，有时我们只能得到多个有偏好的模型（弱监督模型，在某些方面表现的比较好）。集成学习就是组合这里的多个弱监督模型以期得到一个更好更全面的强监督模型，集成学习潜在的思想是即便某一个弱分类器得到了错误的预测，其他的弱分类器也可以将错误纠正回来。

集成学习在各个规模的数据集上都有很好的策略、

Bagging

是bootstrap aggregating的缩写，bootstrap也称作自助法，又放回的抽样方法

- 采用重抽样方法（有放回抽样）从原始样本中抽取一定数量的样本
- 根据抽出的样本计算想要得到的统计量T
- 重复上述N次（一般大于1000），得到N个统计量T
- 算出统计量的置信区间

Boosting

boosting称为提升方法，减少监督学习偏差的学习算法，其中有代表性的有两种，分别是GDBT和AdaBoost

> AdaBoost:(Adaptive Boost)在一开始训练的时候对所有的训练数据赋予相同的权重，但是在每轮训练失败之后都会为失败的样本增加权重

1. begin initial D={x1，y1，...，xn，yn}，kmax（最大循环次数），Wk(i)=1/n，i=1，...，n
2. k ← 0
3. do k ← k+1
4. 训练使用按照Wk(i)采样的D的弱学习器Ck
5. Ek ←对使用Wk(i)的D测量的Ck的训练误差
6. $\alpha *{k}\gets {\tfrac {1}{2}}\ln {\frac {1-E*{k}}{E_{k}}}$
7. $W_{{k+1}}(i)\gets {\dfrac {W_{{k}}(i)}{Z_{{k}}}}\times {\begin{cases}e^{{-\alpha *{{k}}}},&{{if }}h{{k}}(x^{{i}})=y_{{i}}\e^{{\alpha *{{k}}}},&{{if }}h{{k}}(x^{{i}})\neq y_{{i}}\end{cases}}$
8. until k=kmax
9. return Ck和$α_k，k=1，...，kmax$（带权值分类器的总体）
10. end

GBDT:(Gradient Boost Decision Tree)

XGboost

是一个开源的GBDT框架，用于实现高速的GBDT

深度学习

前向传播算法 BP

$$\frac{\partial E_k}{\partial w} = \frac{\partial E_k}{\partial \sigma}\frac{\partial \sigma}{\partial w}$$

在神经网络的每一层，都可以得到某个参数在当前损失函数条件下的梯度值，这个梯度值与

1. 当前层的输出
2. 当前层的输入
3. 从下一层传来的梯度值

有关

而使用批量梯度下降的时候，由于每一次梯度更新对于参数的变化都是独立的而且是加法变换，所以批量梯度需要计算每一次的梯度，在更新的时候进行求和，每个样本的梯度计算可以是并行的。

RBF(Radical basis Function,径向基函数)

假设任意网络都可以使用若干个径向对称的基函数求和表示

CNN

深度残差网络

RNN

> RNN具有的梯度消失的问题:
>
> 由于在BPTT计算的时候，每一个步骤的隐层状态h和前一个隐层状态之间的参数会被多次连乘，所以这个参数值一旦是小的值就会引起梯度消失，大的值会引起梯度爆炸

$$S_t = \sigma(W_h X_t + U_h S_{t-1} + b_h)\ O_t = \sigma(W_o S_t + b_o)$$

$$\frac{\partial O_t}{\partial W_o} = \frac{\partial O_t}{\partial \sigma}\frac{\partial \sigma}{\partial W_o} = \frac{\partial O_t}{\partial \sigma} S_t$$

$$\frac{\partial O_t}{\partial W_h} = \frac{\partial O_t}{\partial \sigma}\frac{\partial \sigma}{\partial W_h} = \frac{\partial O_t}{\partial \sigma} X_t$$

$$\frac{\partial O_t}{\partial U_h} = \frac{\partial O_t}{\partial \sigma}\frac{\partial \sigma}{\partial U_h} = \frac{\partial O_t}{\partial \sigma}(S_t + \frac{\partial S_t}{\partial U_t})$$

长短期记忆网络(LSTM)

激活函数

- Sigmoid
- Softmax
- Tanh
- Relu

批量归一化(Batch Normalize)

Dropout

特征工程

特征归一化

文本的表示方式

TF-IDF

tf-idf（英语：term frequency–inverse document frequency）是一種用於資訊檢索與文本挖掘的常用加權技術。tf-idf是一種統計方法，用以評估一字詞對於一個文件集或一個語料庫中的其中一份文件的重要程度。字詞的重要性隨著它在文件中出現的次數成正比增加，但同時會隨著它在語料庫中出現的頻率成反比下降。tf-idf加權的各種形式常被搜索引擎應用，作為文件與用戶查詢之間相關程度的度量或評級。除了tf-idf以外，互聯網上的搜尋引擎還會使用基於連結分析的評級方法，以確定文件在搜尋結果中出現的順序。

textRank

word2vec

高维特征的处理

PCA

主成分分析法的目标是，对原数据集降维之后，尽可能的得到具有较大方差的数据

使用的方法是构建一组新的坐标系${w_1,w_2,...,w_d}$，使得在这个坐标系上面的投影

奇异值分解(SVD)

模型评估

评价指标

- 准确率
- 召回率
- F值
  - macro F值
  - micro F值
- 其他指标
  - ROUGE
  - BLEU

ROC 曲线

ROC曲线的横坐标是

$$\frac{FP}{N}$$

纵坐标是:

$$\frac{TP}{P}$$

反映出了一个分类器在调节选择阈值的时候，选出正确结果和排除错误结果的综合能力。

余弦相似度以及余弦距离

评估方法(Evaluation)

Holdout

交叉验证

自助法(Bootstrap)

超参数调优

过拟合于欠拟合

表现

解决方法