财经数据分析工具集文档

朱恬骅，2013年6月

支持环境：

1. Matlab
2. C# Script host CsSC。下载地址：<https://github.com/zhuth/CsSC/archive/master.zip>  
   GitHub地址：<https://github.com/zhuth/CsSC>

CsSC安装方法：将下载来的压缩包解压，找到bin文件夹，可以移动到任何便利的位置并重命名。将该文件夹的位置添加到系统Path环境变量中。

CsSC脚本使用方法：启动cmd命令行，运行：“cssc 脚本文件名 脚本参数”以运行脚本。脚本文件名可以不包含.cssc后缀。

## 字符串处理

请注意：包含在压缩包中，而本文档中未出现的脚本文件仅供参考之用。

WordSegment.exe

盘古分词软件。在界面上作了少量修改，以便于命令行调用。盘古分词的相关信息，请参看<http://pangusegment.codeplex.com/>。

参数：<input> <output> [-pos]

将输入文件<input>读入，用默认设置进行分词，分词结果输出到<output>文件中。如果指定了-pos参数，则在分词文件中同时输出词性信息。

DictManage.exe

盘古分词的词典编辑软件。请阅读盘古分词软件的相关说明文档。

striphtml.cssc

以GBK格式（在第57行”GBK”处定义）打开一文件夹（在第3行files处定义）中所有文件，去除其中所有的HTML代码后，写入原文件，原文件的格式同时改为Unicode。

engbuilddict.cssc

建立英语单词的词典。打开当前工作文件夹中所有”.bsd”后缀名的文件，提取其中所有单词，以获得一个不重复的单词列表，输出在当前工作文件夹的dict.txt文件中。

\_\_getBSDFreq.cssc

以GBK编码打开已分词的文本文件（在第1行file处定义），输出该文本文件中所有中英文单词（不含数字）的列表（文件名为BSD.dict，在第2行output\_dict处定义），并在顺序上按照原输入文件中行的顺序，在每行上以字典文件BSD.dict的顺序，输出file文件中每行对应的词频向量，保存为BSD.fq（在第3行output\_fq处定义）。

worddet.cssc

以论文中所提到的方法，按信息熵和词的内部凝聚程度，提取文本中的词和词频。

参数：<input> <freq output>

将文本文件<input>（Unicode编码）读入，提取文本中的词和词频。

词的最大长度由MaxWordLength定义，为5；输出文件中最小的词频由MinFreq定义，为10。内部凝聚程度由PSvPThreshold定义，为100；左右邻字的信息熵阈值由EntropyThreshold定义，为1.0。

worddetsimple.cssc

只考虑子字符串的频率，来抽取文本中的词和词频。参数和变量含义同worddet.cssc。

tf.cssc

参数：<dict> <input> <output>

用<dict>指定的词典（一行一个词；也可以包含其它信息，以制表符分隔，第一列必须是词），将<input>文件中每一行视作一个文档，得到其词频向量，以相同顺序输出到<output>文件中。该程序可以处理包含超长行的文件。

tfidf.cssc

参数同tf.cssc。输出的向量为TF-IDF值。该程序也可以处理包含超长行的文件。

getLinesCount.cssc

获得超大文件的总行数。

matrixconvert.cssc

转换矩阵的表示格式。

参数：[lda2lsa|lsa2lda|lsa2lda2] <input> <output>

将<input>文件中表示的矩阵转换之后输出为<output>。

第一个参数的意义：

“lda2lsa”：将lda格式（每一行用“列号:值”表示的稀疏阵）展开为完全的矩阵。

“lsa2lda”：将完全矩阵转换为lda格式，其中每行开头***不包括***该行非零值的数量。

“lsa2lda2”：将完全矩阵转换为lda格式，其中每行开头***包括***该行非零值的数量。

BayesianAvg.py

Python版本：2.\*

按论文中所提到的贝叶斯平均方法，读取某文件夹（在第10行names、第17行open(…)处指定，需保持一致）中所有词频文件（文件格式为，每行“词\t词频”），提取分值最高的30个词（在第48行>= 30处定义）输出。每个文件看作一篇文档。

## 数据分析

lda\_exe.m

function [WZ, Z] = lda\_exe(data, k, Wd, alpha)

将Matlab中的矩阵data（每行一个文档），乘以Wd之后，写入临时文件，然后调用lda.exe程序（alpha初值由alpha参数给定）进行LDA，主题数量为k个。得到每主题上词的分布读入WZ矩阵，并以inference推断的结果聚为k类，每个文档的标签输出到Z中。最后清理删除临时文件。如果data为一个字符串，其中为一个lda格式的文本文件，则不会删除该文件。

请注意第8行指定了lda.exe的位置。

clustering.m

function cluster = clustering(model, E)

model是一个struct，包含了原数据（data1、data2）、三元分解的P、Q初值（P、Q）、三元分解结果（P0、A、Q0）。调用时一般不给定E。这时，依据最大似然估计，对model中的原数据进行聚类，选取合适的主题对（pair of topics），并给定标签值。最后的标签值写入cluster中。

dtw.m

function [Dist,D,k,w]=dtw(t,r)

对输入向量t、r进行DTW。

Dist is unnormalized distance between t and r

D is the accumulated distance matrix

k is the normalizing factor

w is the optimal path

evaluateClustering.m

function [fe nouse] = evaluateClustering(clusters, nouse)

计算聚类结果的准确率。输入矩阵clusters有两行，分别表示两个聚类结果，其中任一是正确的。

getEXY.m

function E = getEXY(X, Y, removeZero)

根据输入X、Y，进行normalize，计算矩阵E。如果removeZero设为1，则去除X、Y中全为零的行。

lsqA.m

function A = lsqA(E, P, Q)

根据E和P、Q的当前估计值，估计A。

lsqSolvePAQ.m

function [P A Q] = lsqSolvePAQ(E, A, max\_iters, P, Q, mie)

对E进行三元分解。P、A、Q三个矩阵为初值，其中A矩阵也可以为一个向量[s t]给定大小。参数max\_iters决定了最多迭代的次数。当mie为0时，不对二级目标函数（使A的互信息熵尽可能大）进行优化。

nmi.m

function MIhat = nmi( A, B )

计算NMI。

nnmf3.m

function [P A Q] = nnmf3(E, s, t)

进行两次二元分解以实现三元分解。

normalize.m

function X = normalize(X)

注意：该函数会使Matlab自带的normalize失效。

对矩阵X进行normalize，使其各行的和为1。

pevalc.m

function [cor] = pevalc(data, cluster)

计算各聚类簇的平均皮尔森系数，采用data给定的数据。输出向量表示每一个聚类簇的均值，如果是NaN则表示该聚类簇只有一个元素。同时屏幕上会显示所有非NaN值的平均值。

plsa.m

function [pz pdz pwz pzdw]=plsa(x,k)

对输入数据x，用k个主题运行PLSA，输出值同PLSA的定义。对于性质不好的矩阵X或过大的矩阵会失效，返回全零或NaN。

rndprj.m

function y = rndprj(x, k)

将矩阵x进行随机投影，使其列数减少到k列。

skmeans.m

function [l c] = skmeans(x, k)

运行kmeans，输入输出定义同kmeans函数，但保证运行过程中不会因k太大而出错意外终止。相当于运行kmeans(x, k, 'emptyaction' ,'drop')。

sqresidue.m

function r = sqresidue(A, B)

计算A、B两矩阵各元素差的平方和。

EmotionCalc

C#源码和可执行程序，版本：Visual Studio 2010。

根据情感词典计算每日情感分值，并与股价进行DTW和皮尔森系数计算的程序。其中DTW在时间上错位一天的罚分为0.1，这个代价在配置文件中设置。

## 调用Mallet进行LDA的批处理文件

以下命令调用MALLET的LDA，对当前工作目录下所有文件运行LDA，并输出主题和词的分布到result\_keys.txt，输出每个文档的主题构成到result\_composition.txt。

cmd /c "%MALLET\_HOME%\bin\mallet import-dir --input .\ --output result.mallet --keep-sequence --remove-stopwords"

cmd /c "%MALLET\_HOME%\bin\mallet train-topics --input result.mallet --num-topics 100 --output-state result\_topicstate.gz --output-topic-keys result\_keys.txt --output-doc-topics result\_composition.txt --output-model result\_model.mallm"