编程练习 6：

支持向量机

机器学习

# 介绍

在本练习中，您将使用支持向量机 （SVM） 构建垃圾邮件分类器。在开始编程练习之前，我们强烈建议您观看视频讲座并完成相关主题的评论问题。

要 开始练习，您需要下载入门代码并将其内容解压缩到要完成练习的目录中。如果需要，请在 启动此 exerc ise 之前，使用 Octave/MATLAB 中的 cd 命令更改为此目录。

您还可以在课程网站的“环境设置说明”中找到有关安装Octave / MATLAB的说明。

## 本练习中包含的文件

ex6.m - 练习前半部分的 Octave/MATLAB 脚本 ex6data1.mat - 示例数据集 1 ex6data2.mat - 示例数据集 2 ex6data3.mat - 示例数据集 3

svmTrain.m - SVM rraining 函数 svmPredict.m - SVM 预测函数 plotData.m - Plot 2D Data visualizeBoundaryLinear.m - Plot linear boundary visualizeBoundary.m - Plot non-linear boundary linearKernel.m - SVM 的线性核

[*?*] gaussianKernel.m - SVM 的高斯核

[*?*] 数据集3参数.m - 用于数据集 3 的参数

ex6 spam.m - 用于练习后半部分的 Octave/MATLAB 脚本

spamTrain.mat - 垃圾邮件训练集 spamTest.mat - 垃圾邮件测试集 电子邮件样本 1.txt - 示例电子邮件 1 电子邮件样本 2.txt - 示例电子邮件 2 垃圾邮件样本 1 .txt - 示例垃圾邮件 1 垃圾邮件样本2.txt - 示例垃圾邮件 2 词汇.txt - 词汇表 getVocabList.m - 加载词汇表 porterStemmer.m - 词干函数 readFile.m - 将文件读入字符串 submit.m - 将您的解决方案发送到我们服务器的提交脚本

[*?*] processEmail.m - 电子邮件预处理

[*?*] emailFeatures.m - 从电子邮件中提取功能

*?* 表示您需要完成的文件

在整个练习过程中，您将在 ex6.m 之前使用脚本。 这些脚本为问题设置数据集，并调用您将编写的函数。您只需按照此赋值中的说明修改其他 fil es 中的函数。

## 从何处获取帮助

本课程中的练习使用Octave[[1]](#footnote-1) 或MATLAB，这是一种非常适合数值计算的高级编程语言。如果您没有安装Octave或MATLAB，请参考课程网站“环境设置说明”中的安装说明。

在 Octave/MATLAB 命令行中，键入 help 后跟函数名称将显示内置函数的文档。例如，帮助绘图将显示绘图的帮助信息。有关倍频程函数的更多文档，请参阅[倍频程文档页面。](http://www.gnu.org/software/octave/doc/interpreter/) MATLAB 文档可以在 [MATLAB 文档页面上](http://www.mathworks.com/help/matlab/?refresh=true)找到[。](http://www.mathworks.com/help/matlab/?refresh=true)

我们还强烈建议使用 在线 **讨论** 与其他学生讨论练习。但是，不要查看他人编写的任何源代码或与他人共享您的源代码。

# 支持向量机

在本练习的前半部分，您将对各种示例 2D 数据集使用支持向量马赫 ines （SVM）。试验这些数据集将帮助您直观地了解 SVM 的工作原理以及如何将高斯核与 SVM 配合使用。在下半部分的练习中，您将使用支持向量机来构建垃圾邮件分类器。

提供的脚本 ex6.m 将帮助您逐步完成练习的前半部分。

## 示例数据集 1

我们将从一个2D示例数据集开始，该数据集可以通过线性边界分隔。脚本 ex6.m 将绘制训练数据（图 1）。在此数据集中，正面示例（用+表示）和否定示例（用*o*表示）的位置表明由间隙指示的自然分离。但是，请注意，在最左边有一个异常值正示例 +，大约 （0*.*1*，*4*.*1）. 作为本练习的一部分，您还将看到此异常值如何影响 SVM 决策边界。

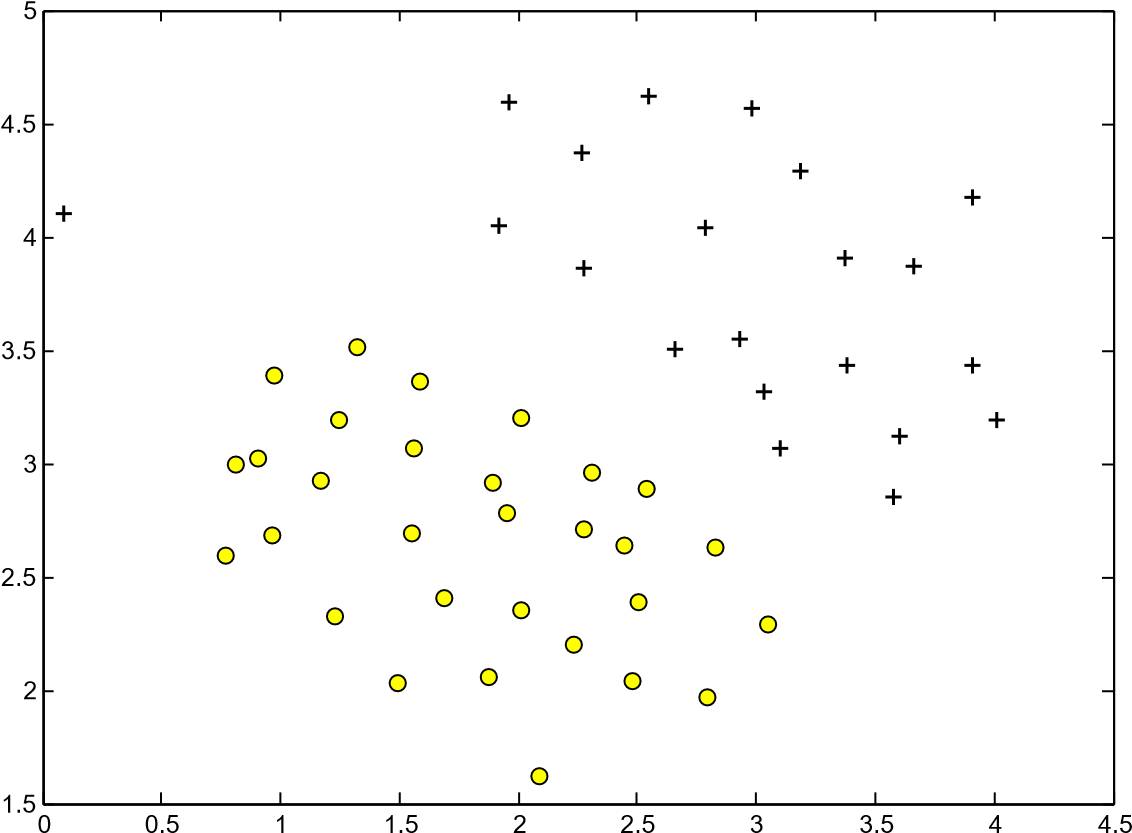


图 1：示例数据集 1

在本部分练习中，您将尝试对 SVM 使用不同的 *C* 参数值。非正式地说，*C* 参数是一个正值，用于控制错误分类的训练示例的惩罚。一个大的 *C* 参数 t会引导 SVM 尝试对所有示例进行正确的分类。 *C* 扮演着类似于 的角色，其中 *λ* 是我们之前用于逻辑回归的正则化参数。

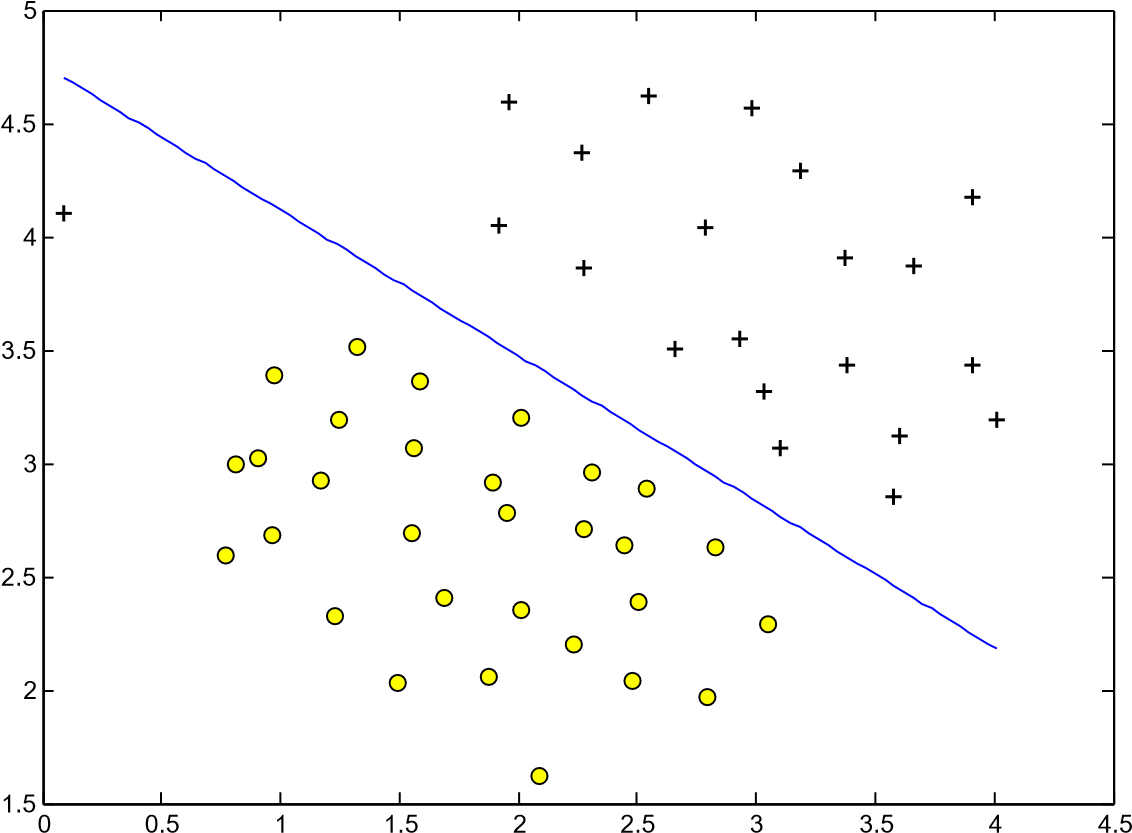


图 2：*C*  = 1 的 SVM 决策边界（示例数据集 1）

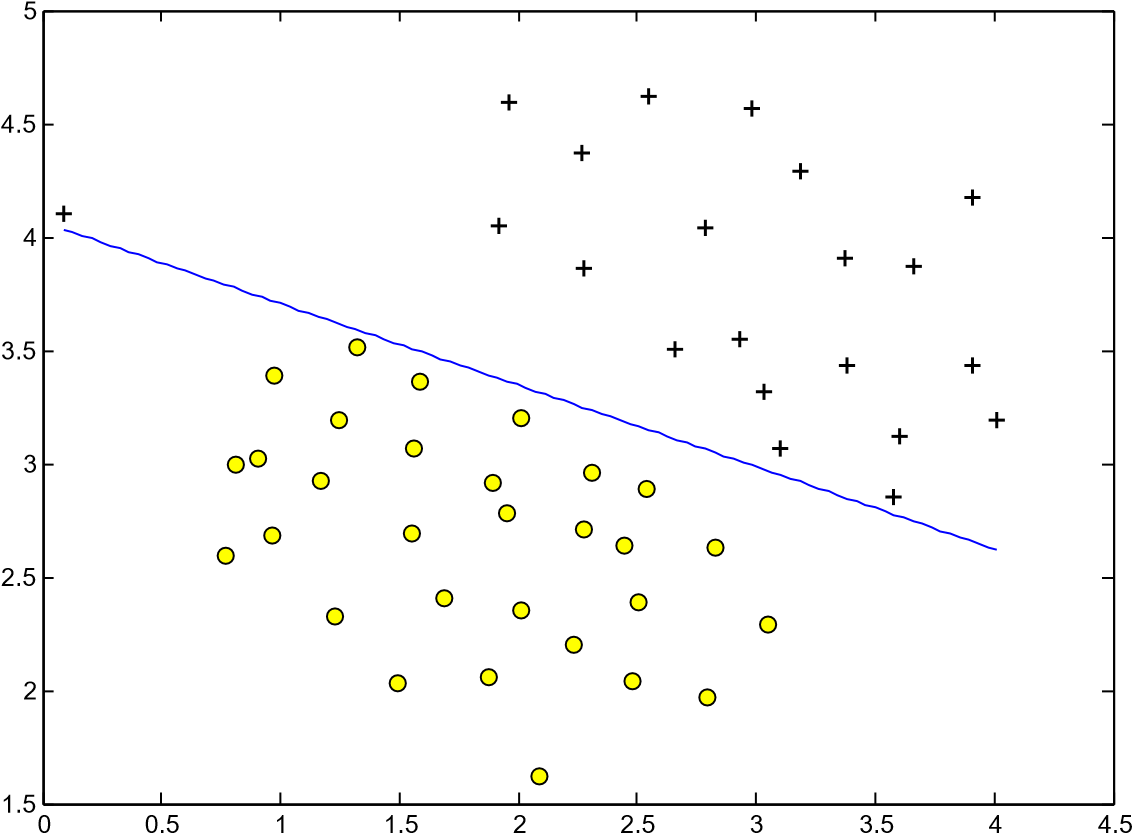


图 3：*C*  = 100时的 SVM 决策边界（示例数据集 1）

ex6.m 中的下一部分将使用 SVM 软件运行 SVM 训练（*C* = 1），该软件已包含在起始代码 svmTrain.m 中。[[2]](#footnote-2)当 *C* = 1 时，您应该会发现 SVM 将 decision 边界放在两个数据集之间的间隙中，并对最左侧的数据点*进行了错误分类*（图 2）。

**实现说明：** 大多数 SVM 软件包（包括 svmTrain.m）会自动为您添加额外功能 *x*0 = 1，并自动处理学习截距项 *θ*0。 因此 ，在将训练数据传递到 SVM 软件时，无需自己添加此额外功能 *x*0 = 1。特别是，在Octave/MATLAB中，你的代码应该使用 训练示例 *x* ∈R*n* （而不是 *x* ∈R*n*+1）;例如，在第一个示例数据集中， *x* ∈R2。

您的任务是在此数据集上尝试不同的 *C*  值。具体来说，您应该将脚本中 *C*  的值更改为 *C* = 100，然后再次运行 SVM 训练。当 *C* = 100 时，您应该会发现 SVM 现在正确地对每个示例进行分类，但其决策边界不会自然地适合数据（图 3）。

## 具有高斯核的 SVM

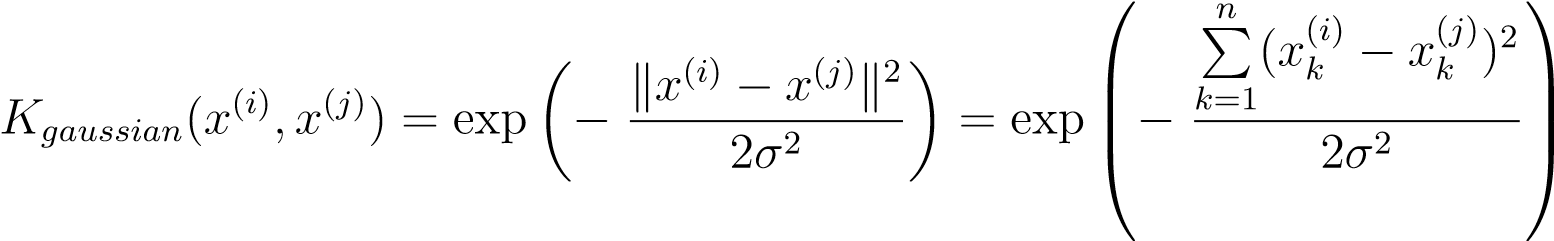
在本部分练习中，您将使用 SVM 执行非线性分类。特别是，您将在不可线性可分割的数据集上使用具有高斯核的SVM。

### 高斯核

要使用 SVM 找到非线性决策边界，我们需要首先实现一个高斯核。您可以将高斯核视为一个相似性函数，用于测量一对示例 （x（*i*），*x*（*j*）） 之间的“距离”。 高斯核也由带宽参数σ进行参数化，该参数决定了随着示例的进一步相距，相似性指标降低（为 0）的速度。

您现在应该在 gaussianKernel.m 中完成代码以进行计算

两个例子之间的高斯核，（*x*（*i*）*，x*（*j*））。高斯核函数定义为：

 *.*

完成函数 gaussianKernel.m 后，脚本 ex6.m 将在提供的两个示例上测试您的内核函数，您应该会看到值 0.324652。

*您现在应该提交您的解决方案。*

### 示例数据集 2

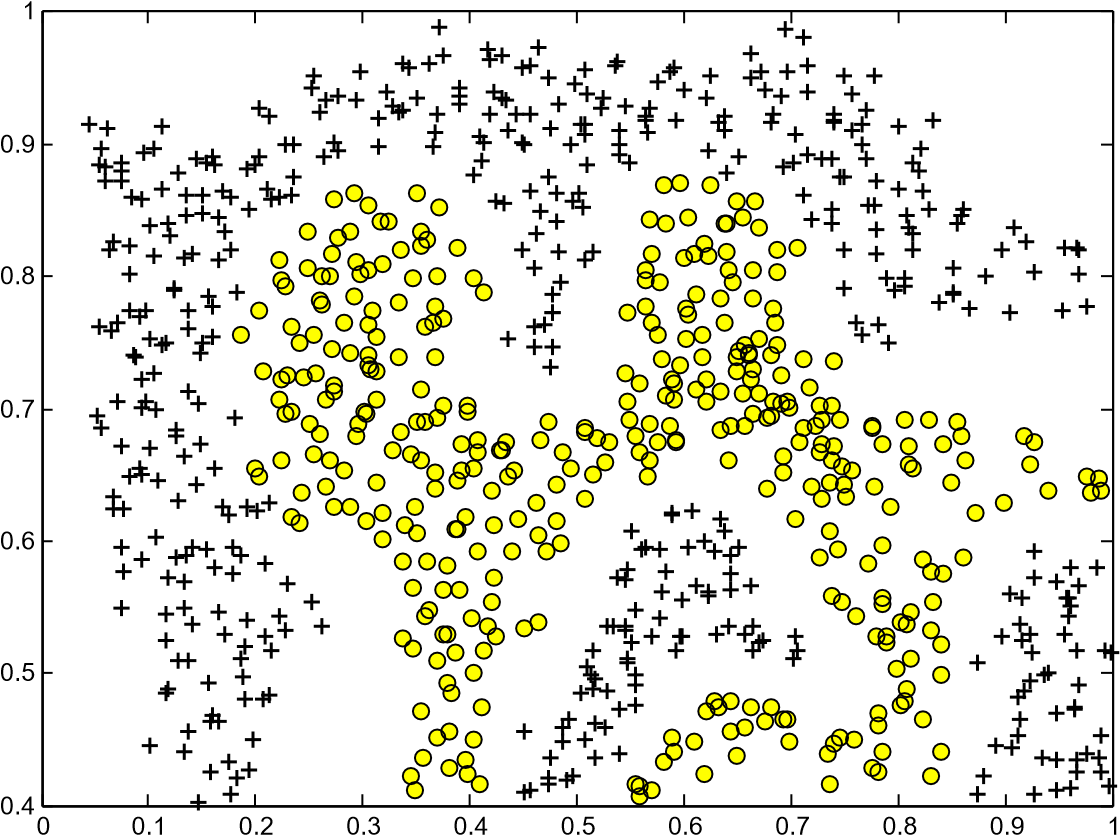


图 4：示例数据集 2

ex6.m 中的下一部分将加载并绘制数据集 2（图 4）。从图中可以看出， 没有线性决策边界将此数据集的正示例和负示例分开。但是，通过将高斯核与 SVM 结合使用，您将能够学习一个非线性决策边界，该边界可以很好地用于数据集。

如果您已正确实现高斯核函数， ex6.m 将继续在此数据集上使用高斯核训练 SVM。

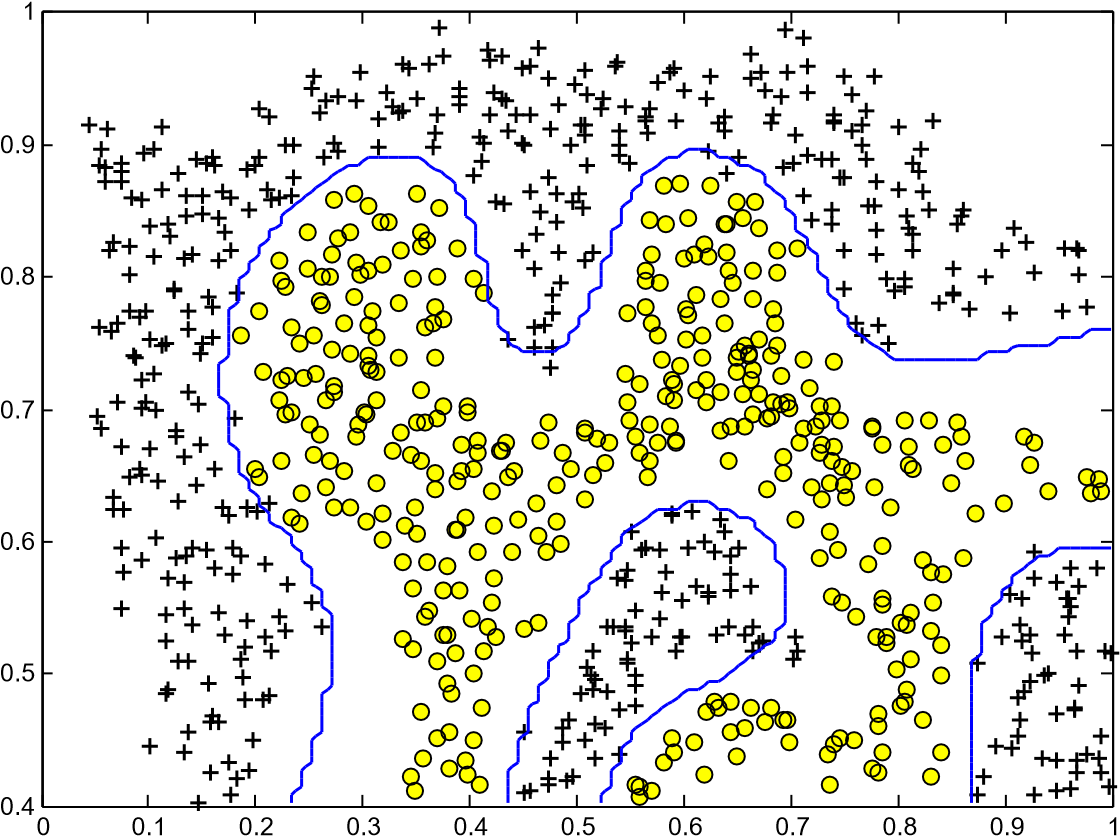


图 5：SVM（高斯 Kernel）决策边界（示例数据集 2）

图 5 显示了 SVM 与高斯核找到的决策边界。决策边界能够正确分离大多数正反示例，并很好地遵循数据的轮廓。

### 示例数据集 3

在本部分练习中，您将获得有关如何将 SVM 与高斯核配合使用的更多实践技能。ex6.m 的下一部分将加载并显示第三个数据集（图 6）。您将使用此数据集的 SVM 和高斯 kernel。

在提供的数据集 ex6data3.mat 中，您将获得变量 X、y、Xval、yval。ex6.m 中提供的代码使用从 dataset3Params.m 加载的参数，使用训练集 （X， y） 训练 SVM 分类器。

您的任务是使用交叉验证集 Xval， yval 来确定要使用的最佳 *C* 和*σ*参数。您应该编写任何其他必要的代码来帮助您搜索参数 *C* 和 *σ*。对于 *C* 和 *σ*，我们建议尝试以乘法ive 步长（例如，0*.*01*，*0*.*03*，*0*.*1*，*0*.*3*，*1*，*3*，*10*，*30）。请注意，您应该尝试 *C* 和 *σ*的所有可能的值对（例如，*C* = 0*。*3 和 *σ* = 0*。*例如，如果您为 *C* 和 *σ*2 尝试上面列出的 8 个值中的每一个，则最终将训练和评估（在交叉验证集上）总共 8 个2 = 64 个不同的模型。

确定要使用的最佳 *C* 和 *σ* 参数后，请修改 dataset3Params.m 中的代码，填写最佳参数

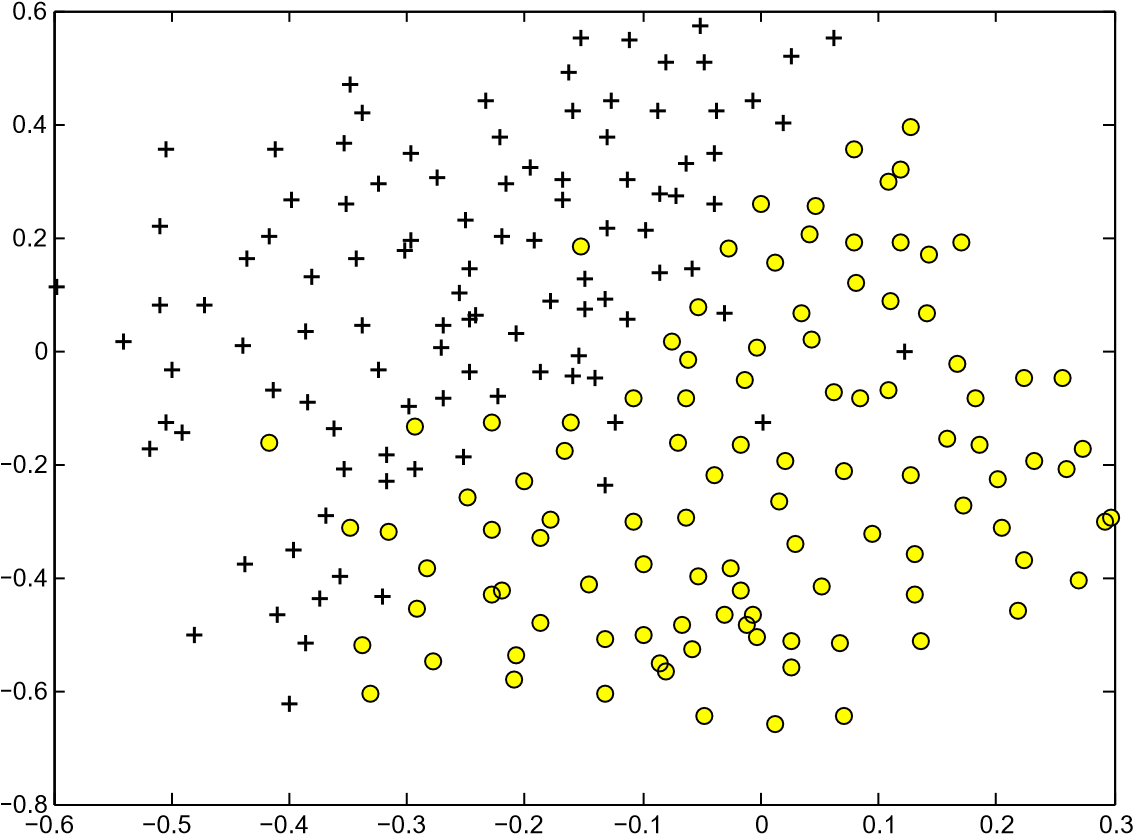


图 6：示例数据集 3

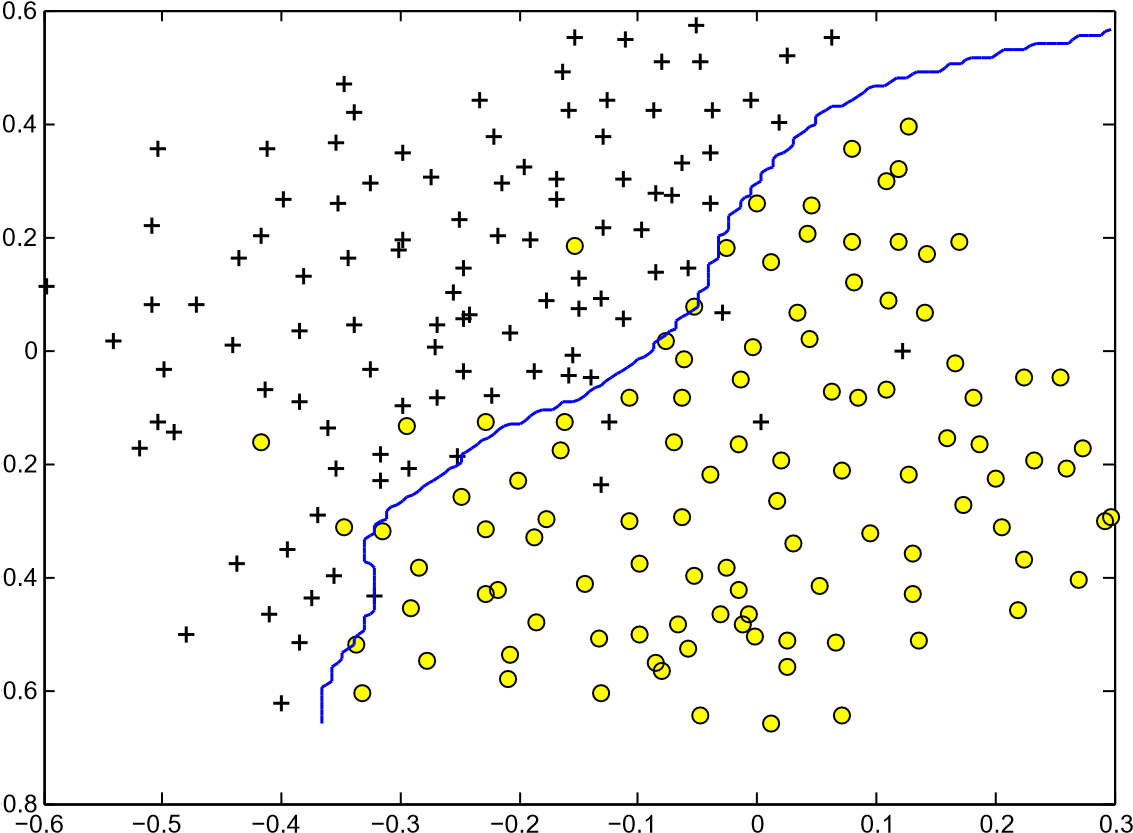


图 7：SVM（高斯核）决策边界（示例数据集 3）

你找到了。对于我们的最佳参数，SVM 返回了如图 7 所示的决策边界。

**实现提示：**在实现交叉验证以选择最佳 *C* 和*σ*参数时使用时，需要评估交叉验证集上的错误。回想一下，对于分类，误差被定义为分类不正确的交叉价值标识示例的分数。在Octave/MATLAB中，您可以使用meanta（double（predictions ~= yval））来计算此误差，其中预测是包含来自SVM的所有预测的向量，而yval是来自交叉验证集的真实标签。可以使用 svmPredict 函数为交叉验证集生成预测。

*您现在应该提交您的解决方案。*

# 垃圾邮件分类

如今，许多电子邮件服务都提供垃圾邮件过滤器，能够将电子邮件高精度地分类为垃圾邮件和非垃圾邮件。在本部分练习中，您将使用 SVM 构建自己的垃圾邮件筛选器。

您将训练一个分类器来对给定电子邮件*x*是垃圾邮件（*y* = 1）还是非垃圾邮件（*y* = 0）进行分类。特别是，您需要将每封电子邮件转换为特征向量*x*∈R*n*。练习的以下部分将引导您了解如何通过电子邮件构造此类特征向量。

在本练习的其余部分中，您将使用脚本 ex6 spam.m。本练习中包含的数据集基于 SpamAssassin 公共语料库的子集。[[3]](#footnote-3) 出于本练习的目的，您将仅使用电子邮件的正文（不包括电子邮件标头）。

## 预处理电子邮件

> 有谁知道托管门户网站需要多少钱 ？

>

好吧，这取决于您期望的访客数量。这可以是任何地方，从每月不到10美元到几美元100美元不等。您应该 http://www.rackspace.com/ 或Amazon EC2（如果您正在 运行大型活动）结帐。

要取消订阅此邮件列表，请发送电子邮件至：groupname-unsubscribe@egroups.com

图 8：示例电子邮件

在开始机器学习任务之前，通常先查看数据集中的示例。图 8 显示了一个示例电子邮件，其中包含 URL、电子邮件地址（在末尾）、数字和美元金额。虽然许多电子邮件将包含类似类型的实体（例如，数字，其他URL或其他电子邮件地址），但特定实体（例如，特定URL或特定美元金额） 在几乎每封电子邮件中都会有所不同。因此，处理电子邮件时经常采用的一种方法是“规范化”这些值，以便所有URL都得到相同的处理，所有数字都得到相同的处理，等等。例如，我们可以将电子邮件中的每个 URL 替换为 唯一字符串“httpaddr”，以指示存在 URL。

这样做的效果是，垃圾邮件分类器可以根据 *是否存在任何* URL 而不是特定 URL 是否存在来做出分类决策。这通常证明了垃圾邮件分类器的性能，因为垃圾邮件发送者经常随机化URL，因此在新的垃圾邮件中再次看到任何特定URL的几率非常小。

在 processEmail.m 中，我们实现了以下电子邮件预处理，并且没有重新映射步骤：

* **小写：**整个电子邮件被转换为小写，以便忽略大写（例如，IndIcaTE被视为与指示相同）。
* **剥离 HTML：** 从电子邮件中删除所有 HTML 标记。许多电子邮件通常带有HTML 格式;我们删除了所有HTML标签，以便仅保留内容。
* **规范化 URL：** 所有 URL 都替换为文本“httpaddr”。
* **规范化电子邮件地址：** 所有电子邮件地址都将替换为文本“emailaddr”。
* **规范化数字：** 所有数字都替换为文本

“数字”。

* **规范化美元：** 所有美元符号 （$） 都替换为文本“美元”。
* **词干：** 单词被简化为词干形式。例如，“折扣”、“折扣”、“折扣”和“折扣”都替换为“折扣”。有时，Stemmer实际上从末尾剥离了其他字符，因此“包含”，“包含”，“包含ed”和“包括”都被替换为“包含”。
* **删除非单词：** 删除了非单词 和标点符号。所有空格（制表符、换行符、空格）都已修剪为单个空格字符。

这些预处理步骤的结果如图 9 所示。虽然预处理留下了单词片段和非单词，但事实证明，这种形式在执行特征提取时要容易得多。

anyon知道托管一个门户网站的成本是多少， 这取决于 你的expect th的mani访问者可以从 每月不到数字到一堆美元，你应该结帐httpaddr或也许亚马逊ecnumb，如果你运行一些大到取消订阅 您自己从邮件列表发送电子邮件到电子邮件addr

图 9：预处理的示例电子邮件

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | 1. 机 管 局 2. 熄灭 3. 阿比尔 ...86 安永 ...916知道... 4. 零 5. 邮编 | | |  | | --- | | 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 | |

图 10：词汇表 图 11：示例电子邮件的单词索引

### 词汇表

预处理电子邮件后，我们为每个电子邮件提供了一个单词列表（例如，图 9）。下一步是选择要在分类器中使用的单词，以及我们想要省略的hich。

在本练习中，我们仅选择最常出现的单词作为我们考虑的单词集（词汇表）。由于在训练集中很少出现的单词只在几封电子邮件中出现，因此它们可能会导致模型过度拟合我们的训练集。完整的词汇表位于文件词汇表中.txt如图 10 所示。我们的词汇表是通过选择垃圾邮件语料库中至少出现100次的所有单词来选择的，从而产生1899个单词的列表。在实践中，经常使用大约10，000至50，000个单词的词汇表。

给定词汇表，我们现在可以将预处理电子邮件中的每个单词（例如，图 9）映射到包含vo cabulary列表中单词索引的单词索引列表中。图 11 显示了示例电子邮件的映射。具体来说，在示例电子邮件中，单词“任何人”首先被 规范化为“anyon”，然后映射到词汇表中的索引86。

您现在的任务是完成进程Email.m 中的代码以执行此映射。在代码中，您将获得一个字符串str，它是已处理电子邮件中的单个单词。您应该在词汇表词汇表中查找该单词 ，并查找该单词是否存在于词汇表中。如果该词存在，则 you 应将该词的索引添加到词索引变量中。如果该单词不存在，因此不在词汇表中，则可以跳过该单词。

实现 processEmail.m 后，脚本 ex6 spam.m 将在电子邮件示例上运行您的代码，您应该会看到类似于图 9 和图 11 的输出。

**Octave/MATLAB 提示：**在 Octave/MATLAB 中，您可以使用 strcmp 函数比较两个字符串。例如，strcmp（str1， str2） 仅在两个字符串相等时才返回 1。在提供的起始代码中，vocabList 是一个包含词汇表中单词的“单元格数组”。在Octave/MATLAB中，单元格数组就像普通数组（即向量）一样，除了它的元素也可以是字符串（它们在正常的Octave/MATLAB矩阵/向量中不能），并且你使用大括号而不是方括号来索引它们。具体来说，要获取索引 i 处的单词，可以使用 vocabList{i}。您还可以使用 length（vocabList） 来获取词汇表中的单词数。

*您现在应该提交您的解决方案。*

## 从电子邮件中提取特征

现在，您将实现将每封电子邮件转换为 R*n* 中的向量的功能提取。在本练习中，您将在 vocabu 拉里列表中使用 *n* = # 个单词。具体来说，电子邮件的功能 *xi* ∈{0*，*1} 对应于字典中的第 *i* 个单词是否出现在电子邮件中。也就是说，如果电子邮件中存在第 i 个单词，则 *x* *i*= 1;如果 电子邮件中不存在第 i 个单词，则 *x* *i* = 0。

因此，对于 typical 电子邮件，此功能将如下所示：

 0 

.

 .. 

 

 1 

 

 0 

  *x* =  ... ∈R*n.*

 

 1 

 

 0 

 ... 





0

现在，您应该完成 emailFeatures.m 中的代码，以便在给定单词 index 的情况下为电子邮件生成特征向量。

实现 emailFeatures.m 后，ex6 spam.m 的下一部分 将在电子邮件示例上运行代码。您应该看到特征向量的长度为 1899，有 45 个非零条目。

*您现在应该提交您的解决方案。*

## 针对垃圾邮件分类训练 SVM

完成特征提取函数后，ex6 spam.m 的下一步 将加载预处理的训练数据集，该数据集将用于训练 SVM 分类器。 spamTrain.mat 包含 4000 个垃圾邮件和非垃圾邮件的训练示例，而 spamTest.mat 包含 1000 个测试示例。使用 processEmail 和 emailFeatures 函数处理的每封原始电子邮件 w 都函数，并转换为向量 *x*（*i*） ∈R1899。

加载数据集后， ex6 spam.m 将继续训练 SVM 以在垃圾邮件 （*y* = 1） 和非垃圾邮件 （*y* = 0） 电子邮件之间进行分类。训练 完成后，您应该看到分类器的训练准确率约为 99.8%，测试准确率约为 98.5%。

## 垃圾邮件的主要预测因素

我们的点击 删除 保证 访问 基数美元 将价格恳求 nbsp 大多数lo ga dollarnumb

图 12：垃圾邮件的主要预测因素

为了更好地理解垃圾邮件分类器的工作原理，我们可以检查参数，以查看分类器认为哪些单词最能预测垃圾邮件。ex6 spam.m 的下一步是在分类器中查找具有最大正值的参数 s，并显示相应的单词（图 12）。因此，如果电子邮件包含“保证”、“删除”、“美元”和“价格”等词（图 12 中所示的顶级预测因子），则很可能被归类为垃圾邮件。

## 可选（未评分）练习：尝试您自己的电子邮件

现在您已经训练了垃圾邮件分类器，您可以开始在自己的电子邮件上尝试一下。在起始代码中，我们包含了两个电子邮件示例（emailSample1.txt 和 emailSample2.txt）和两个垃圾邮件示例（spamSample1.txt 和 spamSample2.txt）。ex6 spam.m 的 l ast 部分在第一个 垃圾邮件示例上运行垃圾邮件分类器，并使用学习的 SVM 对其进行分类。您现在应该尝试我们提供的其他示例，看看分类器是否正确。您还可以尝试自己的电子邮件，方法是将示例（纯文本文件）与您自己的电子邮件重新连接。

*您无需为此可选（未评分）练习提交任何解决方案。*

## 可选（未评分）练习：构建自己的数据集

在本练习中，我们提供了一个预处理的训练集和 test 集。这些数据集是使用您现在已完成的相同函数（processEmail.m 和 emailFeatures.m）创建的。对于此可选（未评分）练习，您将使用 [SpamAssassin PubLic Corpus](http://spamassassin.apache.org/publiccorpus/) 中的原始电子邮件构建自己的数据集[。](http://spamassassin.apache.org/publiccorpus/)

在此可选（未评分）练习中，您的任务是从公共语料库下载原始文件并提取它们。提取它们后，您应该对每封电子邮件 运行 processEmail[[4]](#footnote-4) 和 emailFeatures 函数，以从ea ch电子邮件中提取特征向量。这将允许您构建一个数据集 X， y 的示例。然后，应将数据集随机划分为训练集、交叉验证集和测试集。

在构建自己的数据集时，我们还鼓励您尝试构建自己的词汇表（通过选择数据集中出现的高频单词）并添加您认为可能有用的任何其他功能。

最后，我们还建议尝试使用高度优化的 SVM 工具箱，如 [LIBSVM。](http://www.csie.ntu.edu.tw/~cjlin/libsvm/)

*您无需为此可选（未评分）练习提交任何解决方案。*

# 提交和评分

完成任务的各个部分后，请务必使用 提交 功能系统将您的解决方案提交给我们的服务器。以下是本练习每个部分的评分方式的细分。

|  |  |  |
| --- | --- | --- |
| **部分** | **提交的文件** | **点** |
| 高斯核 | 高斯内核.m | 25 积分 |
| 数据集 3 的参数 （*C*、*σ*） | dataset3Params.m | 25 积分 |
| 电子邮件预处理 | processEmail.m | 25 积分 |
| 电子邮件功能提取 | emailFeatures.m | 25 积分 |
| 总积分 |  | 100 积分 |

您可以多次提交解决方案，我们将仅考虑最高分。

1. Octave 是 MATLAB 的免费替代品。对于编程练习，您可以自由使用Octave或MATLAB。 [↑](#footnote-ref-1)
2. 为了确保与Octave / MATLAB的兼容性，我们包含了SVM学习算法的这种实现。但是，这部分选择 icular 实现 以最大限度地提高兼容性，并且**不**非常高效。如果您正在针对实际问题训练 SVM，特别是如果您需要扩展到更大的数据集，我们强烈建议您改用高度优化的 SVM 工具箱解决方案h 作为[LIBSVM.](http://www.csie.ntu.edu.tw/~cjlin/libsvm/) [↑](#footnote-ref-2)
3. <http://spamassassin.apache.org/publiccorpus/> [↑](#footnote-ref-3)
4. 原文inal电子邮件将具有您可能希望省略的电子邮件标题。我们已将代码包含在进程电子邮件 这将帮助您删除这些标头。 [↑](#footnote-ref-4)