LIBSVM

1 LIBSVM 简介

LIBSVM 是台湾大学林智仁(Lin Chih-Jen)副教授等开发设计的一个简单、易于使用和快速有效的 SVM 模式识别与回归的软件包,他不但提供了编译好的可在 Windows 系列系统的执行文件，还提供了源代码，方便改进、修改以及在其它操作系统上应用；该软件还有一个特点，就是对 SVM 所涉及的参数调节相对比较少，提供了很多的默认参数，利用这些默认参数就可以解决很多问题；并且提供了交互检验(Cross -SVM 回归等问题，包括基于一对一算法的多类模式识别问题。SVM 用于模式识别或回归时，SVM 方法及其参数、核函数及其参数的选择,目前国际上还没有形成一个统一的模式，也就是说最优 SVM 算法参数选择还只能是凭借经验、实验对比、大范围的搜寻或者利用软件包提供的交互检验功能进行寻优。ν-SVM 回归和ε-SVM 分类、νValidation)的功能。该软件包可以在 http://www.csie.ntu.edu.tw/~cjlin/免费获得。该软件可以解决 C-SVM 分类、-SVM 回归等问题，包括基于一对一算法的多类模式识别问题。SVM 用于模式识别或回归时，SVM 方法及其参数、核函数及其参数的选择，目前国际上还没有形成一个统一的模式，也就是说最优 SVM 算法参数选择还只能是凭借经验、实验对比、大范围的搜寻或者利用软件包提供的交互检验功能进行寻优。

2 LIBSVM 使用方法

LibSVM 是以源代码和可执行文件两种方式给出的。如果是 Windows 系列操作系统，可以直接使用软件包提供的程序，也可以进行修改编译；如果是 Unix 类系统，必须自己编译，软件包中提供了编译格式文件，我们在 SGI 工作站(操作系统 IRIX6.5)上，使用免费编译器 GNU C++3.3 编译通过。

2.1 LIBSVM 使用的一般步骤：

1) 按照 LIBSVM 软件包所要求的格式准备数据集；
2) 对数据进行简单的缩放操作；
3) 考虑选用 RBF 核函数；
4) 采用交叉验证选择最佳参数 C 与 g；
5) 采用最佳参数 C 与 g 对整个训练集进行训练获取支持向量机模型；
6) 利用获取的模型进行测试与预测。

2.2 LIBSVM 使用的数据格式

该软件使用的训练数据和检验数据文件格式如下：

<label> <index1>:<value1> <index2>:<value2> ...

其中<label> 是训练数据集的目标值，对于分类，它是标识某类的整数(支持多个类)；对于回归，是任意实数。<index> 是以 1 开始的整数，可以是不连续的；<value> 为实数,也就是我们常说的自变量。检验数据文件中的 label 只用于计算准确度或误差，如果它是未知的，只需用一个数填写这一栏，也可以空着不填。在程序包中，还包括有一个训练数据实例：heart_scale，方便参考数据文件格式以及练习使用软件。可以编写小程序，将自己常用的数据格式转换成这种格式。

2.3 Svmtrain 和 Svmpredict 的用法

Svmtrain(训练建模)的用法：

svmtrain [options] training_set_file [model_file]

Options：可用的选项即表示的涵义如下

-s svm 类型：SVM 设置类型(默认 0)

0 -- C-SVC

1 --v-SVC

2 – 一类 SVM

3 -- e -SVR

4 -- v-SVR

-t 核函数类型：核函数设置类型(默认 2)

0 – 线性：u'v

1 – 多项式：(r*u'v + coef0)^degree

2 – RBF 函数：exp(-r|u-v|^2)

3 –sigmoid：tanh(r*u'v + coef0)

-d degree：核函数中的 degree 设置(默认 3)

-g r(gama)：核函数中的?函数设置(默认 1/ k)

-r coef0：核函数中的 coef0 设置(默认 0)

-c cost：设置 C-SVC，? -SVR 和?-SVR 的参数(默认 1)

-n nu：设置?-SVC，一类 SVM 和?- SVR 的参数(默认 0.5)

-p e：设置? -SVR 中损失函数?的值(默认 0.1)

-m cachesize：设置 cache 内存大小，以 MB 为单位(默认 40)

-e ε：设置允许的终止判据(默认 0.001)

-h shrinking：是否使用启发式，0 或 1(默认 1)

-wi weight：设置第几类的参数 C 为 weight?C(C-SVC 中的 C)(默认 1)

-v n: n-fold 交互检验模式

其中-g 选项中的 k 是指输入数据中的属性数。option -v 随机地将数据剖分为 n 部分并计算交互检验准确度和均方根误差。以上这些参数设置可以按照 SVM 的类型和核函数所支持的参数进行任意组合，如果设置的参数在函数或 SVM 类型中没有也不会产生影响，程序不会接受该参数；如果应有的参数设置不正确，参数将采用默认值。

training_set_file 是要进行训练的数据集；model_file 是训练结束后产生的模型文件，文件中包括支持向量样本数、支持向量样本以及 Lagrange 系数等必须的参数；该参数如果不设置将采用默认的文件名，也可以设置成自己惯用的文件名。

2.4 Svmpredict 的用法

Svmpredict(使用已有的模型进行预测)的用法：

svmpredict test_file model_file output_file

model_file 是由 svmtrain 产生的模型文件；test_file 是要进行预测的数据文件；Output_file 是 svmpredict 的输出文件。svm-predict 没有其它的选项。

svmtrain -s 0 -c 1000 -t 1 -g 1 -r 1 -d 3 data_file

训练一个由多项式核(u'v+1)^3 和 C=1000 组成的分类器。

svmtrain -s 1 -n 0.1 -t 2 -g 0.5 -e 0.00001 data_file

在 RBF 核函数 exp(-0.5|u-v|^2)和终止允许限 0.00001 的条件下，训练一个?-SVM (? = 0.1)分类器。

svmtrain -s 3 -p 0.1 -t 0 -c 10 data_file

以线性核函数 u'v 和 C=10 及损失函数?= 0.1 求解 SVM 回归。

LIBSVM 使用笔记

1 程序准备：

C:\Python25 //需要安装 python2.5

E:\svm\libsvm-2.83 //解压缩 libsvm-2.83

E:\svm\libsvm-2.83\gp373w32 //画图工具 gnuplot

E:\svm\libsvm-2.83\python //SVM, cross validation

E:\svm\libsvm-2.83\tools //easy.py, grid.py, subset.py

E:\svm\libsvm-2.83\tools1 //生成所需格式的数据，两种方式：matlab 或 excel

E:\svm\libsvm-2.83\windows //windows 下面的 4 个可执行程序

2 修改 E:\svm\libsvm-2.83\tools\easy.py：

程序开始加入以下两句：

svmpath = "E:\svm\libsvm-2.83\windows" // easy 需要调用 svm 可执行程序

pythonpath = "E:\svm\libsvm-2.83\tools" // easy 需要调用 grid.py

gnuplot_exe = r"c:\tmp\gnuplot\bin\pgnuplot.exe" 改为：

gnuplot_exe = r"..\gp373w32\pgnuplot.exe"

grid_py = r".\grid.py" 改为：grid_py = r"..\tools\grid.py"

3 修改 E:\svm\libsvm-2.83\tools\grid.py：

程序开始加入以下两句：

svmpath = "E:\svm\libsvm-2.83\windows" // grid 需要调用 svm 可执行程序

gnuplotpath = "E:\svm\libsvm-2.83\gp373w32" // grid 需要调用画图工具

gnuplot_exe = r"c:\tmp\gnuplot\bin\pgnuplot.exe" 改为：

gnuplot_exe = r"..\gp373w32\pgnuplot.exe"

4 数据准备：

1、 使用 excel 宏

手工转换数据格式，可以通过 excel 文件 FormatDataLibsvm.xls 打开包含数据的文本文件，然后用宏将数据转换为 libsvm 格式。也可以用宏将 libsvm 格式的数据转换回来。

2、使用 matlab

FormatSplitBat.m 是一个批量将以 Tab 或者空格分隔的数据转换为 libsvm 格式，并且分成占全部数据比例为 p 的训练集和 1-p 的测试集，然后调用 easy.py 的例子。//摘自 Libsvm-2_6.mht

修改程序中的三处路径与本机实际存储情况一致，并确保 easy.py 的第 28 行能够正确调用 grid.py，之后将 fileList.txt 文件中写入要转换格式的文件列表。这些文件名都在当前路径(E:\svm\libsvm-2.83\data)之下。

运行 E:\svm\libsvm-2.83\Tools1\FormatSplitBat.m，程序自动将 fileList.txt 中提到的文件名的数据内容按照概率 p 进行随机划分，本程序 p 为 0.5，则有一半划分到 train，一半划分到 test。另外，本程序划分是循环 10 次，可以按照自己的要求进行调整。每次循环将每个原数据文档在文件夹 E:\svm\libsvm-2.83\data 中生成 4 个文件，分别是：file_i.train, file_i.test, file_iL.train, file_iL.test. 其中，L 表示标示变量在左边，为 libsvm 能够使用的文件。

运行过程中，每一次循环程序都要调用 easy.py，grid.py 默认采用 5 倍交叉验证，对 c 采用的搜索范围和步长分别为[-10, 15]和 1，g 的为[10, -15]和-1，可根据需要自行修改 grid.py。运行完 grid.py，在..\tools1\中每两个有 L 的文件(train, test)生成了 7 个文件，分别是 train5 个，test2 个。其中 train 中生成 3 个 scale (-, out, png)，一个 range，一个 model。Test 中一个 scale，一个 predict。

利用 Libsvm 进行 SVM 分类

SVM 分类器通常具有较高的分类精度。我这里不想过多的去说 SVM 是怎么回事，只是提供一种使用 SVM 进行判别的方法。我使用的是开源的 LibSVM 实现 SVM 分类。Google 上输入 LIBSVM 可以很容易的找到代码下载。我使用的是 C#版（不过是 2.6 版），也可以使用 C++的 2.81 版。下面我说说如何使用 2.81 版中带的编译好的程序完成聚类工作。该版本支持多类判别。

1、数据准备工作

首先输入数据进行处理，生成符合 SVM 处理规范的文本格式。关于格式的更多说明可以参考软件使用手册。另外此步可以不用归一化，因为 LibSVM 工具中提供了 scale 工具，可以自动完成数据的归一化处理工作。

2、程序配置工作

LibSVM 2.81 版下载下来后并不能直接操作，还需要一些辅助工作，否则在默认判别的方式下工作判别精度可能非常低。关于此方面的更多内容可以参考《A Practical Guide to Support Vector Classification》一文，该 PDF 文档可以从 LibSVM 的网站上下载到。

在使用 LibSVM 之前首先要安装 Pathon，Pathon 2.4 可以从 Pathon 的网站上下载到。

紧接着就是需要装“pgnuplot.exe”，LibSVM 使用它完成参数搜索时的绘图工作，该程序没有包含在 LibSVM 2.81 版的压缩包中，需要自己到网上搜索并下载。另外在 LibSVM 2.81 中 grid.py 代码里默认 pgnuplot.exe 的路径是“c:\tmp\gnuplot\bin\pgnuplot.exe”，你可以将“pgnuplot.exe”放到该路径下或修改 grid.py 代码指向你自己的路径。

所有这些准备工作完成后，就可以进行 SVM 分类工作了。

3、SVM 分类

使用 SVM 分类可以执行“libsvm-2.81\tools”目录下的 easy.py 程序，该程序提供了一套默认的、精度较高的 SVM 分类算法。

在 DOS 窗口下输入：C:\Python\Python easy.py Train.txt Test.txt 就可以利用 Train.txt 中的数据进行训练，然后对 Test.txt 中的数据进行判别。

LIBSVM 入门

许多想用 lib-svm 解决分类或者回归的初学者可能像我一样一开始摸不着头绪。面对长篇的 English，头都大了。有好心人把自己的经验总结了，我们就一起共享吧！

1. LIBSVM 软件包简介

LIBSVM 是台湾大学林智仁(Chih-Jen Lin)博士等开发设计的一个操作简单、易于使用、快速有效的通用 SVM 软件包，可以解决分类问题（包括 C- SVC、n - SVC ）、回归问题（包括 e - SVR、n - SVR ）以及分布估计（one-class-SVM ）等问题，提供了线性、多项式、径向基和 S 形函数四种常用的核函数供选择，可以有效地解决多类问题、交叉验证选择参数、对不平衡样本加权、多类问题的概率估计等。LIBSVM 是一个开源的软件包，需要者都可以免费的从作者的个人主页 http://www.csie.ntu.edu.tw/~cjlin/ 处获得。他不仅提供了 LIBSVM 的 C++语言的算法源代码，还提供了 Python、Java、R、MATLAB、Perl、Ruby、LabVIEW 以及 C#.net 等各种语言的接口，可以方便的在 Windows 或 UNIX 平台下使用，也便于科研工作者根据自己的需要进行改进（譬如设计使用符合自己特定问题需要的核函数等）。另外还提供了 WINDOWS 平台下的可视化操作工具 SVM-toy，并且在进行模型参数选择时可以绘制出交叉验证精度的等高线图。

2. LIBSVM 使用方法简介

LIBSVM 在给出源代码的同时还提供了 Windows 操作系统下的可执行文件，包括：进行支持向量机训练的 svmtrain.exe；根据已获得的支持向量机模型对数据集进行预测的 svmpredict.exe；以及对训练数据与测试数据进行简单缩操作的 svmscale.exe。它们都可以直接在 DOS 环境中使用。如果下载的包中只有 C++的源代码，则也可以自己在 VC 等软件上编译生成可执行文件。

LIBSVM 使用的一般步骤是：

1） 按照 LIBSVM 软件包所要求的格式准备数据集；

2） 对数据进行简单的缩放操作；

3） 考虑选用 RBF 核函数 2 K(x,y) e x y = -g - ；

4） 采用交叉验证选择最佳参数 C 与 g ；

5） 采用最佳参数 C 与 g 对整个训练集进行训练获取支持向量机模型；

6） 利用获取的模型进行测试与预测。

2.1 LIBSVM 使用的数据格式

LIBSVM 使用的训练数据和测试数据文件格式如下:： ：< 2> …其中 是训练数据集的目标值，对于分类，它是标识某类的整数(支持多个类)；对于回归，是任意实数。是以 1 开始的整数，表示特征的序号；为实数，也就是我们常说的特征值或自变量。当特征值为 0 时，特征序号与特征值都可以同时省略，即 index 可以是不连续的自然数。与第一个特征序号、前一个特征值与后一个特征序号之间用空格隔开。测试数据文件中的 label 只用于计算准确度或误差，如果它是未知的，只需用任意一个数填写这一栏，也可以空着不填。例如：

+1 1:0.708 2:1 3:1 4:-0.320 5:-0.105 6:-1 8:1.21

2.2 svmscale 的用法

对数据集进行缩放的目的在于：

1）避免一些特征值范围过大而另一些特征值范围过小；

2）避免在训练时为了计算核函数而计算内积的时候引起数值计算的困难。因此，通常

将数据缩放到[-1,1]或者是[0,1]之间。用法：svmscale [-l lower] [-u upper] [-y y_lower y_upper] [-s save_filename] [-r restore_filename] filename（缺省值： lower = -1，upper = 1，没有对 y 进行缩放）其中，

-l：数据下限标记；lower：缩放后数据下限；

-u：数据上限标记；upper：缩放后数据上限；

-y：是否对目标值同时进行缩放；y_lower 为下限值，y_upper 为上限值；

-s save_filename：表示将缩放的规则保存为文件 save_filename；

-r restore_filename：表示将缩放规则文件 restore_filename 载入后按此缩放；

filename：待缩放的数据文件（要求满足前面所述的格式）。

缩放规则文件可以用文本浏览器打开，看到其格式为：

lower upper

lval1 uval1

lval2 uval2

其中的 lower 与 upper 与使用时所设置的 lower 与 upper 含义相同；index 表示特征序号；lval 为该特征对应转换后下限 lower 的特征值；uval 为对应于转换后上限 upper 的特征值。数据集的缩放结果在此情况下通过 DOS 窗口输出，当然也可以通过 DOS 的文件重定向符号“>”将结果另存为指定的文件。

使用实例：

1) svmscale –s train3.range train3>train3.scale 表示采用缺省值（即对属性值缩放到[-1,1]的范围，对目标值不进行缩放）对数据集 train3 进行缩放操作，其结果缩放规则文件保存为 train3.range，缩放集的缩放结果保存为 train3.scale。

2）svmscale –r train3.range test3>test3.scale 表示载入缩放规则 train3.range 后按照其上下限对应的特征值和上下限值线性的地对数据集 test3 进行缩放，结果保存为 test3.scale。

2.3 svmtrain 的用法

svmtrain 实现对训练数据集的训练，获得 SVM 模型。

用法： svmtrain [options] training_set_file [model_file]其中，

options（操作参数）：可用的选项即表示的涵义如下所示

-s svm 类型：设置 SVM 类型，默认值为 0，可选类型有：

0 -- C- SVC

1 -- n – SVC

2 -- one-class-SVM

3 -- e – SVR

4 -- n – SVR

-t 核函数类型：设置核函数类型，默认值为 2，可选类型有：

0 -- 线性核：u'*v

1 -- 多项式核： (g*u'*v+ coef 0)degree

2 -- RBF 核：e(u v 2) g –

3 -- sigmoid 核：tanh(g*u'*v+ coef 0)

-d degree：核函数中的 degree 设置，默认值为 3； -g g ：设置核函数中的 g ，默认值为 1/ k ；

-r coef 0：设置核函数中的 coef 0，默认值为 0；

-c cost：设置 C- SVC、e - SVR、n - SVR 中从惩罚系数 C，默认值为 1；

-n n ： 设置 n - SVC、one-class-SVM 与 n - SVR 中参数 n ，默认值 0.5；

-p e ： 设置 n - SVR 的损失函数中的 e ，默认值为 0.1；

-m cachesize：设置 cache 内存大小，以 MB 为单位，默认值为 40；

-e e ：设置终止准则中的可容忍偏差，默认值为 0.001；
-h shrinking：是否使用启发式，可选值为 0 或 1，默认值为 1；
-b 概率估计：是否计算 SVC 或 SVR 的概率估计，可选值 0 或 1，默认 0；
-wi weight：对各类样本的惩罚系数 C 加权，默认值为 1；
-v n：n 折交叉验证模式。其中-g 选项中的 k 是指输入数据中的属性数。操作参数 -v 随机地将数据剖分为 n 部分并计算交叉检验准确度和均方根误差。以上这些参数设置可以按照 SVM 的类型和核函数所支持的参数进行任意组合，如果设置的参数在函数或 SVM 类型中没有也不会产生影响，程序不会接受该参数；如果应有的参数设置不正确，参数将采用默认值。training_set_file 是要进行训练的数据集；model_file 是训练结束后产生的模型文件，该参数如果不设置将采用默认的文件名，也可以设置成自己惯用的文件名。
使用实例：
1）svmtrain train3.scale train3.model 训练 train3.scale，将模型保存于文件 train3.model，并在 dos 窗口中输出如下结果：
optimization finished, #iter = 1756
nu = 0.464223
obj = -551.002342, rho = -0.337784
nSV = 604, nBSV = 557
Total nSV = 604 其中，#iter 为迭代次数，nu 与前面的操作参数-n n 相同，obj 为 SVM 文件转换为的二次规划求解得到的最小值，rho 为判决函数的常数项 b，nSV 为支持向量个数，nBSV 为边界上的支持向量个数，Total nSV 为支持向量总个数。训练后的模型保存为文件 train3.model，用记事本等文本浏览器打开可以看到其内容如下（其后“%”后内容为笔者所加注释）：
svm_type c_svc % 训练所采用的 svm 类型，此处为 C- SVC
kernel_type rbf % 训练采用的核函数类型，此处为 RBF 核
gamma 0.047619 % 与操作参数设置中的 g 含义相同
nr_class 2 % 分类时的类别数，此处为两分类问题
total_sv 604 % 总共的支持向量个数
rho -0.337784 % 决策函数中的常数项 b
label 0 1 % 类别标签
nr_sv 314 290 % 各类别标签对应的支持向量个数
SV % 以下为支持向量
1 1:-0.963808 2:0.906788 ... 19:-0.197706 20:-0.928853 21:-1
1 1:-0.885128 2:0.768219 ... 19:-0.452573 20:-0.980591 21:-1... ... ...
1 1:-0.847359 2:0.485921 ... 19:-0.541457 20:-0.989077 21:-1
% 对于分类问题，上面的支持向量的各列含义与训练数据集相同；对于回归问题，略有不同，与训练数据中的标签 label（即 y 值）所对应的位置在模型文件的支持向量中现在存放的是 Lagrange 系数 a 值，即为下面决策函数公式中的 a 值：
* *
1
() ( )(( ) ( )) ( ) (,)
(,)
K
i i i i i i
i i sv

i i
i sv
fx a a x x b a a k x x b
ak x x b
= = - F F + = - +
= + g

2.4 svmpredict 的用法

svmpredict 是根据训练获得的模型，对数据集合进行预测。
用法：svmpredict [options] test_file model_file output_file
options（操作参数）：
-b probability_estimates：是否需要进行概率估计预测，可选值为 0 或者 1，默认值为 0。
model_file 是由 svmtrain 产生的模型文件；test_file 是要进行预测的数据文件；output_file 是 svmpredict 的输出文件，表示预测的结果值。svmpredict 没有其它的选项。