## 原始特征

在原始特征经过卡方10000筛选以后，

-s6 的准确率如下：

total precise: 0.926

avg\_precise: 0.904

30 0.819

33 0.822

31 0.830

26 0.836

25 0.847

7 0.860

17 0.860

14 0.865

11 0.865

16 0.869

9 0.883

27 0.888

19 0.892

15 0.894

28 0.897

3 0.904

29 0.909

6 0.913

23 0.915

2 0.917

20 0.920

13 0.925

4 0.929

1 0.931

21 0.936

18 0.942

5 0.943

10 0.966

24 0.968

12 0.969

22 0.971

32 0.972

8 0.986

预测的不同情况下的类别：

INFO job done has\_label:1147133, fill\_label:9625028, default\_label:15423

## 对tf归一化

### s6 B1

每一个特征之前是直接取其出现次数来做，现在使用tf/n，并将其归一化成单位向量以后，进行训了和测试，结果如下：

total precise: 0.926

avg\_precise: 0.903(降低0.1%个百分点)

33 0.789

30 0.797

7 0.824

31 0.832

26 0.844

25 0.846

11 0.851

29 0.861

9 0.865

17 0.867

14 0.869

27 0.877

16 0.894

23 0.894

19 0.897

15 0.897

2 0.906

3 0.906

6 0.915

28 0.918

13 0.920

20 0.939

1 0.940

5 0.944

4 0.949

18 0.952

10 0.960

24 0.969

22 0.969

12 0.972

21 0.972

32 0.976

8 0.987

可以看出来，归一化后，对小类别分对的概率比之前偏低，而对大类别分类的正确率比之前高，简单总结是好的越好，差的越差，但因为是随机切分数据进行训练的，这些微弱的差异应该不算什么。

目前一次模型训了使用s6，大概在20分钟左右。将eps设置成大于0.01的值应该可以加快训练速度

### s7 B1

训练模型需要8分钟, 训练出来的模型文件有200+M，太大了。

total precise: 0.919

avg\_precise: 0.891

7 0.758

33 0.780

30 0.790

31 0.813

26 0.826

25 0.827

17 0.831

11 0.835

29 0.836

9 0.845

14 0.861

27 0.865

15 0.874

19 0.876

16 0.886

23 0.892

3 0.897

2 0.899

6 0.906

13 0.908

28 0.918

20 0.934

1 0.937

5 0.942

4 0.946

10 0.949

18 0.957

22 0.967

24 0.968

21 0.968

12 0.968

32 0.973

8 0.986

### s1 B1

花费了4分钟，模型文件50M结果如下：

total precise: 0.932

avg\_precise: 0.914

33 0.815

30 0.816

31 0.856

7 0.857

25 0.864

11 0.866

9 0.875

14 0.876

29 0.877

26 0.879

17 0.887

27 0.894

2 0.911

19 0.913

3 0.913

16 0.917

15 0.917

28 0.918

6 0.919

13 0.926

23 0.927

1 0.941

20 0.945

4 0.948

5 0.949

18 0.953

21 0.964

10 0.969

24 0.970

22 0.971

12 0.973

32 0.977

8 0.988

### s0 B1

花费了将近半个小时，模型文件138M，太大了

total precise: 0.918

avg\_precise: 0.890

7 0.746

33 0.780

30 0.787

31 0.813

25 0.827

17 0.828

26 0.828

29 0.837

11 0.837

9 0.843

14 0.859

27 0.862

15 0.871

19 0.875

16 0.884

3 0.894

23 0.894

2 0.899

6 0.907

13 0.907

28 0.917

20 0.934

1 0.937

5 0.941

4 0.943

10 0.951

18 0.957

12 0.966

22 0.966

24 0.968

21 0.968

32 0.972

8 0.981

### s4 B1

训练时间1m14s，模型文件43M

total precise: 0.934

avg\_precise: 0.917

33 0.822

30 0.825

31 0.857

7 0.862

11 0.868

25 0.869

9 0.881

29 0.888

17 0.889

26 0.889

27 0.894

14 0.896

2 0.912

3 0.914

16 0.914

19 0.915

15 0.917

23 0.923

6 0.924

28 0.925

13 0.928

1 0.942

20 0.947

18 0.949

5 0.949

4 0.954

24 0.970

21 0.971

10 0.972

22 0.973

12 0.974

32 0.976

8 0.988

## 评测方式

使用label.svm数据进行模型训练完了以后，获取所有数据的预测类标

predict merge\_keyword\_class\_title.tf.no\_label.svm all.model.s4 predict.no\_label.data

将预测的类标与query进行合并：

./merge\_test\_and\_true\_label.py ../data/query.no\_label ../data/predict.no\_la

bel.data > ../data/query.predict\_label

将获得的预测(query, label)对填充到输出文件中

./fill\_label\_with\_predict\_value.py ../data/query.predict\_label < ../data/keyword\_class\_inner\_label.txt > ../data/keyword\_class.predict.txt

## 其他事项

用grep –c –P “^[\s0]+$” 来统计所有特征是0的个数，发现与grep -P “^[\s0]+$” | wc –l 的结果不一样，要少很多。

## 评测结果

在使用普通的卡方检验选出特征以后，-s 6准确率76.56%，-s 4 准确率77.65%