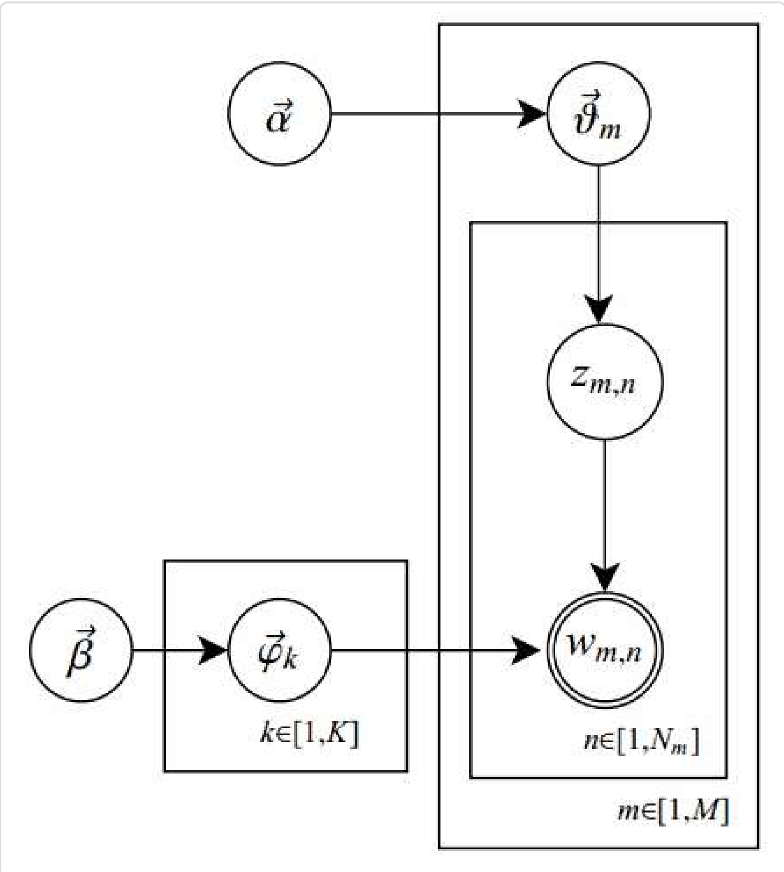
BLOG | 逍遥郡

主题模型之LDA

2013-02-23 15:25

在原始的pLSA模型中，我们求解出两个参数：“主题-词项”矩阵 Φ 和“文档-主题”矩阵 Θ ，但是我们并未考虑参数的先验知识；而LDA的改进之处，是对这俩参数之上分别增加了先验分布，相应参数称作超参数(hyperparamter)。概率图表示如下：



其中，单圆圈表示隐变量；双圆圈表示观察到的变量；把节点用方框(plate)圈起来，表示其中的节点有多种选择。所以这种表示方法也叫做plate notation，具体可参考PRML 8.0 Graphical Models。

对应到上图，只有 $w_{m,n}$ 是观察到的变量，其他都是隐变量或者参数，其中 $\vec{\alpha}$ 和 $\vec{\beta}$ 是超参数；方框中，$\Phi=\{\vec{\varphi}_k\}_{k=1}^K$ 表示有 K 种“主题-词项”分布； $\Theta=\{\vec{\vartheta}_m\}_{m=1}^M$ 有 M 种“文档-主题”分布，即对每篇文档都会产生一个 $\vec{\vartheta}_m$ 分布；每篇文档 m 中有 n 个词，每个词 $w_{m,n}$ 都有一个主题 $z_{m,n}$ ，该词实际是由 $\vec{\varphi}_{z_{m,n}}$ 产生。具体生成过程下面再说。

§ 贝叶斯估计

在pLSA原参数之上增加先验分布，其实就是用贝叶斯估计取代最大似然估计，具体要了解各种参数估计方法可以参考Heinrich论文的第二部分。简单说，最大似然估计(MLE)和最大后验估计(MAP)都是把待估计的参数看作一个拥有固定值的变量，只是取值未知。通常估计的方法都是找使得相应的函数最大时的参数；由于MAP相比于MLE会考虑先验分布的影响，所以MAP也会有超参数，它的超参数代表的是一种信念(belief)，会影响推断(inference)的结果。比如说抛硬币，如果我先假设是公平的硬币，这也是一种归纳偏置(bias)，那么最终推断的结果会受我们预先假设的影响。贝叶斯估计是对MAP的扩展，但它不再对参数做直接的估计，而是把待估计的参数看作服从某种分布的随机变量。根据贝叶斯法则：

$$\text{posterior}=\frac{\text{likelihood}\cdot\text{prior}}{\text{evidence}}$$

即

$$p(\vartheta|\mathcal{X})=\frac{p(\mathcal{X}|\vartheta)\cdot p(\vartheta)}{p(\mathcal{X})}$$

在MLE和MAP中，由于是要求函数最大值时的参数，所以都不会考虑evidence。但在贝叶斯估计中，不再直接取极值，所以还会考虑evidence，下面的这个积分也是通常贝叶斯估计中最难处理的部分：

$$p(\mathcal{X})=\int_{\vartheta\in\Theta}p(\mathcal{X}|\vartheta)p(\vartheta)\mathrm{d}\vartheta$$

evidence相当于对所有的似然概率积分或求和(离散时)，所以也称作边界似然。

§ 共轭先验

由于有积分的存在，贝叶斯估计常常会很难推算，这里我们就需要利用一种共轭先验(Conjugate Prior)的数学知识。在贝叶斯统计理论中，如果某个随机变量 ϑ 的先验分布 $p(\vartheta)$ 和后验分布 $p(\vartheta|\mathcal{X})$ 属于统一分布簇(也就是说有同样的函数形式)，则称先验分布 $p(\vartheta)$ 和后验分布 $p(\vartheta|\mathcal{X})$ 为共轭分布，先验分布 $p(\vartheta)$ 是似然函数 $p(\mathcal{X}|\vartheta)$ 的共轭先验。

$$\begin{aligned}\mathrm{Dir}(\vec{p}|\vec{\alpha}) &\triangleq \frac{\Gamma\left(\sum_{k=1}^{K}\alpha_k\right)}{\prod_{k=1}^{K}\Gamma(\alpha_k)}\prod_{k=1}^{K}p_k^{\alpha_k-1} \\ &\triangleq \frac{1}{\Delta(\vec{\alpha})}\prod_{k=1}^{K}p_k^{\alpha_k-1}\end{aligned} \tag{1}$$

其中， $\vec{p}$ 是要猜测的随机向量， $\vec{\alpha}$ 是超参数， $\Delta(\vec{\alpha})$ 称作Delta函数，可以看作Beta函数的多项式扩展，是Dirichlet分布的归一化系数，定义如下：

$$\Delta(\vec{\alpha}) = \frac{\prod_{k=1}^{\dim\vec{\alpha}}\Gamma(\alpha_k)}{\Gamma\left(\sum_{k=1}^{\dim\vec{\alpha}}\alpha_k\right)} = int\prod_{k=1}^{V}p_k^{\alpha_k-1}\mathrm{d}\vec{p} \tag{2}$$

相应的多项分布定义：

$$\mathrm{Mult}(\vec{n}|\vec{p},N) \triangleq \binom{N}{\vec{n}}\prod_{k=1}^{K}p_k^{n_k} \tag{3}$$

其中， $\vec{p}$ 和 $\vec{n}$ 服从约束 $\sum_k p_k = 1$ 和 $\sum_k n_k = N$ 。

由于 $\Gamma(x+1) = x!$ 就是阶乘在实数集上的扩展，显然，公式(1)和(3)有相同的形式，所以，这俩分布也称作Dirichlet-Multinomail共轭。

如果 $\vec{p} \sim \mathrm{Dir}(\vec{p}|\vec{\alpha})$ ，则 $\vec{p}$ 中的任一元素 p_i 的期望是：

$$\begin{aligned}E(p_i) &= \int_0^1 p_i \cdot \mathrm{Dir}(\vec{p}|\vec{\alpha})\mathrm{d}p \\ &= \frac{\Gamma\left(\sum_{k=1}^{K}\alpha_k\right)}{\Gamma(\alpha_i)} \cdot \frac{\Gamma(\alpha_i+1)}{\Gamma\left(\sum_{k=1}^{K}\alpha_k+1\right)} \\ &= \frac{\alpha_i}{\sum_{k=1}^{K}\alpha_k}\end{aligned} \tag{4}$$

可以看出，超参数 α_k 的直观意义就是事件先验的伪计数(prior pseudo-count)。

§ LDA生成模型

在LDA中，“文档-主题”向量 $\vec{\vartheta}_m$ 由超参数为 $\vec{\alpha}$ 的Dirichlet分布生成，“主题-词项”向量 $\vec{\varphi}_k$ 由超参数为 $\vec{\beta}$ 的Dirichlet分布生成，根据概率图，整个样本集合的生成过程如下：

- 生成当前文档 m 相应的“文档-主题”分布 $\vec{\vartheta}_m \sim \mathrm{Dir}(\vec{\alpha})$（$K$ 维向量，即第 m 篇文档对应的每个主题的概率)
- 生成当前文档 m 的长度 $N_m \sim \mathrm{Poiss}(\xi)$
- 对当前文档 m 中的所有词 $n \in [1, N_m]$ ：
  - 生成当前位置的词的所属主题 $z_{m,n} \sim \mathrm{Mult}(\vec{\vartheta}_m)$
  - 根据之前生成的主题分布 Φ ，生成当前位置的词的相应词项 $w_{m,n} \sim \mathrm{Mult}(\vec{\varphi}_{z_{m,n}})$

由该生成过程可知，第 m 篇文档中第 n 个词 t 的生成概率：

$$p(w_{m,n}=t|\vec{\vartheta}_m,\Phi)=\sum_{k=1}^{K}p(w_{m,n}=t|\vec{\varphi}_k)p(z_{m,n}=k|\vec{\vartheta}_m)$$

其中， $\Phi=\{\vec{\varphi}_k\}_{k=1}^{K}$ 。

根据所有已知信息和带超参数的隐变量，我们可以写出联合分布：

$$p(\vec{w}_m,\vec{z}_m,\vec{\vartheta}_m,\Phi|\vec{\alpha},\vec{\beta})=\overbrace{\underbrace{\prod_{n=1}^{N_m}p(w_{m,n}|\vec{\varphi}_{z_{m,n}})p(z_{m,n}|\vec{\vartheta}_m)}_{\text{word plate}}\cdot p(\vec{\vartheta}_m|\vec{\alpha})}^{\text{document plate (1 document)}}\cdot\underbrace{p(\Phi|\vec{\beta})}_{\text{topic plate}}$$

通过对 $\vec{\vartheta}_m$ 和 Φ 积分以及 $z_{m,n}$ 求和，可以求得 $\vec{w}_{m,n}$ 的分布：

$$\begin{aligned}p(\vec{w}_m|\vec{\alpha},\vec{\beta})&=\int\int p(\vec{\vartheta}_m|\vec{\alpha})\cdot p(\Phi|\vec{\beta})\cdot\prod_{n=1}^{N_m}\sum_{z_{m,n}}p(w_{m,n}|\vec{\varphi}_{z_{m,n}})p(z_{m,n}|\vec{\vartheta}_m)\mathrm{d}\Phi\mathrm{d}\vec{\vartheta}_m\\&=\int\int p(\vec{\vartheta}_m|\vec{\alpha})\cdot p(\Phi|\vec{\beta})\cdot\prod_{n=1}^{N_m}p(w_{m,n}|\vec{\vartheta}_m,\Phi)\mathrm{d}\Phi\mathrm{d}\vec{\vartheta}_m\end{aligned}$$

整个样本的分布：

$$p(\mathcal{W}|\vec{\alpha},\vec{\beta})=\prod_{m=1}^{M}p(\vec{w}_m|\vec{\alpha},\vec{\beta})$$

符号解释：

- M 文档数(固定值)
- K 主题(component)数(固定值)

- N_m 文档 m 的长度，这里由Poisson分布决定
- $z_{m,n}$ 文档 m 中第 n 个词所属的主题
- $w_{m,n}$ 文档 m 中第 n 个词的词项

§ Gibbs Sampling

Blei的原始论文使用变分法(Variational inference)和EM算法进行贝叶斯估计的近似推断，但不太好理解，并且EM算法可能推导出局部最优解。Heinrich使用了Gibbs抽样法，这也是目前LDA的主流算法。

通常均匀分布 $\text{Uniform}(0,1)$ 的样本，即我们熟悉的类 `rand()` 函数，可以由线性同余发生器生成；而其他的随机分布都可以在均匀分布的基础上，通过某种函数变换得到，比如，正态分布可以通过Box-Muller变换得到。然而，这种变换依赖于计算目标分布的积分的反函数，当目标分布的形式很复杂，或者是高维分布时，很难简单变换得到。

当一个问题无法用分析的方法来求精确解，此时通常只能去推断该问题的近似解，而随机模拟(MCMC)就是求解近似解的一种强有力的方法。随机模拟的核心就是对一个分布进行抽样(Sampling)。随机模拟也可用于类pLSA算法，但现在很少有人这么做。

MCMC的基础：Markov链通过转移概率矩阵可以收敛到稳定的概率分布。这意味着MCMC可以借助Markov链的平稳分布特性模拟高维概率分布 $p(\vec{x})$ ；当Markov链经过burn-in阶段，消除初始参数的影响，到达平稳状态后，每一次状态转移都可以生成待模拟分布的一个样本。Gibbs抽样是MCMC的一个特例，它交替的固定某一维度 x_i ，然后通过其他维度 $\vec{x}_{\neg i}$ 的值来抽样该维度的值。它的基本算法如下：

1. 选择一个维度 i ，可以随机选择；
2. 根据分布 $p(x_i|\vec{x}_{\neg i})$ 抽样 x_i 。

所以，如果要完成Gibbs抽样，需要知道如下条件概率：

$$p(x_i|\vec{x}_{\neg i}) = \frac{p(\vec{x})}{p(\vec{x}_{\neg i})} = \frac{p(\vec{x})}{\int p(\vec{x})\mathrm{d}x_i}, \quad \vec{x} = \{x_i, \vec{x}_{\neg i}\}$$

如果模型包含隐变量 $\vec{z}$ ，通常需要知道后验概率分布 $p(\vec{z}|\vec{x})$ ，所以，包含隐变量的Gibbs抽样器公式如下：

$$p(z_i|\vec{z}_{\neg i}, \vec{x}) = \frac{p(\vec{z}, \vec{x})}{p(\vec{z}_{\neg i}, \vec{x})} = \frac{p(\vec{z}, \vec{x})}{\int_Z p(\vec{z}, \vec{x})\mathrm{d}z_i} \tag{5}$$

§ LDA Gibbs Sampler

为了构造LDA Gibbs抽样器，我们需要使用隐变量的Gibbs抽样器公式。在LDA模型中，隐变量为 $z_{m,n}$ ，即样本中每 的主 参数 等 以 过 察到的 相应的 分求得 这种处

这里省略了超参数，这个分布涉及很多离散随机变量，并且分母是 K_W 个项的求和，很难求解。此时，就需要Gibbs sampling发挥用场了，我们期望Gibbs抽样器可以通过Markov链利用全部的条件分布 $p(z_i|\vec{z}_{\neg i},\vec{w})$ 来模拟 $p(\vec{z}|\vec{w})$ 。根据公式(4)，我们需要写出联合概率分布 $p(\vec{w},\vec{z})$ ：

$$p(\vec{w},\vec{z}|\vec{\alpha},\vec{\beta})=p(\vec{w}|\vec{z},\vec{\beta})p(\vec{z}|\vec{\alpha})$$

由于此公式第一部分独立于 $\vec{\alpha}$ ，第二部分独立于 $\vec{\beta}$ ，所以可以分别处理。

第一部分，可以由观察到的词数以及相应主题的多项分布产生：

$$p(\vec{w}|\vec{z},\Phi)=\prod_{i=1}^{W}p(w_i|z_i)=\prod_{i=1}^{W}\varphi_{z_i,w_i}$$

由于样本中的 W 个词服从参数为主题 z_i 的独立多项分布，这意味着，我们可以把上面的对词的乘积分解成对主题和对词项的两层乘积：

$$p(\vec{w}|\vec{z},\Phi)=\prod_{k=1}^{K}\prod_{\{i:z_i=k\}}p(w_i=t|z_i=k)=\prod_{k=1}^{K}\prod_{t=1}^{V}\varphi_{k,t}^{n_k^{(t)}} \tag{6}$$

其中， $n_k^{(t)}$ 是词项 t 在主题 k 中出现的次数。目标分布 $p(\vec{w}|\vec{z},\vec{\beta})$ 需要对 Φ 积分，根据 $\Delta(\vec{\alpha})$ 函数公式(2)可得：

$$\begin{aligned}p(\vec{w}|\vec{z},\vec{\beta})&=\int p(\vec{w}|\vec{z},\Phi)p(\Phi|\vec{\beta})\mathrm{d}\Phi\\&=\int\prod_{z=1}^{K}\frac{1}{\Delta(\vec{\beta})}\prod_{t=1}^{V}\varphi_{z,t}^{n_z^{(t)}+\beta_t-1}\mathrm{d}\vec{\varphi}_z\\&=\prod_{z=1}^{K}\frac{\Delta(\vec{n}_z+\vec{\beta})}{\Delta(\vec{\beta})},\quad\vec{n}_z=\{n_z^{(t)}\}_{t=1}^{V}\end{aligned}$$

这个结果可以看作 K 个Dirichlet-multinomial模型的乘积。

第二部分，类似于 $p(\vec{w}|\vec{z},\vec{\beta})$ 的步骤，先写出条件分布，然后分解成两部分的乘积：

$$p(\vec{z}|\Theta)=\prod_{i=1}^{W}p(z_i|d_i)=\prod_{m=1}^{M}\prod_{k=1}^{K}p(z_i=k|d_i=m)=\prod_{m=1}^{M}\prod_{k=1}^{K}\theta_{m,k}^{n_m^{(k)}} \tag{7}$$

其中， d_i 是单词 i 所属的文档， $n_m^{(k)}$ 是主题 k 在文章 m 中出现的次数。对 Θ 积分可得：

$$p(\vec{z}|\vec{\alpha})=\int p(\vec{z}|\Theta)p(\Theta|\vec{\alpha})\mathrm{d}\Theta$$

$$\int\prod^{M}\ 1\ \prod^{K}\ n^{(k)}+\alpha_k\ 1\ \to$$

根据联合分布，求解下标为 $i=(m,n)$ 的词，即第 m 篇文档中的第 n 个词，的全部的条件概率。令 $\vec{w}=\{w_i=t,\vec{w}_{\neg i}\}$ ， $\vec{z}=\{z_i=k,\vec{z}_{\neg i}\}$ ，有：

$$\begin{aligned}
p(z_i=k|\vec{z}_{\neg i},\vec{w}) &= \frac{p(\vec{w},\vec{z})}{p(\vec{w},\vec{z}_{\neg i})} \\
&= \frac{p(\vec{w},\vec{z})}{p(\vec{w}_{\neg i}|\vec{z}_{\neg i})p(w_i)} \cdot \frac{p(\vec{z})}{p(\vec{z}_{\neg i})} \\
&\propto \frac{\Delta(\vec{n}_z+\vec{\beta})}{\Delta(\vec{n}_{z,\neg i}+\vec{\beta})} \cdot \frac{\Delta(\vec{n}_m+\vec{\alpha})}{\Delta(\vec{n}_{m,\neg i}+\vec{\alpha})} \\
&= \frac{\Gamma(n_k^{(t)}+\beta_t)\Gamma(\sum_{t=1}^V n_{k,\neg i}^{(t)}+\beta_t)}{\Gamma(n_{k,\neg i}^{(t)}+\beta_t)\Gamma(\sum_{t=1}^V n_k^{(t)}+\beta_t)} \cdot \frac{\Gamma(n_m^{(k)}+\alpha_k)\Gamma(\sum_{k=1}^K n_{m,\neg i}^{(k)}+\alpha_t)}{\Gamma(n_{m,\neg i}^{(k)}+\alpha_t)\Gamma(\sum_{k=1}^K n_m^{(k)}+\alpha_k)} \\
&= \frac{n_{k,\neg i}^{(t)}+\beta_t}{\sum_{t=1}^V n_{k,\neg i}^{(t)}+\beta_t} \cdot \frac{n_{m,\neg i}^{(k)}+\alpha_k}{[\sum_{k=1}^K n_m^{(k)}+\alpha_k]-1} \\
&\propto \frac{n_{k,\neg i}^{(t)}+\beta_t}{\sum_{t=1}^V n_{k,\neg i}^{(t)}+\beta_t}(n_{m,\neg i}^{(k)}+\alpha_k)
\end{aligned} \tag{8}$$

最终，我们需要根据Markov链的状态 z_i 获取多项分布的参数 Θ 和 Φ 。根据贝叶斯法则和Dirichlet先验，以及公式(6)和(7)：

$$\begin{aligned}
p(\vec{\vartheta}_m|\vec{z}_m,\vec{\alpha}) &= \frac{1}{Z_{\vartheta_m}}\prod_{n=1}^{N_m} p(z_{m,n}|\vec{\vartheta}_m)\cdot p(\vec{\vartheta}_m|\vec{\alpha}) = \mathrm{Dir}(\vec{\vartheta}_m|\vec{n}_m+\vec{\alpha}) \\
p(\vec{\varphi}_k|\vec{z},\vec{w},\vec{\beta}) &= \frac{1}{Z_{\varphi_k}}\prod_{\{i:z_i=k\}} p(w_i|\vec{\varphi}_k)\cdot p(\vec{\varphi}_k|\vec{\beta}) = \mathrm{Dir}(\vec{\varphi}_k|\vec{n}_k+\vec{\beta})
\end{aligned}$$

其中， $\vec{n}_m$ 是构成文档 m 的主题数向量， $\vec{n}_k$ 是构成主题 k 的词项数向量。求解Dirichlet分布的期望，即公式(4)可得：

$$\varphi_{k,t} = \frac{n_k^{(t)}+\beta_t}{\sum_{t=1}^V n_k^{(t)}+\beta_t} \tag{9}$$

$$\vartheta_{m,k} = \frac{n_m^{(k)}+\alpha_k}{\sum_{k=1}^K n_m^{(k)}+\alpha_k} \tag{10}$$

梳理一下Gibbs sampling中所用到的数据结构：统计量 $n_m^{(z)}$ 和 $n_z^{(t)}$ 分别是 $M\times K$ 和 $K\times V$ 矩阵，它们每行的和分别是 M 维向量 n_m (文档长度)和 K 维向量 n_z 。Gibbs sampling算法有三个阶段：初始化、burn-in和sampling。具体算法如下：

算法 $(\{\vec{\ }\ \})$

- 设置全局变量 $n_m^{(k)}$ 、 $n_k^{(t)}$ 、 n_m 、 n_k 为零
- 对所有文档 $m \in [1, M]$ ：
    - 对文档 m 中的所有单词 $n \in [1, N_m]$ ：
        - 初始化每个单词对应的主题 $z_{m,n} = k \sim \mathrm{Mult}(1/K)$
        - 增加“文档-主题”计数： $n_m^{(k)} + = 1$
        - 增加“文档-主题”总数： $n_m + = 1$
        - 增加“主题-词项”计数： $n_k^{(t)} + = 1$
        - 增加“主题-词项”总数： $n_k + = 1$

[迭代下面的步骤，直到Markov链收敛]

- 对所有文档 $m \in [1, M]$ ：
    - 对文档 m 中的所有单词 $n \in [1, N_m]$ ：
        - 删除该单词的主题计数： $n_m^{(k)} - = 1; n_m - = 1; n_k^{(t)} - = 1; n_k - = 1;$
        - 根据公式(8)采样出该单词的新主题： $\tilde{k} \sim p(z_i | \vec{z}_{\neg i}, \vec{w})$
        - 增加该单词的新主题计数： $n_m^{(\tilde{k})} + = 1; n_m + = 1; n_{\tilde{k}}^{(t)} + = 1; n_{\tilde{k}} + = 1;$

- 如果Markov链收敛：
    - 根据公式(9)生成主题-词项分布 Φ
    - 根据公式(10)生成文档-主题分布 Θ

§ 参考

- Gregor Heinrich, Parameter estimation for text analysis
- David M.Blei, Andrew Y.Ng, Michael I.Jordan, Latent Dirichlet Allocation
- Philip Resnik, Eric Hardisty, Gibbs Sampling for the Uninitiated <!-- - Yi Wang, Distributed Gibbs Sampling of Latent Topic Models: The Gritty Details -->

ml topic-model

2 Comments **Julian Qian's Home Page** 1 **Login**

Recommend **Share** 按评分高低排序

Join the discussion…

daoluan • 2年前

这是统计学的内容？

Reply • Share ›

Yushneng • 2年前

写得比较清楚！

Reply • Share ›

ALSO ON **JULIAN QIAN'S HOME PAGE**

Hadoop Streaming编程框架mrjob

1 comment • 3年前•

olivetree123 — 数据源如果想用 mysql 或者 mongoldb，要怎么配置呢？

wndr3700刷openwrt固件

2 comments • 4年前•

Julian Qian — 我的是v1。v3改用broadcom芯片了，看起来也没有计划支持：http://wiki.openwrt.org/toh/st...

LRU Cache的C++实现

1 comment • 3年前•

Hector — c++0x 有unorderd_map了，刚也实现了一个 https://github.com/myourys/Alg...

静态绑定和动态绑定

1 comment • 3年前•

tiehkaiwoo — 厲害