**Bradley-Terry model 사용법**

* [소개슬라이드](http://wikin.nhncorp.com/download/attachments/648481/bradleyterrymodelguide.ppt?version=1&modificationDate=1182132687000)
* 기존 logistic regression model directory에 함께 넣어놓았습니다.
* <http://han749.nhncorp.com:8080/rank_source/trunk/logregm>
* R language에서의 Bradley-Terry model  
  <http://www2.warwick.ac.uk/fac/sci/statistics/staff/academic/firth/software/bradleyterry/bradleyterry-jss.pdf>
* Microsoft에서 사용하고 있는 RankNet은 Bradley-Terry model의 neural net version임.

**Bradley-Terry model 소개**

* **wine (술) 50병이 있는데, 그 wine들을 품질별로 sorting하고 싶을 때, 평가자가 50병을 동시에 음미해서 절대값을 주는 것은 무리가 따른다. 하지만 두 개씩 음미하고 A가 B보다 더 좋은 품질이라고 말하는 것은 쉽다. 만약, 50병중 2병씩 pair로 마셔보고 더 좋은 품질의 wine을 이야기한후, 그 자료로부터 50병의 품질을 sorting할 수 있을까 --> 이러한 문제를 Bradley-Terry model로 해결한다.**
* 검색 랭킹 문제의 경우, 검색된 결과들을 relevancy 순으로 sorting하는 문제이지, 사람들이 tagging해 놓은 점수를 최적화하는 문제가 아니다. 그런 의미에서 Bradley-Terry가 더 적합한 모델일 수 있다.
* 단, 대부분의 test collection 구축시 해당 relevancy 점수를 tagging하지, 두개씩 추출해서 선호도 검사를 하지 않는다. 즉, 엄밀하게 말하면 자료 구축시에도 사람이 두 개씩 비교한 후 선호도를 적도록 하는게 맞다.
* 기존 logistic regression은 입력 feature를 가지고 해당 target 값을 추정하고, 추정의 optimization은 실제값과 추정치간의 차이를 최소화하는 것을 목표로 하고 있다.
* 한편, Bradley-Terry model은 해당 target들값의 대소관계 (관측된 대소 관계)를 optimization하므로, 추정치는 실제값과 많이 다를 수 있지만 추정된 값들간의 대소 관계의 정확률은 기존 logistic regression보다 더 정확할 수 있다.

**BT model package 특성**

* Bradley-Terry model driver는 btlogregdrvr이다.
* make를 수행하면 btlogregdrvr가 만들어짐 (기존 logistic regression driver인 logregdrvr도 함께 만들어짐).
* 해당 directory에는 example 파일인 btexamp.tr과, logistic regression 비교를 위한 btexamp\_logreg.tr이 함께 있다.
* Bradley-Terry model은 입력 feature가 모두 실변수 (real-valued value)로 가정된다.
* 만약 discrete feature를 사용하고 싶다면 본인이 binary coding을 직접해야 한다.
  + '이미지 유무'가 feature라면 '0 1'과 '1 0'과 같이 두 개의 feature로 표현해야함.
* 첫번째 column은 해당 list set의 index이다. 즉 feature가 아님.

[?](http://wikin.nhncorp.com/pages/viewpage.action?pageId=648481)

|  |
| --- |
| [neo@dstest5 logregm]$ cat btexamp.tr  1 3 2 0.1  1 7 1 0.2  1 10 2 0.3  1 11 3 0.4  1 9 5 0.5  2 1 2 0.2  2 2 1 0.2  2 5 2 0.3  2 8 2 0.3  2 4 9 0.3 |

* 위자료에서 첫번째 column은 해당 list 번호임.
* 즉, query가 두개이었고, feature는 두번째, 세번째이 있으며, 마지막 column은 해당 문서의 만족도를 평가한 점수라고 가정하자.
* 기존 logistic regression을 가지고 학습을 할 때는 첫번째 column (query index)를 무시하고 다음과 같은 자료를 만든 후 학습시킨다.

[?](http://wikin.nhncorp.com/pages/viewpage.action?pageId=648481)

|  |
| --- |
| [neo@dstest5 logregm]$ cat btexamp\_logreg.tr  3 2 0.1  7 1 0.2  10 2 0.3  11 3 0.4  9 5 0.5  1 2 0.2  2 1 0.2  5 2 0.3  8 2 0.3  4 9 0.3  [neo@dstest5 logregm]$ ./logregdrvr -train -input btexamp\_logreg.tr  # of records : 10  # of features : 3  Program runs on TRAIN mode.  [neo@dstest5 logregm]$ cat beta.lrg  -1.974713  0.116720  0.097933  [neo@dstest5 logregm]$ cat outfile  0.100000 0.193303  0.200000 0.257359  0.300000 0.351683  0.400000 0.402034 <---- 0.4, 0.5에 대한 추정이 0.40, 0.39임을 확인!  0.500000 0.393035 <---- logistic regression은 추정값의 차이가 중요함.  0.200000 0.159477  0.200000 0.162011  0.300000 0.232322  0.300000 0.300464  0.300000 0.348319 |

* 기존 logistic regression에서 확률을 구하는 방법은 다음과 같았다.
  + 맨위의 것은 intercept이고 나머지부터는 feature의 weight임.
  + 수식은 파라미터의 음수의 weighted sum이다.
  + P(target) = 1/(1+exp(sum of negative weighted features))
  + P(target) = 1/(1+exp(1.974713-0.116720\*x1-0.097933\*x2))

**Bradley-Terry model 학습하기**

* btlogregdrvr의 활용법은 다음과 같다.
* '-train'이 있으면 '-input' 파일이 학습자료가 되고 학습된 파라미터는 '-mdl' 파일에 write된다.
* 학습된 파라미터로 예측된 결과와 실제값들은 자동적으로 '-output' 파일에 write된다.
* '-noexp'는 weighted sum 결과를 해당 feature를 가지고 있는 문서의 strength로 가정한다.
  + Bradley-Terry 모형은 exp(weighted sum of features)가 해당 문서의 stength이다.
  + 이 때, log 함수는 monotonic 함수 (ex. if x>y then f( x ) > f( y ))이므로, log를 취한 값을 이용할 수도 있다.
* '-sort'는 첫번째 column으로 sorting을 내부 프로그램에서 수행해 준다.
  + 외부에서 sorting해도 되므로 이 option을 사용할 가능성은 낮다. 이 option을 사용하지 않으면 반드시 sorting을 한 후 모델을 학습시켜야 한다.

[?](http://wikin.nhncorp.com/pages/viewpage.action?pageId=648481)

|  |
| --- |
| [neo@dstest5 logregm]$ cat btexamp.tr  1 3 2 0.1  1 7 1 0.2  1 10 2 0.3  1 11 3 0.4  1 9 5 0.5  2 1 2 0.2  2 2 1 0.2  2 5 2 0.3  2 8 2 0.3  2 4 9 0.3  [neo@dstest5 logregm]$ ./btlogregdrvr -h  syntax: ./btlogregdrvr {options}  -h, -help this message  -train if set, train a bradley-terry logistic reg, if not test it  -mdl beta.btm output bradley-terry LogReg file name (human-readable)  -input datafname input data file name  -output outfile output file name, when testing on input file  -how How-To page  -noexp output weighted sum only  -sort do sort the data by the first column  [neo@dstest5 logregm]$ ./btlogregdrvr -train -input btexamp.tr -noexp  # of records : 10  # of features : 4  Program runs on TRAIN mode.  [neo@dstest5 logregm]$ cat outfile  1 0.100000 8.601262  1 0.200000 11.064760  1 0.300000 17.210154  1 0.400000 20.895864  1 0.500000 23.347919 <---- 첫번째 query에 대한 목적값의 순서를 만족한다.  2 0.200000 6.141579  2 0.200000 4.915552  2 0.300000 11.060946  2 0.300000 14.750471  2 0.300000 27.022186 <---- 두번째 query에 대한 목적값의 순서도 만족한다.  [neo@dstest5 logregm]$ cat beta.btm  0.000000  1.229842  2.455869 |

* 위에서 beta.btm 파일에 있는 값들은 Bradley-Terry 모형의 파리미터이다.
  + 맨위의 것은 intercept이고 나머지부터는 feature의 weight임.
  + intercept는 반드시 0이 되어야 한다. 0이 아니라면 프로그램을 잘못 수행시킨 것이다.
  + BT 모형은 확률을 구하는 것이 아니라 순서를 구하는 것이므로, 모든 값에 intercept를 넣을 필요가 당연히 없다.
  + 수식은 파라미터의 exp(weighted sum)이다.
  + Strength(document) = exp(1.229842 \* x1 + 2.455869 \* x2)
  + log (Strength(document)) = 1.229842 \* x1 + 2.455869 \* x2

**Bradley-Terry model test하기**

* '-train'이 없으면, testing mode가 된다.
* testing 결과는 '-output' 파일에 남는다.
* query 순서에 해당하는 첫번째 column이 함께 출력되게 되어있음.

[?](http://wikin.nhncorp.com/pages/viewpage.action?pageId=648481)

|  |
| --- |
| [neo@dstest5 logregm]$ ./btlogregdrvr -input btexamp.tr -noexp  # of records : 10  # of features : 4  Program runs on TEST mode.  [neo@dstest5 logregm]$ cat outfile  1 0.100000 8.601264  1 0.200000 11.064763  1 0.300000 17.210158  1 0.400000 20.895869  1 0.500000 23.347923  2 0.200000 6.141580  2 0.200000 4.915553  2 0.300000 11.060948  2 0.300000 14.750474  2 0.300000 27.022189 |

* 첫번째 column : query 순서, 혹은 list의 순서
* 두번째 column : 입력으로 들어오는 최종 목적값
* 세번째 column : 해당 feature들을 이용한 그 문서의 strength
  + BT 모형을 평가할 때는 최종 목적값과 예측값과의 차이를 평가하면 안되고,  
    해당 list내에서의 목적값의 순위와 얼마나 비슷한지를 평가해야한다.
  + 현재 순서에 대한 평가 프로그램은 구현되어 있지 않다.

**Bradley-Terry model 변수의 신뢰도 검증**

[?](http://wikin.nhncorp.com/pages/viewpage.action?pageId=648481)

|  |
| --- |
| [neo@dstest5 logregm]$ ./btlogregdrvr -train -input btquality.hot.dat -noexp  # of records : 7117  # of features : 8  Program runs on TRAIN mode.  ----------------------------------------------  Coeff. Std.Err Z score  ----------------------------------------------  1 var. 0.487786 0.043982 11.090651  2 var. 6.081734 0.030306 200.677936  3 var. 0.459605 0.011912 38.582182  4 var. -0.184317 0.008624 -21.372577  5 var. 0.190294 0.018287 10.405768  6 var. 1.195213 0.011385 104.982464  ---------------------------------------------- |

* 위와 같이 학습한후 각 변수의 coeff.에 대해 Z score가 나오게 되는데, Z score가 -1.96과 +1.96사이에 오면 그 변수가 0이어도 된다는 것에 대해 95% 신뢰할 수 있다는 뜻이다. 그러므로, Z score의 절대값이 클수록 좋으며, 절대값이 작을 때 그 변수를 제거해도 된다는 뜻이 된다.