编程实验2

分类+降维

CLASSIFICATION

Classification Dataset

SpamAssassin语料库

一个垃圾邮件语料库。其中的标签分别为

spam：垃圾邮件;

easy ham：易识别的正常邮件

Dataset Representation

a) 原始邮件预处理：去除头部，保留邮件正文；

b) 构建特征词类别知识库，通过从邮件正文中抽取特征词来构建邮件分类器的特征集。

c) 语料库构造：量化特征词频率——构造一个词项-文档矩阵(TDM) N*M的矩阵，[i, j]表示词项i在文档j中出现的次数

Experiments

Algorithms	Naïve Bayes	Least Squares	SVM
SpamAssassin (2 classes)	√	√	√

1. Build a Naïve Bayes classifier

- Write a Python function "`nBayesClassifier(traindata, trainlabel, testdata, testlabel, threshold)`" that takes 5 arguments, `traindata`, `trainlabel`, `testdata`, `testlabel`, `threshold` as input, and returns a vector `ypred` as the predictions of the test data, as well as the performance measures including SP, SR, F.

  if P(spam|email)>threshold, then *spam*

  ypred与SP, SR, F以tuple形式返回

2. Build a least squares classifier

- Write a Python function "`lsClassifier(traindata, trainlabel, testdata, testlabel, lambda)`" that takes 5 arguments, `traindata`, `trainlabel`, `testdata`, `testlabel`, `lambda` as input, and returns a vector `ypred` as the predictions of the test data, as well as the performance measures including SP, SR, F.

$$\min_{\mathbf{w}} (X\mathbf{w} - \mathbf{y})^2 + \lambda \|\mathbf{w}\|^2$$

3. Build a support vector machine

- Write a Python function "`softsvm(traindata, trainlabel, testdata, testlabel, sigma, C)`" that takes 6 arguments, `traindata, trainlabel, testdata, testlabel, sigma, C` as input, and returns a vector `ypred` as the predictions of the test data, as well as the performance measures including SP, SR, F.

  when sigma=0, use linear kernel $K(\mathbf{x}_i,\mathbf{x}_j)= \mathbf{x}_i^{\mathbf{T}}\mathbf{x}_j$,

  otherwise use the RBF kernel $K(\mathbf{x}_i,\mathbf{x}_j)$ = $e^{-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{\sigma^2}}$

4. Cross Validation

- On each dataset:
  - Implement 5 fold cross validation to tune the parameters for each algorithm
  - For each algorithm:
    - Return a matrix: parameter (set) X F-measure on each fold
    - Select the parameter (set) with best F-measure

Cross-validation

- The improved holdout method: k-fold *cross-validation*
  - Partition data into k roughly equal parts;
  - Train on all but j-th part, test on j-th part

For Naïve Bayes, select threshold from…? (e.g.: threshold=[0.5 0.6 0.7 0.75 0.8 0.85 0.9])

For least squares, select lambda from…?

$$\min_{\mathbf{w}} (X\mathbf{w} - \mathbf{y})^2 + \lambda \|\mathbf{w}\|^2$$

For SVM, select (C, sigma) value combination from:

C=[1, 10, 100, 1000], sigma?

5. Testing

- SP（垃圾邮件识别的准确率）

$$\mathrm{SP} = \frac{n_{pam\to pam}}{n_{pam\to pam} + n_{normal\to pam}}$$

- SR（垃圾邮件识别的查全率）

$$\mathrm{SR} = \frac{n_{pam\to pam}}{N_{pam}}$$

$n_{pam\to pam}$是正确识别的垃圾邮件数；
$n_{normal\to pam}$是正常邮件被误识别为垃圾邮件数；
N_{pam}是垃圾邮件总数。

5. Testing

- 综合考虑了准确率和查全率两方面的指标F
  其数学公式如下：

$$\mathrm{F} = \frac{SP \times SR \times 2}{SP + SR}$$

总体来说，F值越高，模型的综合性能越优。

Notes on building an SVM

- Make sure you understand the math
- Use some simple synthetic data（模拟数据） to verify
- Use the same kernel during training and testing
- When calculating b, remember

  to use the same kernel!
- Check α_i to debug
  - Do they satisfy the constraints?

Calculate b in SVM

Dual optimization problem:

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i^\top \mathbf{x}_j) \quad \text{subject to} \quad 0 \leq \alpha_i \leq C, \forall i$$

$$\textstyle\sum_{i=1}^{n} \alpha_i y_i = 0$$

b can be recovered by

$$b = y_i - \sum_{j=1}^{n} \alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j) \quad \text{for any i that } \alpha_i \neq 0$$

$$b = y_i - \sum_{j=1}^{n} \alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j) \quad \text{for any i with maximal } \alpha_i$$

$$b = avg_{i:\alpha_i \neq 0} \left(y_i - \sum_{j=1}^{n} \alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j) \right)$$

DIMENSIONALITY REDUCTION

PCA for Face Recognition

Dataset:

40个人的人脸图片，每个人10张，共400张，相同的人脸图片在同一文件夹下，图像尺寸是92×112，图像背景为黑色。

PCA for Face Recognition

- 在训练过程中通过PCA算法来进行降维
- 将每个文件夹随机取出一张头像作为测试集，其余头像作为训练集将测试集中的每一幅降维图像与降维的训练集进行匹配，然后将其分类到距离最小的训练集头像中，如果两个头像表示一个人，表示识别成功，否则表示识别失败。并计算出识别率。

PCA for Face Recognition

- 实现并提交一个Python函数PCA(traindata, testdata, threshold)

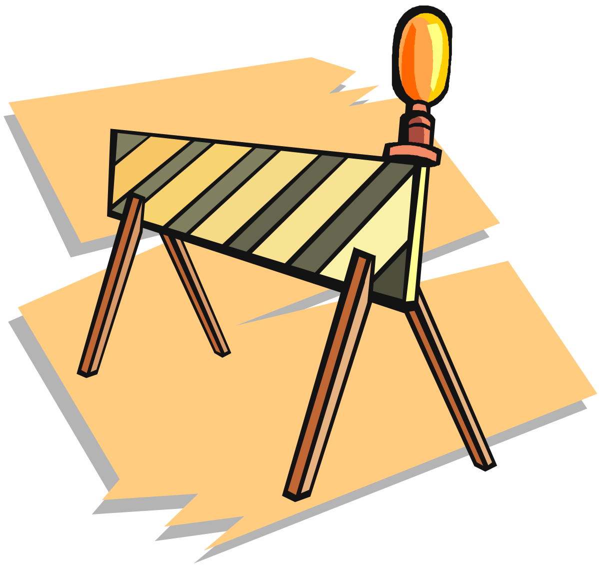
- 其中threshold表示特征值的累计贡献率。即选择前m个特征向量，使得

  Sum(first m-1 eigenvalues)/Sum(all eigenvalues)<threshold<=Sum(first m eigenvalues)/Sum(all eigenvalues)

PCA for Face Recognition

- 实验验证PCA算法效果
  - 检验随着threshold不同取值，PCA选择的降维维度和识别准确率会有什么样的变化。
  - 仔细观察，并尝试提出算法的改进方案。
  - 在实验报告中总结以上的实验结果。
  - 提交将训练集降维后的图像矩阵。

Caution



Academic Integrity