戴尔企业特惠月
全场满额送1TB硬盘

限时特供：8月8日-8月13日

SVM实现邮件分类

标签： 邮件 svm kernel

2016-12-16 17:20 785人阅读 评论(0)

分类：

机器学习（15）

首先学习一下svm分类的使用。

主要有以下步骤：

- Loading and Visualizing Dataj
- Training Linear SVM
- Implementing Gaussian Kernel
- Training SVM with RBF Kernel
- 选择最优的C, sigma参数
- 画出边界线

线性keneral实现

```
C = 1;
model = svmTrain(X, y, C, @linearKernel, 1e-3, 20);
visualizeBoundaryLinear(X, y, model);
```

访问： 337046次
积分： 5374
等级： BLOG 6
排名： 第4997名

原创： 191篇 转载： 43篇
译文： 0篇 评论： 47条

文章搜索

文章分类

编辑 (6)
C语言 (3)
C++ (4)
数据结构 (17)
linux (5)
cocos2dx (11)
网络开发 (8)
android (74)
python (9)
java (22)
图像处理 (23)
Unity3D (1)
数据库 (3)
javascript (2)
自定义控件 (18)
Material (1)
React (6)
php (12)
ionic (2)
ios (2)
机器学习 (16)

数据结构之图的实现 (4697)

评论排行

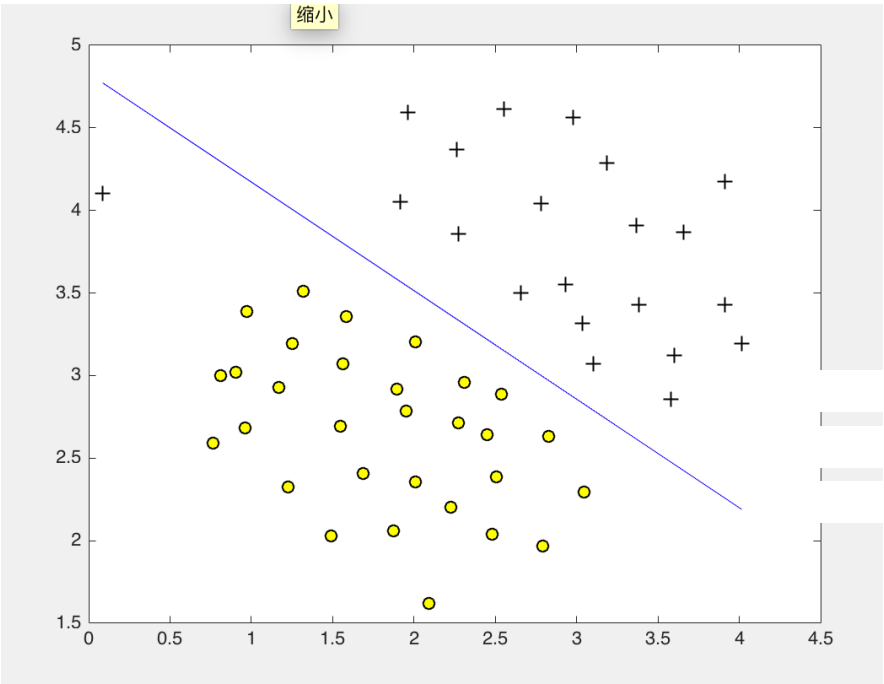
Android实现电子邮箱客户端 (16)
图像特征提取 (6)
基于PCA的人脸特征抽取 (6)
数字图像处理入门 (4)
图像识别初步 (3)
PHP之MVC项目实战 (2)
Ionic2 Tutorial (1)
struts2服务端与android交互 (1)
Android高性能ORM数据库D... (1)
坦克大战实现 (1)

推荐文章

* CSDN日报20170817——《如果不从事编程，我可以做什么？》
* Android自定义EditText：你需要一款简单实用的SuperEditText(一键删除&自定义样式)
* 从JDK源码角度看Integer
* 微信小程序——智能小秘“遥知之”源码分享（语义理解基于olami）
* 多线程中断机制
* 做自由职业者是怎样的体验

最新评论

坦克大战实现
qq_39848231 ：请问有源码吗？
Ionic2 Tutorial
淘到星星的小笛子 ：楼主大神，全英博客啊
机器学习之异常检测
Mr_Alvin ：你好，想请教一个问题，【异常检测（anomaly detection）】这个章节中，最终用单元高斯...
Android高性能ORM数据库DBFlow入门
Mr_FuSS ：尝试过存储图片么？
图像特征提取
江军祥 ：@wcolorfulrainbow:数学图像处理与机器视觉_Matlab与VC++实现
图像特征提取
江军祥 ：@liweichong_3:数学图像处理与机器视觉_Matlab与VC++实现



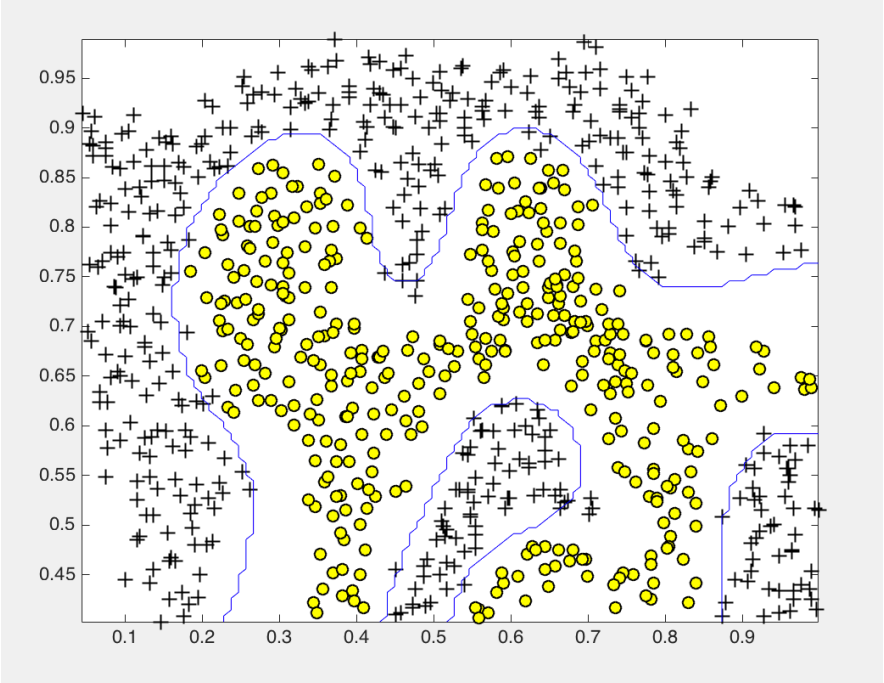
高斯keneral实现

```
function sim = gaussianKernel(x1, x2, sigma)
x1 = x1(:); x2 = x2(:);
sim = 0;
sim = exp( - (x1-x2)'* (x1-x2) / (2 * sigma *sigma ) );
end


load('ex6data2.mat');

% SVM Parameters
C = 1; sigma = 0.1;

% We set the tolerance and max_passes lower here so that the code will run
% faster. However, in practice, you will want to run the training to
% convergence.
model= svmTrain(X, y, C, @(x1, x2) gaussianKernel(x1, x2, sigma));
visualizeBoundary(X, y, model);
```



选择合适的参数

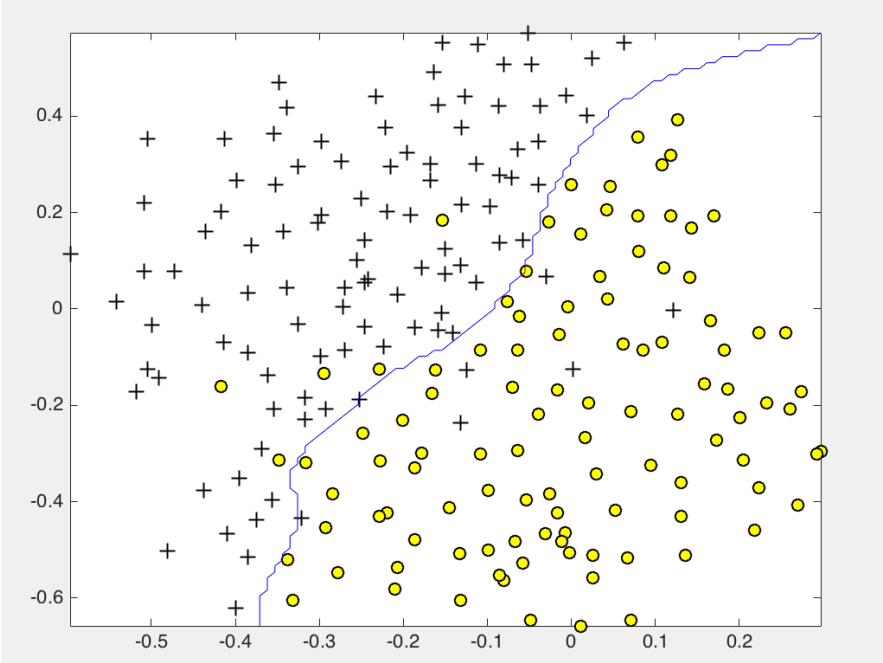
```
function [C, sigma] = dataset3Params(X, y, Xval, yval)
C = 1;
sigma = 0.3;
C_vec = [0.01 0.03 0.1 0.3 1 3 10 30]';
sigma_vec = [0.01 0.03 0.1 0.3 1 3 10 30]';
error_val = zeros(length(C_vec),length(sigma_vec));
error_train = zeros(length(C_vec),length(sigma_vec));
for i = 1:length(C_vec)
    for j = 1:length(sigma_vec)
      model= svmTrain(X, y, C_vec(i), @(x1, x2) gaussianKernel(x1, x2, sigma_vec(j)));
      predictions = svmPredict(model, Xval);
      error_val(i,j) = mean(double(predictions ~= yval));
    end
end
% figure
% error_val
% surf(C_vec,sigma_vec,error_val)   % 画出三维图找最低点

[minval,ind] = min(error_val(:));   % 0.03
[I,J] = ind2sub([size(error_val,1) size(error_val,2)],ind);
C = C_vec(I)          %   1
sigma = sigma_vec(J)  %   0.100

% [I,J]=find(error_val ==  min(error_val(:)) );    % 另一种方式找最小元素位子
% C = C_vec(I)          % 1
% sigma = sigma_vec(J)  % 0.100
end

[C, sigma] = dataset3Params(X, y, Xval, yval);

% Train the SVM
model= svmTrain(X, y, C, @(x1, x2) gaussianKernel(x1, x2, sigma));
visualizeBoundary(X, y, model);
```



邮件分类

主要步骤如下：

- 邮件数据归一化处理
- 特征提取
- Train Linear SVM for Spam Classification
- Test Spam Classification

- Top Predictors of Spam
- 测试自己的email

归一化处理

In processEmail.m, we have implemented the following email prepro- cessing and normalization steps:

- Lower-casing: The entire email is converted into lower case, so that captialization is ignored (e.g., IndIcaTE is treated the same as Indicate).
- Stripping HTML: All HTML tags are removed from the emails. Many em come with HTML formatting; we remove all the HTML tags, so that onl remains.
- Normalizing URLs: All URLs are replaced with the text "httpaddr".
- Normalizing Email Addresses: All email addresses are replaced with the text "emailaddr".
- Normalizing Numbers: All numbers are replaced with the text "number".
- Normalizing Dollars: All dollar signs ($) are replaced with the text "dollar".
- Word Stemming: Words are reduced to their stemmed form. For ex- ample, "discount", "discounts", "discounted" and "discounting" are all replaced with "discount". Sometimes, the Stemmer actually strips off additional characters from the end, so "include", "includes", "included", and "including" are all replaced with "includ".
- Removal of non-words: Non-words and punctuation have been re- moved. All white spaces (tabs, newlines, spaces) have all been trimmed to a single space character.

处理之后效果如下:

```
anyon know how much it cost to host a web portal well it depend on how
mani visitor your expect thi can be anywher from less than number buck
a month to a coupl of dollarnumb you should checkout httpaddr or perhap
amazon ecnumb if your run someth big to unsubscrib yourself from thi
mail list send an email to emailaddr
```

Figure 9: Preprocessed Sample Email

```
1 aa
2 ab
3 abil
...
86 anyon
...
916 know
...
1898 zero
1899 zip
```

Figure 10: Vocabulary List

```
86 916 794 1077 883
370 1699 790 1822
1831 883 431 1171
794 1002 1893 1364
592 1676 238 162 89
688 945 1663 1120
1062 1699 375 1162
479 1893 1510 799
1182 1237 810 1895
1440 1547 181 1699
1758 1896 688 1676
992 961 1477 71 530
1699 531
```

Figure 11: Word Indices for Sample Email

Vocabulary List

我们取垃圾邮件中最常见的单词放入单词表中。

Our vocabulary list was selected by choosing all words which occur at least a 100 times in the spam corpus, resulting in a list of 1899 words. In practice, a vocabulary list with about 10,000 to 50,000 words is often used.

将我们邮件中有的单词在单词表中的id存储在word_indices中

```
    for i=1:length(vocabList)
        if( strcmp(vocabList{i}, str) )
          word_indices = [word_indices;i];
        end
    end
```

Extracting Features from Emails

然后查找我们的邮件中的单词在单词表中的位置，有则置1,无则跳过。

You should look up the word in the vocabulary list vocabList and find if the word exists in the vocabulary list. If the word exists, you should add the index of the word into the word indices variable. If the word does not exist, and is therefore not in the vocabulary, you can skip the word.

```
function x = emailFeatures(word_indices)
% Total number of words in the dictionary
n = 1899;

% You need to return the following variables correctly.
x = zeros(n, 1);
x(word_indices) = 1;
end
```

Training SVM for Spam Classification

```
load('spamTrain.mat');

fprintf('\nTraining Linear SVM (Spam Classification)\n')
fprintf('(this may take 1 to 2 minutes) ...\n')

C = 0.1;
model = svmTrain(X, y, C, @linearKernel);

p = svmPredict(model, X);

fprintf('Training Accuracy: %f\n', mean(double(p == y)) * 100);

%% ================== Part 4: Test Spam Classification ================
load('spamTest.mat');

fprintf('\nEvaluating the trained Linear SVM on a test set ...\n')

p = svmPredict(model, Xtest);

fprintf('Test Accuracy: %f\n', mean(double(p == ytest)) * 100);
```

After loading the dataset, ex6 spam.m will proceed to train a SVM to classify between spam (y = 1) and non-spam (y = 0) emails. Once the training completes, you should see that the classifier gets a training accuracy of about 99.8% and a test accuracy of about 98.5%.

Top Predictors for Spam

找出最易被判断为垃圾邮件的单词。

```
[weight, idx] = sort(model.w, 'descend');
vocabList = getVocabList();

fprintf('\nTop predictors of spam: \n');
for i = 1:15
    fprintf(' %-15s (%f) \n', vocabList{idx(i)}, weight(i));
end
```

```
Top predictors of spam:
 our             (0.499603)
 click           (0.467479)
 remov           (0.423328)
 guarante        (0.384096)
 visit           (0.370542)
 basenumb        (0.341289)
 dollar          (0.327056)
 will            (0.271494)
 pleas           (0.265977)
 price           (0.261521)
 nbsp            (0.259023)
 most            (0.257291)
 lo              (0.251212)
 ga              (0.239615)
 al              (0.239520)
```

Try your own emails

```
filename = 'spamSample1.txt';

% Read and predict
file_contents = readFile(filename);
word_indices  = processEmail(file_contents);
x             = emailFeatures(word_indices);
p = svmPredict(model, x);

fprintf('\nProcessed %s\n\nSpam Classification: %d\n', filename, p);
fprintf('(1 indicates spam, 0 indicates not spam)\n\n');
```

可以看出我们的邮件判断准确率大概在98%左右。

- 上一篇 机器学习之异常检测
- 下一篇 机器学习之推荐系统

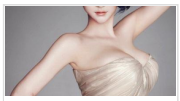
戴尔企业特惠月
全场满额送1TB硬盘
限时特供：8月8日-8月13日

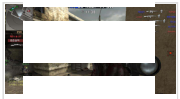
-
- 【直播】机器学习之凸优化--马博士
- svm分类器的实现（matlab）
- 【直播】计算机视觉原理及实战--屈教授
- SVM实战之垃圾邮件过滤
- 机器学习&数据挖掘7周实训--韦玮
- 这里实现了四种SVM工具箱的分类与回归算法
- 机器学习之数学基础系列--AI100
- 垃圾邮件二分类 NaiveBayes v.s SVM (matlab)
-
- 多核svm分类器
-
-
- 具有操作界面的SVM分类实现系统
- SVM实现垃圾邮件分类（java调用libsvm.jar）
- 用MatLab实现SVM分类.


胸模丰胸了吗


人工智能机器人


短信验证码接口


胸大且下垂



查看评论

暂无评论

发表评论

用户名： tcftrees

评论内容：

提交

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场