清 华 大 学

综 合 论 文 训 练

题目：主成分分析在引力波数据处理中的应用

系　　别：自动化系

专　　业：自动化

姓　　名：张政

指导教师：曹军威

2014 年 6 月 10 日

关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：学校有权保留学位论文的复印件，允许该论文被查阅和借阅；学校可以公布该论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存该论文。

(涉密的学位论文在解密后应遵守此规定)

签　名：_____导师签名：_____日　期：_____

中文摘要

关于万有引力的本质问题，从之前的超时距的控制到爱因斯坦的引力波理论，一直都在被争论。引力波在时空中的扰动会引发空间的扭曲和释放能量，这样我们可以通过各种各样的方式测量它的电磁辐射。引力波的测量通常来说是比较困难的，原因在于引力波的强度和距离成反比。

LIGO 作为测量引力波(时空结构波动)强有力的工具，提供了大量宝贵的实验数据。本篇论文首先对 10 组 LIGO S6 数据集进行特性分析，然后在这个基础上使用主成分分析压缩数据，减少不必要的信息并且使用多种可能改进机器学习最终效果的方法，分析多种方法的组合分析对机器学习方法的最终影响，比较结果得出结论并提出改进策略。

关键词：LIGO；引力波；主成分分析；机器学习；LIBSVM

ABSTRACT

The nature of the gravitation is always a hot topic from earlier idea that it is an effect beyond time and distance to modern physical description that it is essentially a form of gravity-wave put forward by Einstein. The gravity-wave could lead to distortion of time and space and release energy, thus we can detect electromagnetic radiation in different ways. Gravity-wave is difficult to be detected because the strength diminishes with distance.

LIGO is a tool to detect gravity wave which is the distortion of space and time. The experiments in this paper use ten data sets from LIGO S6 to analyze the characteristics at first. Then on the base of it using PCA to analyze all channels to apply reduction on original data and analyze the final effect on machine learning with combination of some methods. Finally make some conclusions and raise some improvements.

Keyword: LIGO; gravity wave; PCA; machine learning; LIBSVM

目 录

第1章 引 言

1.1 LIGO 与引力波

在 1916 年，爱因斯坦的广义相对论为揭开宇宙内幕奠定了不可缺少的基础。它的理论描述了时间和空间是怎么被质量所影响的。我们可以将时空想象为一块布料，当我们放东西上去的时候，它就会弯曲。记住 2 维的布料这个比喻恰恰是我们用来描述 4 维时空（时间和 3 维空间）的模型，见图 1.1[23]。

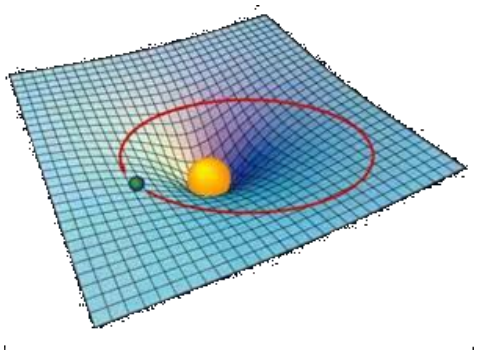


图 1.1 质量怎样影响空间和时间

关于量子计算机的物理实现，目前已经有一些可行方案，诸如液态核磁共振量子计算机，固态核磁共振量子计算机，离子阱方案，腔室量子电动力学（cavity QED）方案等。2012 年 10 月 9 日，美国科学家大卫 维因兰德((David Wineland) 和法国科学家塞尔日 阿罗什(Serge Haroche) 获得诺贝尔物理学奖，因为发现了“测量和操控单个量子系统的突破性实验方法“。他们这是因为在离子阱中囚禁和操纵少量量子上获得突破进展而获得这一荣誉的。

牛顿曾从经典力学角度提出万有引力是一种超时空的作用力，而爱因斯坦在近现代提出引力其实是一种波。很多科学家都把引力波描述成为“时空中的波纹”。就像一艘船在大海中行进产生波纹一样，星星或者是黑洞这样的移动物体在时空中产生引力的波纹。一个更大质量的物体会产生更大的波动而一个移动更快的物体则会在一定时间内产生更多波纹。

引力波通常是在两个甚至更多联系紧密的物体相互作用中产生的。这些作用包括两个黑洞的双星轨道，两个星系的合并或者是两个中子星互相绕转。对于黑洞，恒星和星系互相绕转，他们发出引力辐射的波纹会到达地球，但是它们一旦到达地球，就会变得很弱。这是因为就像水纹一样在离开发生源时强度会慢慢减弱。但是尽管它们很弱，它们也可以通畅地穿过时空。对于双星绕转而言，绕转的频率就是引力波的频率。引力波一般可以使用频带分类。1-10Hkz 是高频波源，来自中子星、黑洞等。1mHz-1Hz 称为低频波源，来自超重黑洞、白矮星等。探测引力波究竟有什么作用？首先是探测引力波可以打开一种观测宇宙的全新的方式，因为引力波所携带的信息和通常电磁波的不同。其次观测引力波有助于加深我们对物理本质的理解，引力波的存在能验证广义相对论的正确性。我们现在所知道的是引力波是一种信号。它的形状取决于发生源的引力场的变换。任何人在波的路径上都会在与其传播方向垂直面上感受到定时涨落的振动的引力。如果这样的话，那么我们可以通过监视物体小的变动就能知道是否有引力波经过。科学家尝试了各种各样的方式来测量微小的变化，其中干涉测量法是宇航员测量时空拉伸的技术。这项技术要求测试物体相互距离非常远。激光器将连续测量每个物体之间的距离。测试物体可以随意移动因此当引力波经过的时候，两个物体之间的距离就会波动。也就是说时空被扭曲了。激光器记录这些扰动并且科学家通过

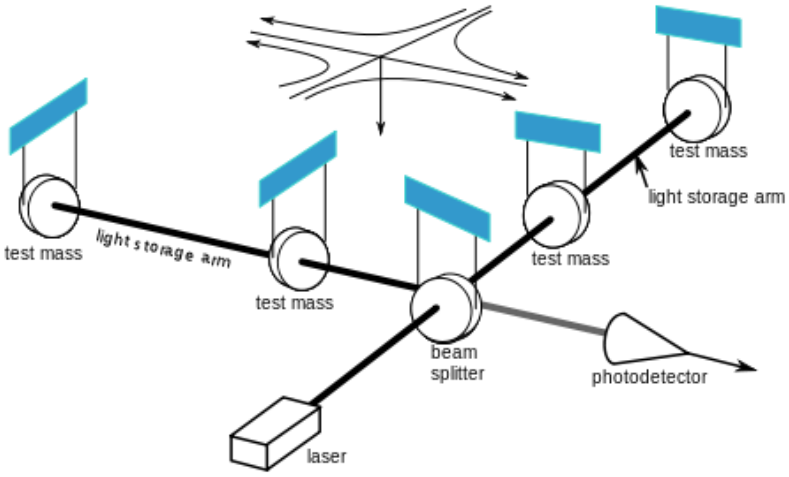


图 1.2　LIGO 原理图

这些数据知道一个引力波经过。这些测试物体的记录越远，激光器对小波动就越敏感。为了能探测到这些引力波，有必要建立一个到来的引力波的模型。因为在同一时刻有很多发生源，科学家们需要建立一个引力波模型让他们知道在一大堆数据中找什么。现在有一些地基的在建或者运行的探测器，包括 LIGO(USA)、

VIRGO(Italy/France)和 TAMA(Japan)，空间探测器有 LISA（2011）。

LIGO 的全称是 Laser Interferometer Gravitational Wave Observatory（激光干涉仪引力波探测器）。探测器采用迈克尔逊干涉仪的原理，主要部分是相互垂直的超高真空长臂，每个臂长 4km，末端悬挂反射镜。激光在管道中反射大约 50 次，使得等效臂长大大提升。引力波会引发光程差变化，从而导致干涉条纹变动。见图 1.2[22]。

LIGO 目前在美国华盛顿州 Hanford 天文台和路易斯安那州的 Livingston 天文台都有探测器，这两个地方相距 3002 公里。因为引力波的传播速度是光速，这个距离对于引力波到达的时间差只有 10ms。通过使用三角测量，到达的时间差可以判断在空中的波源。在一个真空系统中一共可以安装 5 个干涉仪。在 Hanford 大学第二个干涉仪和主干涉仪并行运转。这个辅助干涉仪只有主干涉仪长度的一半，也就是 2km,另外它的佩若手臂腔有相同的光学技术和一半的存储时间。从而理论上的应变领灵敏度和满长度的干涉仪在高于 200Hz 的频段是一样好的而在低频段则会效果减半。在 Livingston 大学则有一台主配置的干涉仪，这台干涉仪在 2004 年成功使用基于液压执行装置的主动隔振系统升级，这在 0.1-5Hz 频段提供了 10 倍隔绝效果，而地震在这个频段是主要的因为地震波和人为源。每个主干涉仪都在 L 形角上悬挂许多镜子，一个预先稳定的激光器发射 200W 的光束在到达光束分离器之前需要经过光模式筛选器。

基于当前天文事件的模型和广义相对论的预言，起源于千万光年之外的引力波应该会导致 4km 反射镜间距变换 10^{-18}m，比质子直径的千分之一还少。换句话说导致 10^{21} 分之一的距离变换。引发这些变换的事件一般有晚期的内旋态，两个 10 倍太阳质量的黑洞碰撞。LIGO 自 2002 年开始到 2010 结束，一共进行了 6 次科学运行。在 2004 年的第四次科学运行中，LIGO 探测器通过设计演示了提高用于测量这些距离变换的敏感程度的一倍。在 2005 年也就是第五次 LIGO 科学运行期间，敏感程度在 100Hz 宽的频段达到了 10^{21} 分之一的最初设计规范。两个大约太阳质量的中子星的内旋态的基线如果在 26,000,000 ly 之外应该是可以被观测到的，或者是近邻星系群对所有方向和偏振是平均的。在当时，LIGO 和 GEO600（一个德国和英国合作的探测器）进行了一个联合运行，在此期间他们收集了数个月的数据。Virgo（一个法国和意大利合作的探测器）在 5 月份加入。第五次科学运行在 2007 年结束。我们希望在进行密集的数据分析之后，来自这次科学运行的数据能够揭示两个明确的探测事件，这将会是物理学上的一个里程碑。在 2007 年 2 月，GRB 070201（一个 gamma 射线爆发）从邻近的仙女座星系方向到达地

球。普遍对于短暂的 gamma 射线爆发的解释是两个中子星或者是一个中子星和一个黑洞合并造成的。LIGO 报告一个非探测的结果，排除了在仙女座距离范围有合并现象。这样一个约束最终得以被 LIGO 断言，展示了对引力波的直接测量。在 2010 年 8 月，通过对来自射电望远镜的数据分析，一个射电脉冲星被发现了。尽管这很有意义，它的发现并不意味着引力波的侦测，但是也使用了 LIGO 的干涉仪。在第五次运行之后，原始的 LIGO 将升级为 Advanced LIGO,它的目标是将分辨率提高 1 倍，除了有美国国家科学基金会的最初的敏感度目标和更加先进的设备，还有一下改进：

1) 增加激光的功率
2) 零差检测
3) 输出模式筛选
4) 真空读数硬件

第六次科学运行（S6）从 2009 年 7 月开始，在 4km 探测器上使用增加的配置，到 2010 年结束。本文实验数据采用的是最后也就是第六次采集的数据，共有 1250 个有用频道。

1.2 主成分分析(PCA)

主成分分析(PCA,见图 1.3[24])是在多元统计学中使用正交变换将相关的观测

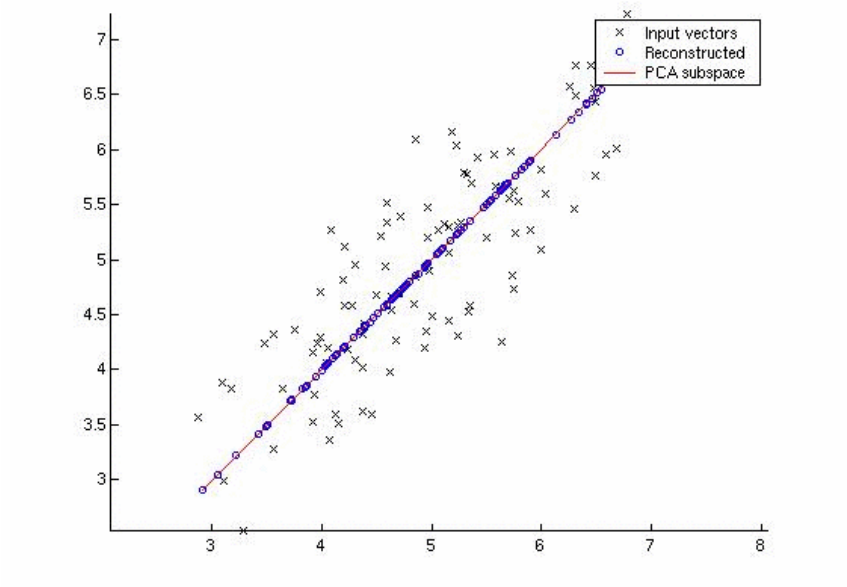


图 1.3 PCA 示意图

值转换为线性不相关的变量也就是主成分，以此用来简化分析数据集。主成分的个数毫无疑问和原来数据集相同或更少。第一个主成分有最大可能的方差，并且每个随后的成分在与前一个成分正交且也有最大的方差。只要数据集是联合正态分布，那么主成分就是相互独立的。PCA 对原始数据的归一化是敏感的。取决于应用的领域，它还有很多名称。在信号处理中，它叫做 Karhunen–Loève 变换（KLT），在机械工程中叫做本征正交分解(POD),数据集 X 的 SVD 分解，在线性代数中，X^TX 的特征值分解(EVD)等等。

Karl Pearson 在 1901 发明了 PCA，用于模拟力学主轴理论。之后被 Harold-Hotelling 在 1930 年独立开发研究。这种方法经常被用于探索性数据分析和建立预测模型。PCA 可以通过数据协方差矩阵的特征值分解或者是数据的 SVD 分解得到，通常数据集需要先进行正规化。PCA 的结果通常用每个成分的所占全部成分比重或者是叫做权重来衡量。

PCA 有很多变种，比如 KPCA，在 PCA 对于噪点(outliers)的敏感性，需要进行一些处理。对于 Kernel PCA 的方法而言，处理的不同是需要定义一个到高维的空间的一个映射。还有 RPCA，可以增强 PCA 的鲁棒性，减少噪点的干扰，实质上就是一个矩阵和分解的过程。

1.3 机器学习和 LIBSVM

首先需要了解机器学习的有关问题。二分类是我们经常讨论的问题，它在过去的一个世纪引发大量的理论研究和实践。LIGO 数据中的 0 和 1 也就是有无引

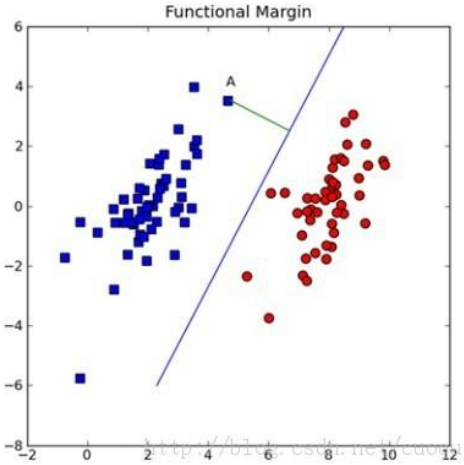


图 1.4　机器学习原理（二分类）

力波到来也是一个二分类（见图 1.4[25]）的判断。它所研究的对象是人工智能，

从数据到系统的重建和学习。1959 年 Arthur Samuel 定义机器学习是“给予计算机非事先编程而学习的能力的研究领域”。而对于一个学习者来说学习的核心是从经验归结出规律。机器学习算法可以通过想要得到的输出或者是输入的类型将其分类:

1）有监督学习

一般训练都是在有标签的数据集上进行的。

2）无监督学习

无标签的数据集上的学习。

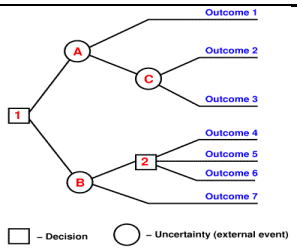
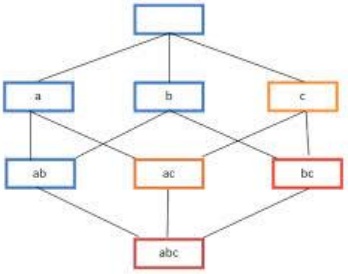
3）半监督学习

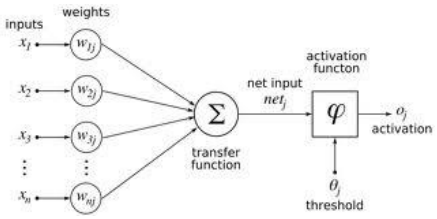
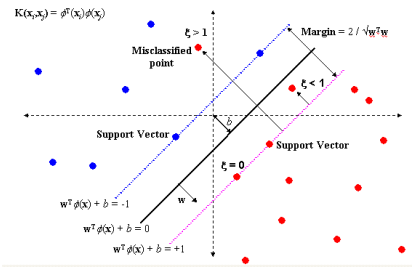
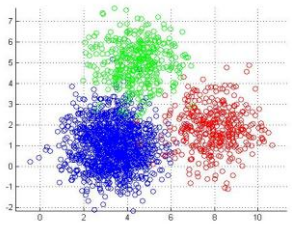
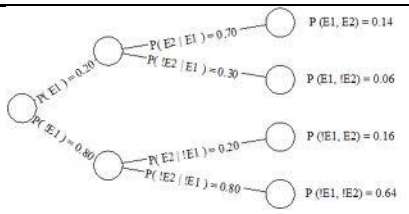
结合了前两种方法，同时带有标签和不带的训练学习方式。

4）强化学习方法

通过对环境反应分析，不改善学习的策略达到最大回报。所以又被称为再励学习。

表 1-1　各种机器学习方法和示意图

机器学习方法	简单描述	示意图
决策树学习	使用决策树来作为预测模型，将观测结果和特征进行映射。	
关联式学习	用于在大数据集中发现变量之间的关系。	

人工神经网络（ANN）	模仿生物学上的神经网络建立模型。能够根据外部信息改变自身结构，ANN 具有记忆性，输出依赖链接方式和激励函数。	
支持向量机（SVM）	是一种一般化的线性分类器，目标是找到最大间隔的超平面。	
聚类 (Clustering)	分为分散性和结构性聚类方法。经典的算法是 K-means。	
贝叶斯网络（信仰网络）	是一个概率图模型，其中的有向无环图的节点是随机变量。	

本文使用的是 LIBSVM，基于支持向量机的学习方式。LIBSVM 是台湾大学的 Lin Chih-Jen 副教授开发出来的一个 SVM 工具包，现在拥有很多的接口，包括 python，C#，Java，C，C++等等。

使用 LIBSVM 通用的步骤是：

1）准备 LIBSVM 需要的格式的数据，通常是 label 1:data … index:data index+1:data ….,工具中有相应的简易的检查输入数据格式的函数。对数据进行 scaling，也就是缩放，通常会使得学习过程更加准确，避免数量级差的过大。

2）选取 kernel function，有线性核，RBF 核，多项式核等。

3）进行 grid-search，选取最佳的参数，通常这个寻优的过程是非常耗时间的，所以需要利用 ligo 的网格机器进行计算。

4）对结果进行预测，画出 ROC 曲线并评估相应的 ligo-auc 指标。LIBSVM 工具包中关于训练和预测的函数：

Svm_train, Svm_predict, Svm_scale。

1.4 Condor 和网格计算

网格计算（见图 1.5[26]）是长时间数据分析的计算的一个非常实用的工具。网格计算的简单的想法是在互相连接的计算机中充分利用一些没占用的 cpu 执行周期。网格通常是由很多不同架构机器组成的，它们可以运行不同的操作系统。如果一个工作站标记为 IDLED，那么它的 cpu 将会被需要的程序利用。在网上众所周知的伯克利的 SETI@home 和斯坦福的 Folding@home 项目就是利用大量参与者的电脑的空闲 cpu 来执行海量数据分析的计算。很多运营商提供了商业化的并行计算的解决方案，然而我们可以使用 condor 这样的开源免费项目。它是由美国 Wisconsin 大学开发的，它背后的理念是非常简单的，在你希望成为那个堆的一部分的机器上安装 condor 就可以了。在 condor 术语中，一个 condor 堆也叫做池（pool）。你可以在任何机器上启动程序并且 condor 会在衡量空闲机器后匹配

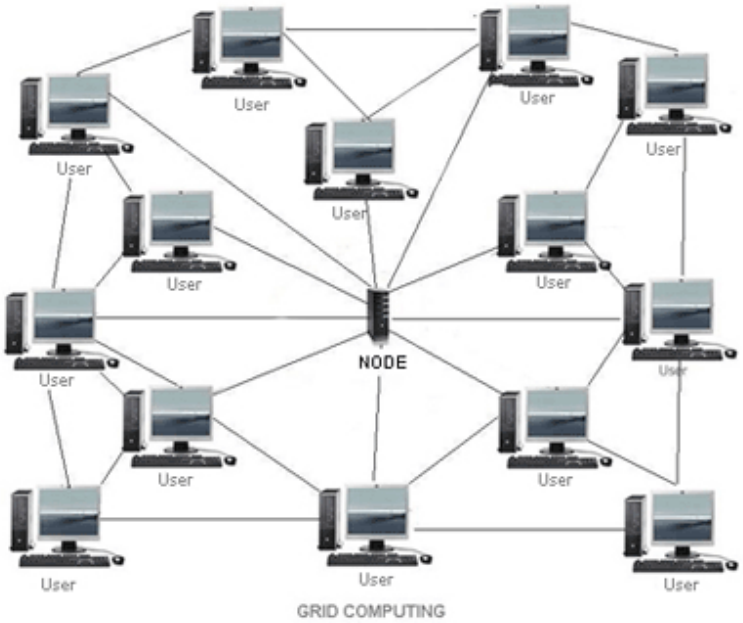


图 1.5　网格计算的示意图

工作的需求。Condor 的一个特征是它不需要程序在堆上进行修改。

当我们大多数人听到 condor 这个词语的时候，从来不会想到一个研究小组和他们周围发生的各种事情。相反，如果我们严格说是一个 Condor 小组开发出来的一个软件时，我们能够想到就是 the condor high throughput computing system。

Condor（见图 1.6[27]）是一个特别的工作和资源管理系统（RMS），对于计算密集型的任务来说，这是尤其重要的。和同类型系统一样，condor 有自己的任务管理机制，调度规则，优先方案，资源监视和管理。用户将任务提交给 condor，然后 condor 随后将选择任务何时何处基于准则运行，监视它的过程，最终告诉用户任务完成情况。condor 的目标就是高生产力的计算能力，在网络上优化资源的利用提供大量容错能力。Condor 的一些机制如下：

ClassAds：它为资源请求提供了一个极其具有弹性和表达性的框架，它允许 condor 采用任何满足资源利用规则的资源和合并资源的计划方法。

Job checkpoint 和 migration：对于某些特定的工作任务，condor 可以透明的记录检查点并可以将应用恢复到那个检查点。周期性的检查点可以提供一定容错能力和节省计算时间。一个检查点可以从一个机器移植到另外一个机器上，这使得 condor 能够运行低惩罚的抢占式恢复计划任务。

Remote system calls：当我们在远程机器上跑任务的时候，condor 通常会保存本地运行环境。远程调用就是 condor 重定向所有 IO 关联任务的移动沙盒机制。所以用户不必要使得数据文件在远程服务器上。

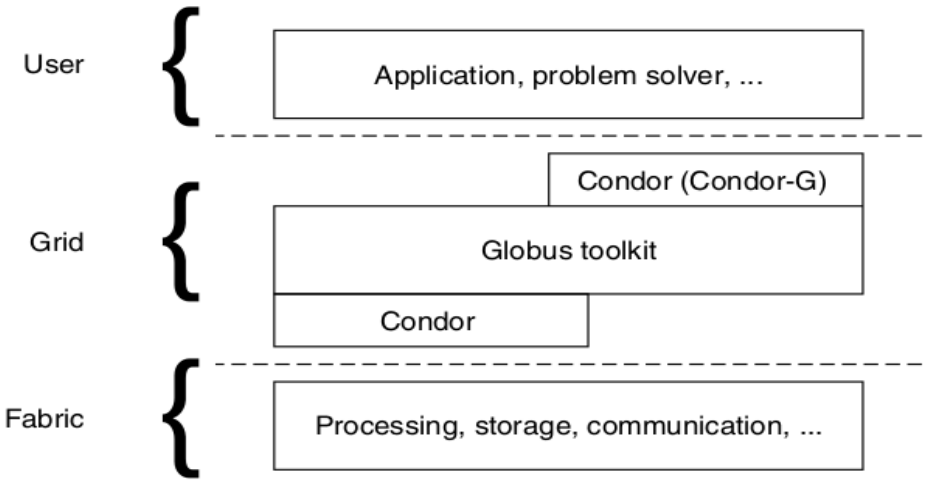


图 1.6　Condor 结构示意图

下面说下 condor-G，它是网格计算管理代理。也是 globus 和 condor 项目的一个合作成果。从 globus 中使用特定协议进行域内的安全访问并标准化对于不同的远程机器的访问。而 condor 中用户关心作业的提交和分配，错误查找和运行环境的创建。这个结果对于终端用户来说是非常好的，他们能够在大范围域内使用大量的资源，如果它们都属于用一个用户。

进程介绍：

condor_master: 这是持续运行的进程保证其他 condor 程序正常运行，如果有程序崩溃了，它就会重新启动崩溃程序。condor_collector:这个程序是 condor 中心管理进程的一部分。它收集所有在池中的电脑和哪些用户要运行作业的信息。

condor_negotiator:这个程序是 condor 中心管理进程的一部分。它决定哪些用户的工作在哪些机器上运行。

condor_startd:如果这个程序在运行，那么它允许作业在这台机器上运行。也就是这台机器是可运行的，这会告诉中心管理程序来注册自己的信息。

condor_schedd:如果这个程序在运行，它允许作业从这个电脑上提交，也就是这台机器是可提交的。

condor_shadow:对于每个作业从这台机器提交的作业而言，都有一个 shadow 进程在运行。它监视程序在远程工作的情况，并且有时候会提供一些辅助功能。

1.5 特征选择与 FSCORE

特征选择是另外一个可以作为数据压缩的手段。它也被称作子集选择或者是属性选择。主要的作用是剔除一些相关性差的特征，降低维数，从而能够大大减少机器学习训练的时间，同时也有助于数据简洁明了。

对于特定的方法，一般有评价函数，对数据进行简单的评价。特征选择算法一般情况有三种：

1) 指数算法

使用穷举算法，和名字相同，复杂度是 $O(e^n)$的，但是结果却是最优的，可以使用穷举搜索，分支界定，定向搜索等方法。

2) 随机算法

这种方法只能找到近似的最优解。在 N 巨大的时候，这个方法复杂度不高。如著名的遗传算法，退火算法.。

3) 序列算法

可能找不到全局最优解，复杂度不高，会陷入局部凹点。如前后向选择，双向选择方法。

F-Score 是根据矩阵的统计特征来最终确定每个特征之间的相关性，并不复杂，是线性复杂度的计算量。在 N 不大时计算速度很快。

在特征空间的全集中，我们找到一个特征子集，然后使用评价函数对该特征子集进行评估，评估的结果与停止标准进行比较，若评估结果比停止准则好就停止，否则就继续迭代生成下一个特征子集，继续进行特征选择。选出来的特征子集一般还要验证其有效性。[28]

经过上面的讨论，我们可以得到结论。特征选择过程包括 4 个部分，分别是评价函数，产生过程，验证过程和停止准则。

(1) 产生过程(Generation Procedure)

产生过程是搜索特征子集的过程，负责为评价函数提供特征子集。搜索特征子集的过程有多种。

(2) 评价函数(Evaluation Function)

评价函数是评估一个特征子集质量好与坏的一个函数。评价函数将在下面展开介绍。

(3) 验证过程(Validation Procedure)

在选择出来的特征数据子集上验证它的有效性。

(4) 停止准则(Stopping Criterion)

停止准则实质上就是一个阈值(threshhold),它是与评价函数相关的。当评价函数返回的值达到上限就停止搜索。

评价函数

评价函数的作用是评估产生过程所提供的特征子集的好坏，如下图[28]。

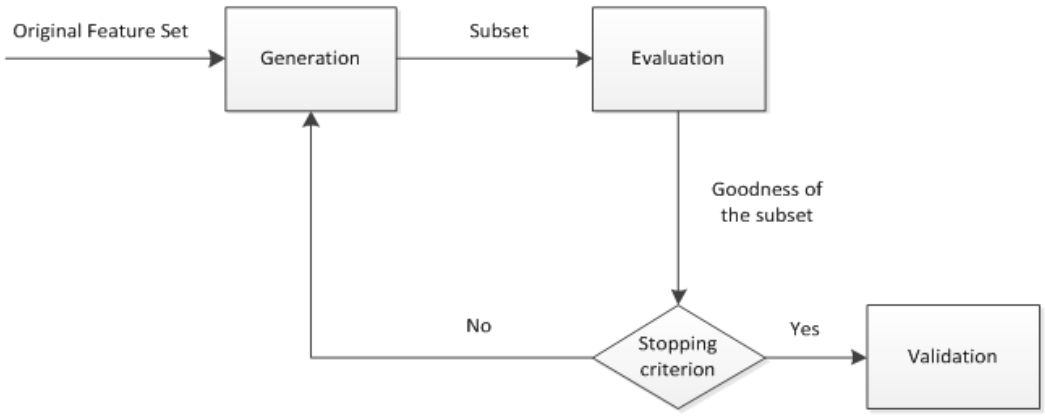


图 1.7　特征选择的过程（M. Dash and H. Liu 1997)

评价函数

根据其工作原理，主要分为筛选器(Filter)、封装器(Wrapper)两大类。

筛选器通过分析特征子集内部的特点来衡量其好坏。筛选器一般用作预处理，与分类器的选择无关。筛选器的原理如下图[28]。

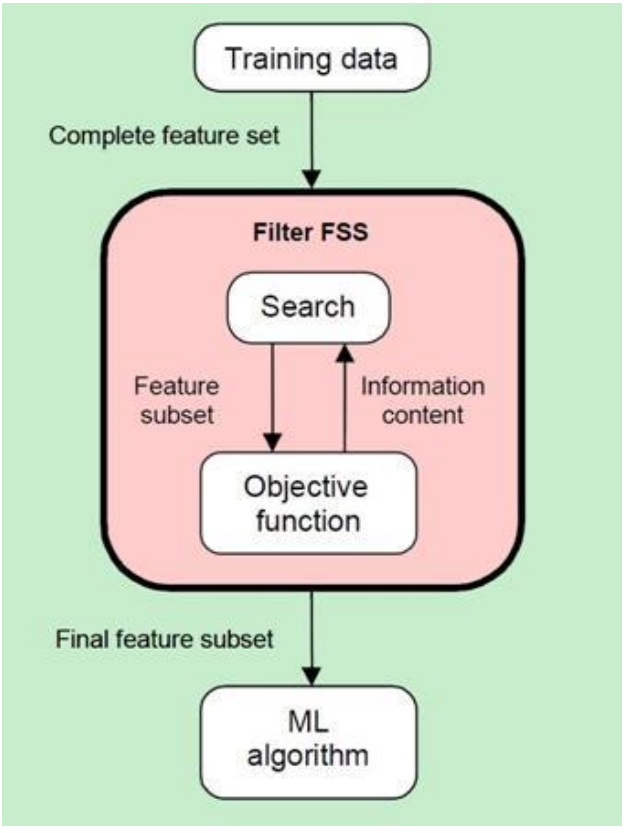


图 1.8　Filter 原理(Ricardo Gutierrez-Osuna 2008)

封装器实质上是一个分类器，封装器用选取的特征子集对样本集进行分类，分类的精度作为衡量特征子集好坏的标准。如下图[28]

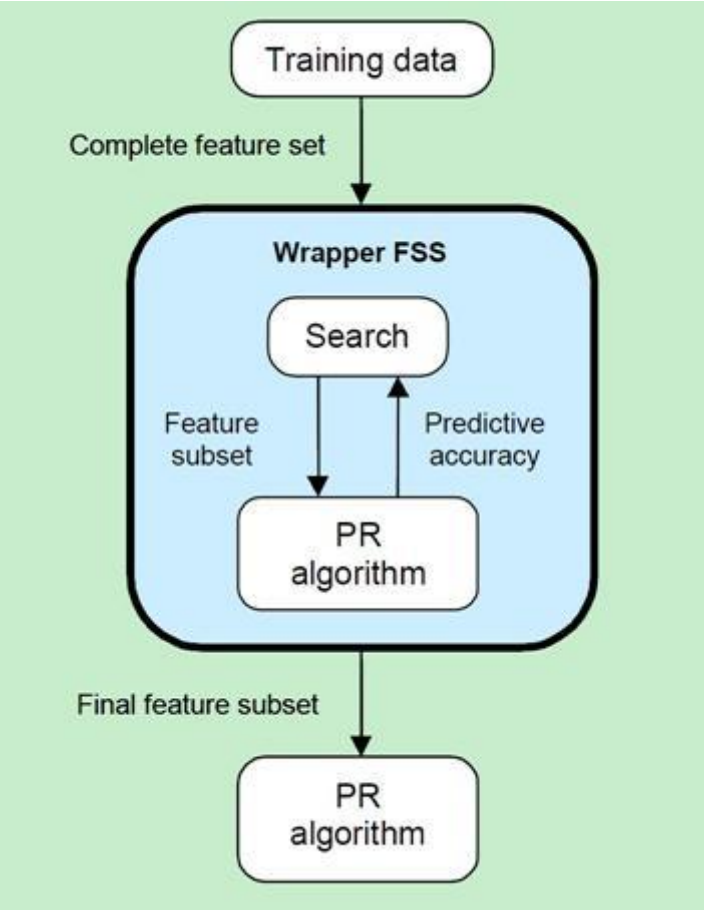


图 1.9　Wrapper 原理 (Ricardo Gutierrez-Osuna 2008)

下面简单介绍常见的评价函数。

(1) 相关性(Correlation)

运用相关性来度量特征子集的好坏是基于这样一个假设：好的特征子集所包含的特征应该是与分类的相关度较高（相关度高），而特征之间相关度较低的（亢余度低）。

可以使用线性相关系数(correlation coefficient) 来衡量向量之间线性相关度。

$$R(i) = \frac{cov(X_i, Y)}{\sqrt{var(X_i)var(Y)}}$$

(2) 距离 (Distance Metrics)

运用距离度量进行特征选择是基于这样的假设：好的特征子集应该使得属于同一类的样本距离尽可能小，属于不同类的样本之间的距离尽可能远。

常用的距离度量（相似性度量）包括欧氏距离、标准化欧氏距离、马氏距离等。

(3) 信息增益(Information Gain)

假设存在离散变量 Y，Y 中的取值包括{y_1，y_2，....，y_m} ，y_i 出现的概率为 Pi。则 Y 的信息熵定义为：

$$H(Y) = -\sum_{i=1}^{m} P_i \log_2 P_i$$

信息熵有如下特性：若集合 Y 的元素分布越“纯”，则其信息熵越小；若 Y 分布越“紊乱”，则其信息熵越大。在极端的情况下：若 Y 只能取一个值，即 P_1=1，则 H(Y)取最小值 0；反之若各种取值出现的概率都相等，即都是 1/m，则 H(Y)取最大值 $\log_2 m$。

在附加条件另一个变量 X，而且知道 X=x_i 后，Y 的条件信息熵(Conditional Entropy)表示为：

$$H(Y|X) = \sum_{i=1}^{m} P_{X=x_i} H(Y|X = x_i)$$

在加入条件 X 前后的 Y 的信息增益定义为

$$IG(Y|X) = H(Y) - H(Y|X)$$

类似的，分类标记 C 的信息熵 H(C)可表示为：

$$H(C) = -\sum_{i=1}^{m} P_i \log_2 P_i$$

将特征 Fj 用于分类后的分类 C 的条件信息熵 H(C | Fj)表示为：

$$H(C|F_j) = \sum_{i=1}^{m} P_{F=F_j} \, H(C|F = F_j)$$

选用特征 Fj 前后的 C 的信息熵的变化成为 C 的信息增益(Information Gain)，用：

$$IG(C|F_j) = H(C) - H(C|F_j)$$

表示。假设存在特征子集 A 和特征子集 B，分类变量为 C，若 IG(C|A) > IG(C|B) ，则认为选用特征子集 A 的分类结果比 B 好，因此倾向于选用特征子集 A。

(4)一致性(Consistency)

若两个样本是属于不同的分类的，但在特征 1、 2 上的取值完全一样，那么特征子集{1，2}就不应该被选作最终的特征子集。

(5)分类器错误率 (Classifier error rate)

使用特定的分类器，用给定的特征子集对样本集进行分类，用分类的精度来衡量特征子集的好坏。

以上几种度量方法中，距离、相关性、一致性、信息增益属于筛选器，而分类器错误率则属于封装器。

筛选器是和具体的分类算法无关的，因此它可以在不同的分类算法中可以被推广，而且计算量也不大。而封装器由于在评价的过程中应用了具体的分类算法进行分类，因此其推广到其他分类算法的效果可能性较小，而且计算量也比较的大。

1.6 研究目标

- 通过对 LIGO S6 的 10 组数据集（超过 10 万个样本）的分析掌握数据特性。
- 编写 pcatoolkit 模块实现代码重用，提升实验效率。
- 对数据进行主成分分析，分析压缩后的结果在 LIGO 网格计算机上运行 libsvm 之后的学习效果。

- 在原有的方法上，尝试不同方法的组合，做同样的分析，观察机器学习的效率和结果会有怎样的变化。
- 提出问题并进行改进。

第2章 研究现状与发展

2.1 引力波探测器

在 2000 年第一个锁定的 LIGO 探测器是 H2，然后 2002 年早期其他三个探测器也锁定了，经历了很多调试和科学数据获取的时间。在 2001 年中期到 2002 年中期调试过程提高了探测器的峰值灵敏度数个量级，例如 L1 在 150Hz 处提高了 10^{-17} 到 10^{-20}Hz。

利用架设在南极的 BICEP2 望远镜，美国哈佛-史密森天体物理学中心观测微波背景辐射——弥漫在宇宙空间中的微波背景光子形成的。物理学计算可以得知，一种 B 模式的偏振模式是由微波背景光子和原初引力波相互作用形成，而目前没发现其他的物理现象产生这个偏振模式，所以引力波探测和 B 模式是等价的。参考 BICEP2 2014 Release Papers[3]中的结果图片。

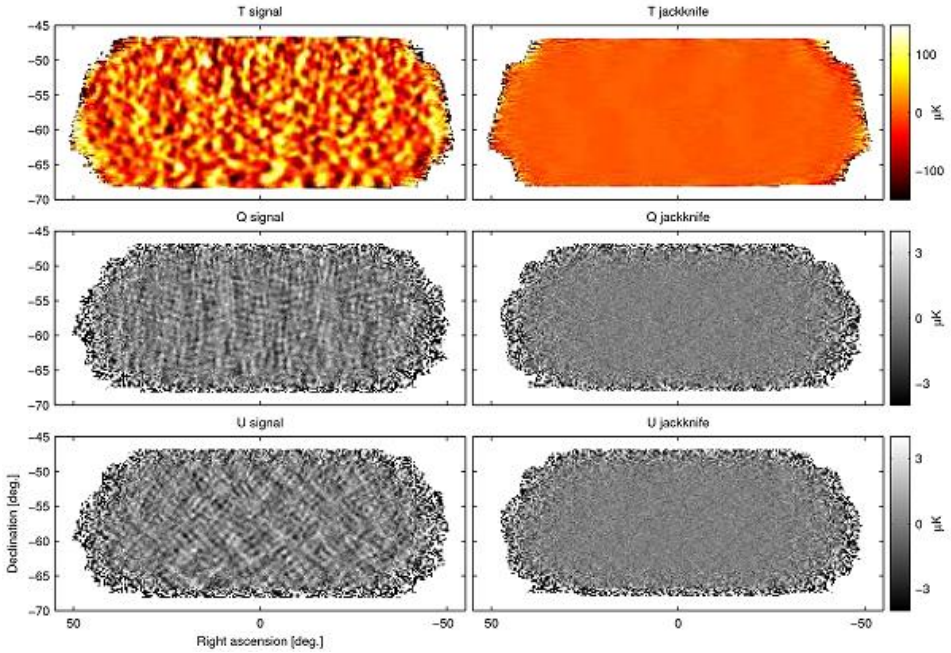


图 2.1 T/Q/U maps

左边显示通过压缩管道 0.25 °数位信号图。右边显示数据的一半和另一半之间的区别，没有额外的过滤除了强加了电子束(FWHM 0.5 °)，注意到 Q&U 信号图正如预期和 E 模式主导的天空一样。

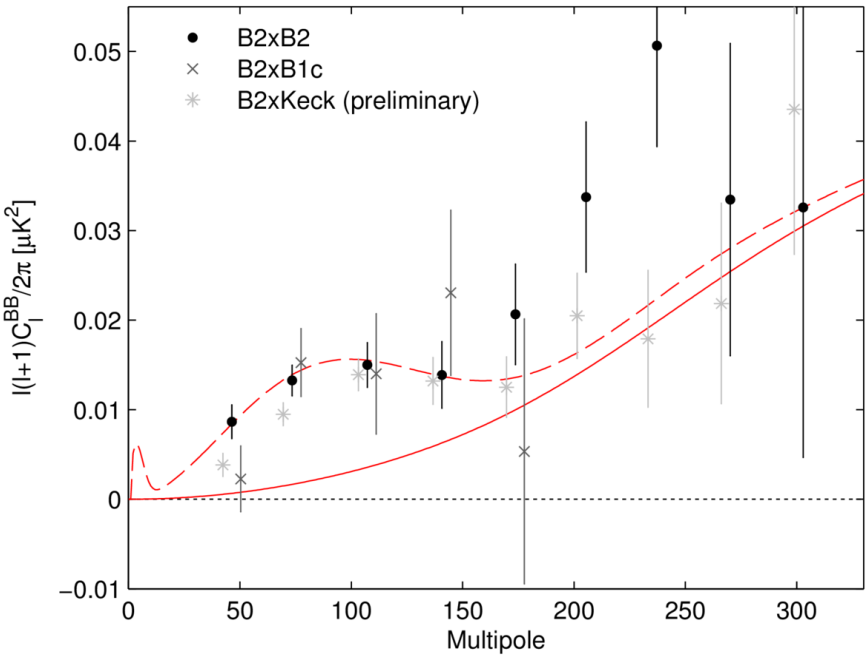


图 2.3　BICEP2 BB 光谱与 BICEP2 ,BICEP1 交叉光谱对比

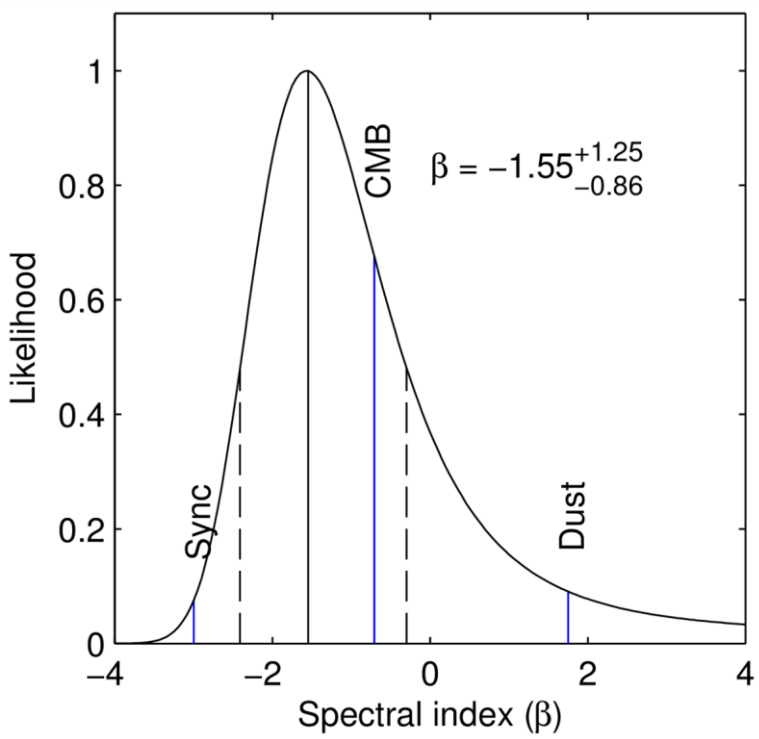


图 2.2　BB 信号指数约束

2.2 PCA 的应用

PCA 在多个领域都有很好的应用，并且有很好的前景。下面是人脸识别（如图 2.4[2]）的一些介绍，是图像处理的特例。

数据集就是图像矩阵，大小是 N*N，然后对它们进行 PCA 处理，找出主成分。通过人脸识别可以说明，PCA 就是找到主要的相似的地方，排除干扰得到骨架模型，新的图像通过模型在新的空间中进行对比，可以判断构造出来的矩阵与原始的数据集的一致性。在普适的 PCA 方法上加以修改，使用例如 ICA、KPCA、RPCA 等方法，可能得到更好的效果。 如果得到的一致性很高，则这个人脸的特征可以作为所有具有类似特征的人脸的代表，就是变换过程类似，换句话说使用相同的变换矩阵。

图 2.4　原始图片和重建模型进行训练

图像压缩使用 Hotelling 算法也称维 KL 变换。使用主成分分析来处理一个图像序列并提取图像矩阵的主成分，然后根据主成分特征大小的顺序去除其中不重要的分量，最后映射到原始空间，则图像会得到很大的压缩。

2.3 SVM 的发展

SVM 最经典的理论是在上个世纪 90 年代提出，它的整个的理论框架在实际应用中有着良好的效果。因此在机器学习领域它是极其重要和受重视的。

1) 最小二乘 SVM:这个方法已经非常古老了，已经十几年的经验累积。现在主要的关心点是大数据以及鲁棒性问题，还有参数选择仿真和调节。

2）模糊 SVM:这种方法提高了抗干扰的能力

3）有偏样本和风险加权 SVM。

4）主动学习与 SVM 结合:

5）粗糙集与 SVM 结合

6）决策树与 SVM 结合

7）多级聚类的 SVM

由于每个数据集本身的性质不同，并且应用的领域的差别，需要使用不同的核函数（kernel function）。核函数的类型有很多种，列出如下：多项式、多层感知器、径向基函数（RBF）以及贝叶斯分类器。选择参数方法是将数据分成若干份然后互相作为训练集和测试集来验证，又叫交叉验证。

从面的讨论我们可以得出目前 SVM 在如下几个方面是研究的焦点所在：构造核函数(kernel function) 并为之选择参数；SVM 除了在 2 分类上有应用，应该在多标签的情况也能应用；把 SVM 和很多的机器学习的方法和数据的预处理（比如本文使用 pca，加入监督学习部分）相结合，这本质上是数据本身的性质融入到算法中，从而可以得到更加适合数据的算法并最终得到较好的学习效果。

第3章 实验资料

3.1 实验环境

本文实验环境是 LIGO 网格计算机。先在本地机器上进行 PCA 模块编程，使用 git 进行版本管理，然后将最终版本传至 LIGO 的网格计算机中，使用 condor 进行任务控制，将核数控制在 200 以下，避免占用更多的计算资源而引起网络堵塞。下面介绍下 condor 和 condor-G 的使用说明。

condor_q：查询所有状态的进程。默认是所有用户的进程，如果需要查看特定用户的进程需要加上参数-submitter USERNAME。我们可以查看 ID，拥有者，提交的时间，运行时间，状态（运行还是闲置），大小以及命令的名称。

具体用法是：condor_q [-debug] [general options] [restriction list] [output options] [analyze options]

condor_rm：删除作业，可以指定 ID，也可以指定用户（-all 删除所有作业）。

具体用法是：condor_rm [-debug] [-pool centralmanagerhostname[:portnumber]] -name scheddname [-addr <a.b.c.d:port>] -all

condor_status：可以查看有哪些电脑在你的任务池中。

具体用法是：condor_status [-debug] [help options] [query options] [display options] [custom options] [name …]

由于安全因素，不能直接通过 ssh 连接至服务器，需要安装 LDG 代理和 Globus 管理程序。从 LDG Client 官网上下载源代码安装，然后运行代理 ligo-proxy-init USERNAME。输入密码之后登陆远程节点 gsissh pcdev1.phys.uwm.edu。demo 程序包括了用于网格搜索的 condor 提交程序(search.py)，结果评估的 evaluate.py 以及 ROC 生成程序 ROCmaker.py，配置文件 auxmvc_S6.ini。

网格搜索

命令：python search.py --config auxmvc_S6.ini

配置：网格范围[grid_search]，数据文件[SVM_file]

结果评估

命令：python evaluate.py –config auxmvc_S6.ini

配置：网格范围[grid_search]，结果文件和评测标准[evaluate]

ROC 曲线生成

命令：python ROCmakrer.py –config auxmvc_S6.ini –result rst.png result.file

配置：ROC 图[auxmvc_roc_result]

文件传输方式

将文件传到远程服务器上:

globus-url-copyfile:/path/to/data gsiftp://pcdev1.phys.uwm.edu/home/USERNAME/path/to/destination

将文件从远程传回来:

scp user@ip:/path/to/data ~/path/to/destination

3.2 PCA 算法具体描述和程序模块的使用

本文使用的是SVD分解方法获得变换矩阵W和变换后的数据，并且保留PCA计算后的模型文件，避免重复计算 PCA。对应的流程图，首先是准备数据集。其次是对原始数据集进行规范化处理，这是为了保证量纲一致，从而避免特征值的数量级过大而夸大某些主成分的作用。之后对处理后的数据集进行 SVD 运算，

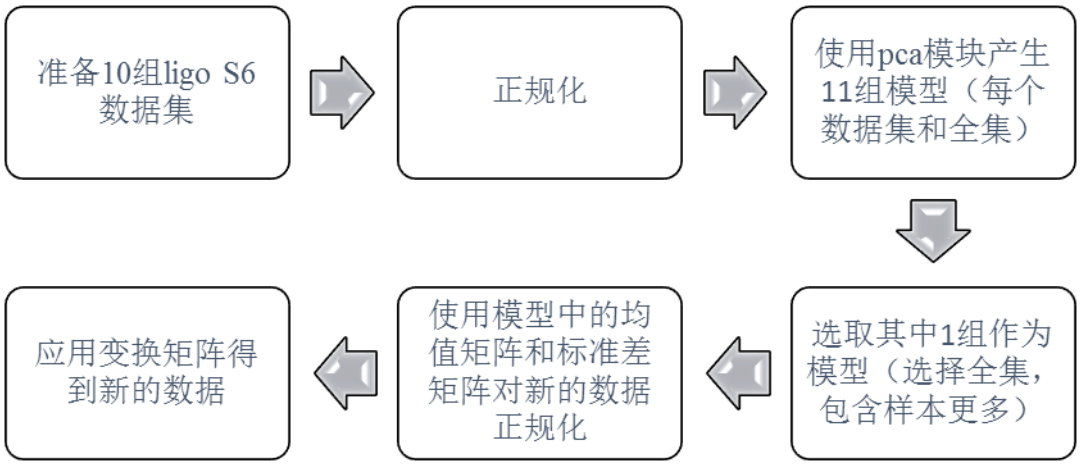


图 3.1 PCA 模型流程图

得到变换矩阵和每个特征的统计量，包括均值和标准差，以及全 0 特征列的标识。在配置文件 default.cfg 中配置一个默认使用的 pca 模型，然后对需要进行线性投影的数据进行规范化，用之前在模型文件中存储的统计量来计算。然后用变换矩阵进行线性变换，就能得到新的 10 组数据，最后将数据链接到训练集上进行机器学习。

PCA 是一个统计学上的技术。它被用来分析大型数据内部变量之间的关系并且将这些变量解释为更少的变量，也就是主成分。

模型文件结构

[行数 列数]： 校验待变换矩阵的行数列数

[均值矩阵]： 用于中心化

[标准差矩阵]： 用于规范化

[0 列矩阵表示]： 标识 0 列矩阵

[变换矩阵]： 将原始矩阵映射到新的空间中的矩阵

实际 PCA 变换公式是：

$$\mathbf{T} = \mathbf{X}\mathbf{W}$$

W 是负载矩阵，也就是权重矩阵。X 是待变换的矩阵，T 变换后的矩阵。

协方差矩阵 Q 可以被分解为对角矩阵：

$$\mathbf{W}^T\mathbf{Q}\mathbf{W} \propto \mathbf{W}^T\mathbf{W}\,\mathbf{\Lambda}\,\mathbf{W}^T\mathbf{W} = \mathbf{\Lambda}\mathbf{Q} \propto \mathbf{X}^T\mathbf{X} = \mathbf{W}\mathbf{\Lambda}\mathbf{W}^T$$

而 Q 满足：

所以这个对角矩阵是 X^TX 的特征值。我们可以通过对特征值和对应特征向量排序得到我们需要的结果。另外一种方法是使用 SVD 分解来求变换矩阵和特征值:$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{W}^T$

$$\begin{aligned}\mathbf{X}^T\mathbf{X} &= \mathbf{W}\mathbf{\Sigma}\mathbf{U}^T\mathbf{U}\mathbf{\Sigma}\mathbf{W}^T \\ &= \mathbf{W}\mathbf{\Sigma}^2\mathbf{W}^T\end{aligned}$$

中间的对角阵∑是 m*m 的矩阵，叫做奇异值矩阵。U 和 W 是正交单位阵。在这种形式下分解，X^TX 可以被写成：

可以看出得到的奇异值是特征值的平方根。使用奇异值分解变换公式可以写成如下形式：

$$\begin{aligned}\mathbf{T} &= \mathbf{XW} \\ &= \mathbf{U\Sigma W}^T\mathbf{W} \\ &= \mathbf{U\Sigma}\end{aligned}$$

本文使用的是 python 的 numpy 库中的 svd 函数，通过得到变换矩阵 W 和均值方差等信息，得到 PCA 的 2 进制模型，按照读写顺序依次是矩阵行数，列数，均值矩阵，标准差矩阵，0 列标识矩阵以及正交变换矩阵。为了代码可重用，编写了用于 ligo 数据的 pcatoolkit 包，其中包括 pca 算法类，读取原始数据类，格式转换类。见图 3.2，每个分支左边是成员函数，右边是成员。第一个是 pca 算法类，我们首先用一个矩阵初始化这个类，然后使用 savemodel(filename)来保存模型信

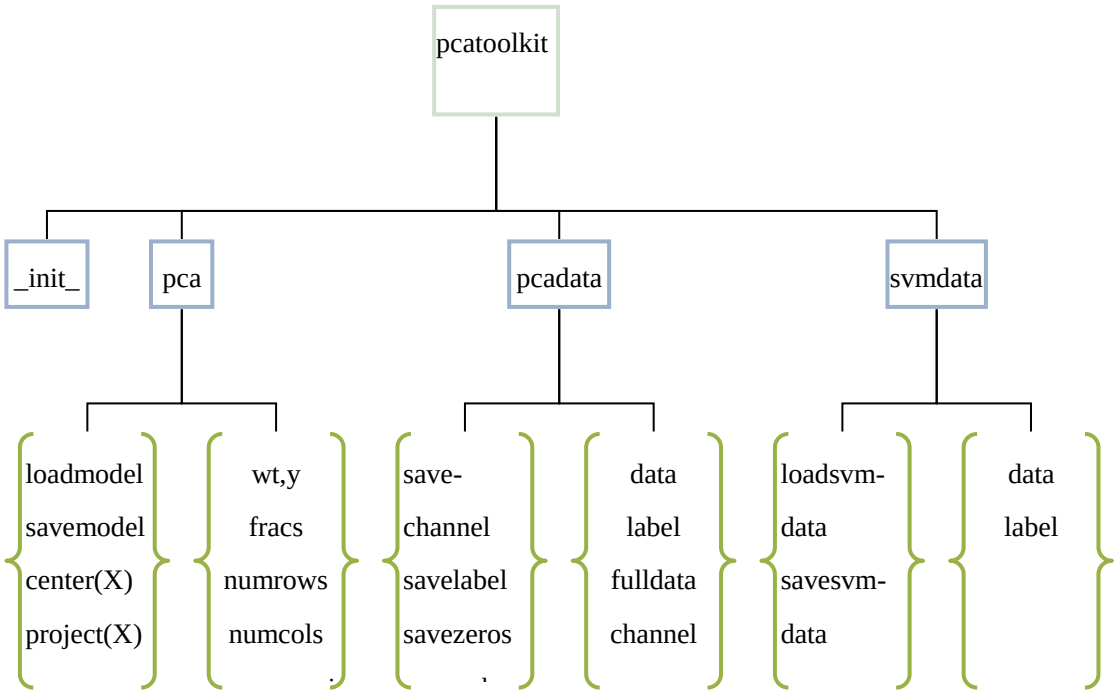


图 3.2　pcatoolkit 模块图

息。这个过程已经完成正规化。然后当我们需要使用模型时，不用再载入原始矩阵，只需要 loadmodel(filename)，然后对数据进行中心化，这个过程在 project 函数中已经完成，所以对于新来的数据 X，我们只需要 project(X)就可以得到变换后的矩阵。Wt 是变换矩阵，y 是原始数据的变换后的矩阵，numrows 和 numcols 是行数和列数，mu 是均值，sigma 是标准差，frac 是原始矩阵各个主成分的百分比。Pcadata 是对原始 ligo 数据进行分析，得出一系列性质，包括提取全 0 列，

频道的 ID，分离数据和标签，去掉不要的信息等等。Svmdata 是格式转换的类，最终我们提交给 ligo 机器的数据是 libsvm 的标准格式：indexofsample index:data。这我们只需要把 project 后的矩阵传递给 svmdata 的构造函数，然后运行 savesvmdata 就完成了。

3.3 ROC 曲线

全名是 Reciever Operating Characteristic Curve。它显示了阈值变化的二分类系统的性能。分类模型 （分类器或诊断） 是实例某类/组之间的映射。分类器或诊断结果可以是一个真正有价值 （连续输出），在这种情况下的分类器边界之间的类必须由一个门限值 （例如，要确定一个人是否有高血压基于血压测量）。或者它可以是一个离散的类标签，指示类之一。

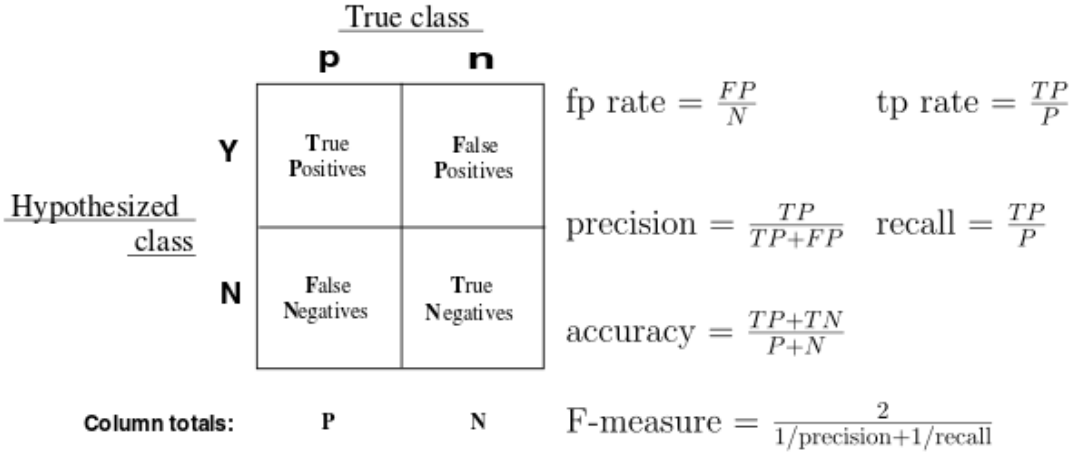


图 3.3 ROC 概念表

这里需要介绍几个概念，都是画 ROC 曲线必须的变量。

1) 阳性 (P, positive) 阴性 (N, Negative)。
2) 真阳性 (TP, true positive) 正确的肯定，又叫做命中 (hit)
3) 真阴性 (TN, true negative) 正确的否定，又叫做正确拒绝 (correct rejection)
4) 伪阳性 (FP, false positive) 错误的肯定，又称：假警报 (false alarm)。
5) 伪阴性 (FN, false negative) 错误的否定，又称：未命中 (miss)。
6) 真阳性率 (TPR, true positive rate) 又叫做命中率 (hit rate)
   TPR = TP / P = TP / (TP+FN)

7) 伪阳性率(FPR, false positive rate) 又叫做错误命中率，假警报率 (false alarm rate)　FPR = FP / N = FP / (FP + TN)

8) 准确度 (ACC, accuracy)

ACC = (TP + TN) / (P + N)　(真阳性+真阴性) / 总样本数

9) 真阴性率 (TNR) 又叫做特异度 (SPC, specificity)

公式为 SPC = TN / N = TN / (FP + TN) = 1 – FPR。

10) 阳性预测值 (PPV)

PPV = TP / (TP + FP)。

11) 阴性预测值 (NPV)

NPV = TN / (TN + FN)。

12) 假发现率 (FDR)

FDR = FP / (FP + TP)。

13) Matthews 相关系数 (MCC)，即 Phi 相关系数

MCC = (TP*TN - FP*FN) / \sqrt{P N P' N'}。

14) F1 SCORE

F1 = 2TP/(P+P')。

ROC 曲线是一个 2 维曲线图，Y 轴是 TP 率，X 轴是 FP 率。ROC 曲线描绘了相对之间的权衡（真阳性）的好处和成本（假阳性）。下图从 A 到 E 的 5 个分类器显示了简单的 ROC 图。

一个离散的分类器输出只有一类的标签。每个离散的分类器产生（TP 率，FP 率）对。对应着 ROC 曲线图中的每一个点。

图中几个点需要注意一下，一个是左下角的（0，0）点，它表示策略从来没有发出一个积极的分类；这种分类器既没有假阳性错误但也没有真阳性。而右上角的（1，1）点则是代表了无条件的发出积极的分类的策略。点（0，1）代表完美策略。

一般地说，在 ROC 曲线中，一个点如果更加靠近左上角（也就是 TP 率更高，FP 率更低），那么这个点就越好。

然后是 AUC 指标，指的是曲线下方的面积。它的范围是 0-1，从 AUC 可以看出曲线的优劣。

AUC=1，是完美分类器

0.5<AUC<1，是比随机猜测的分类器要好，也就是大于一半一半的情况。

0<AUC<0.5，是比随机猜测的分类器要差，也就是小于一半一半的情况。

AUC=0.5，相当于随机猜，一半一半概率。

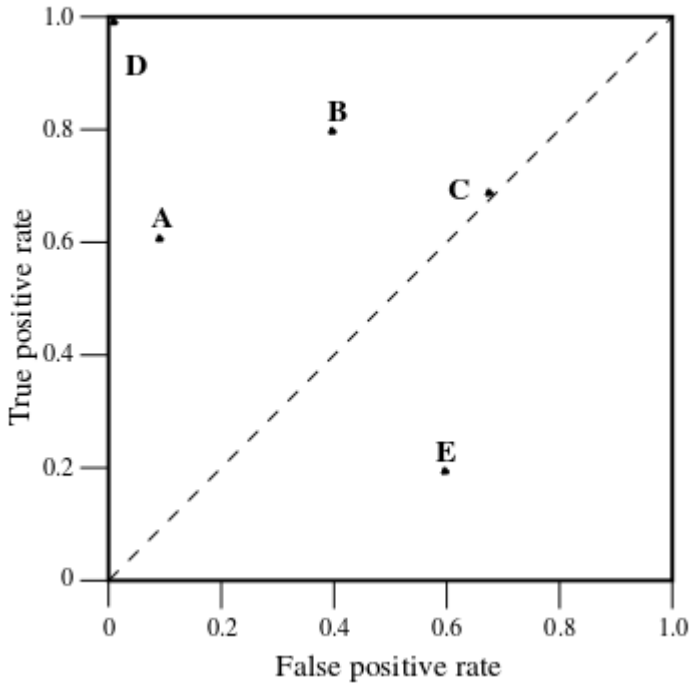


图 3.4　简单的 ROC 图

本文的 ROC 曲线的 AUC 和 LIGO-AUC 指标都是通过梯形法计算的，也就是近似求和。LIGO-AUC 是计算 0.001-0.01 之间的面积。

3.4 Numpy 的使用

NumPy 是 Python 编程语言用来增加对大型、 多维数组和矩阵，以及高级别数学函数来操作这些阵列的一个大扩展库。NumPy 的祖先，Numeric 的最初是由 Jim Hugunin 来自其他几个开发商的捐助开发的。2005 年，Travis Oliphant 将 Numarray 特征纳入 Numeric 并进行大量的修改得到了 Numpy。NumPy 是开源的且有很多的贡献者。

NumPy 的目标是一个非优化字节码编译器/解释器 CPython 引用的 python 执行。和经常为此版本的 Python 编写的数学算法运行已编译的等价物比慢得多。NumPy 旨在通过提供多维数组和函数和运算符，运作有效的阵列上解决这一问

题。因此作为等效的 C 代码任何可以表达主要是作为数组和矩阵运算的算法几乎可以跑得一样快。

在 Python 给使用 NumPy MATLAB 既然他们都相媲美的功能解释，和他们俩都允许用户写快的程序，只要数组或矩阵而不是标量的大多数操作工作。相比较而言，MATLAB 拥有大量的额外的工具箱，值得注意的是 Simulink；而 NumPy 在本质上是与 Python 是一种集成更多的现代、 齐全，并打开源的编程语言。此外互补 Python 包也可用；SciPy 是一个库，添加更多的类似 MATLAB 的功能和 Matplotlib 是一个绘图软件包，提供了类似 MATLAB 的绘图功能。在内部，MATLAB 和 NumPy 依靠 BLAS 和 LAPACK 的高效线性代数计算

NumPy 的核心功能是其"ndarray"，为 n 维数组的数据结构。这些数组是对内存的大尺度的观测。和 Python 的内置列表数据结构 （即不管名字，是一个动态数组） 的对比，这些数组都是均匀类型： 单个数组的所有元素必须都是同一类型的。

这类数组也可以是到 C，c++所分配的内存缓冲区的观察。Cython 和 Fortran 语言的扩展，CPython 解释器，而不需要复制数据周围，给予一定程度的与现有的数值库的兼容性。此功能被 SciPy 包利用，包装了大量的这类库（尤其是 BLAS 和 LAPACK）。NumPy 具有内置支持的内存映射 ndarrays。

```
>>> import numpy as np
>>> x = np.array([1, 2, 3])
>>> x
array([1, 2, 3])
>>> y = np.arange(10)  # like Python's range, but returns an array
>>> y
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

图 3.5 numpy 创建数组

```
>>> a = np.linspace(-np.pi, np.pi, 100)
>>> b = np.sin(a)
>>> c = np.cos(a)
```

图 3.6 numpy 基本操作

```
>>> a = np.array([1,2,3,6])
>>> b = np.linspace(0,2,4)  # create an array with end-points 0 and
>>> c = a - b
>>> c
array([ 1.        ,  1.33333333,  1.66666667,  4.        ])
>>> a**2
array([ 1,  4,  9, 36])
```

图 3.7　numpy 一元函数

NumPy 基于两个较早的 Python 数组包。第一个是所谓的 Numeric，Python 数值扩展或 NumPy，这是一个完整和稳定的库，仍然可用，但现在已经过时了。它最初是在 1995 年，很大程度上由吉姆 Hugunin 创建，然后在麻省理工学院的加入 CNRI JPython 工作的研究生 Hugunin，LLNL 的 Paul Dubois 作为维护者接管了。其他早期的参与者包括 David Ascher，Konrad Hinsen 和 Travis Oliphant。

```
>>> from numpy.random import rand
>>> from numpy.linalg import solve, inv
>>> a = np.array([[1, 2, 3], [3, 4, 6.7], [5, 9.0, 5]])
>>> a.transpose()
array([[ 1. ,  3. ,  5. ],
       [ 2. ,  4. ,  9. ],
       [ 3. ,  6.7,  5. ]])
>>> inv(a)
array([[-2.27683616,  0.96045198,  0.07909605],
       [ 1.04519774, -0.56497175,  0.1299435 ],
       [ 0.39548023,  0.05649718, -0.11299435]])
>>> b =  array([3, 2, 1])
>>> solve(a, b)  # solve the equation ax = b
array([-4.83050847,  2.13559322,  1.18644068])
>>> c = rand(3, 3)  # create a 3x3 random matrix
>>> c
array([[ 3.98732789,  2.47702609,  4.71167924],
       [ 9.24410671,  5.5240412 , 10.6468792 ],
       [10.38136661,  8.44968437, 15.17639591]])
>>> np.dot(a, c)  # matrix multiplication
array([[ 3.98732789,  2.47702609,  4.71167924],
       [ 9.24410671,  5.5240412 , 10.6468792 ],
       [10.38136661,  8.44968437, 15.17639591]])
```

图 3.8　numpy 线性代数的应用

3.5 Plotly beta

Plotly 是一个基于网络浏览器的即时画图工具。它可以使用网页上的按钮也可以使用它的 API 来操作位于远程服务器上的数据。

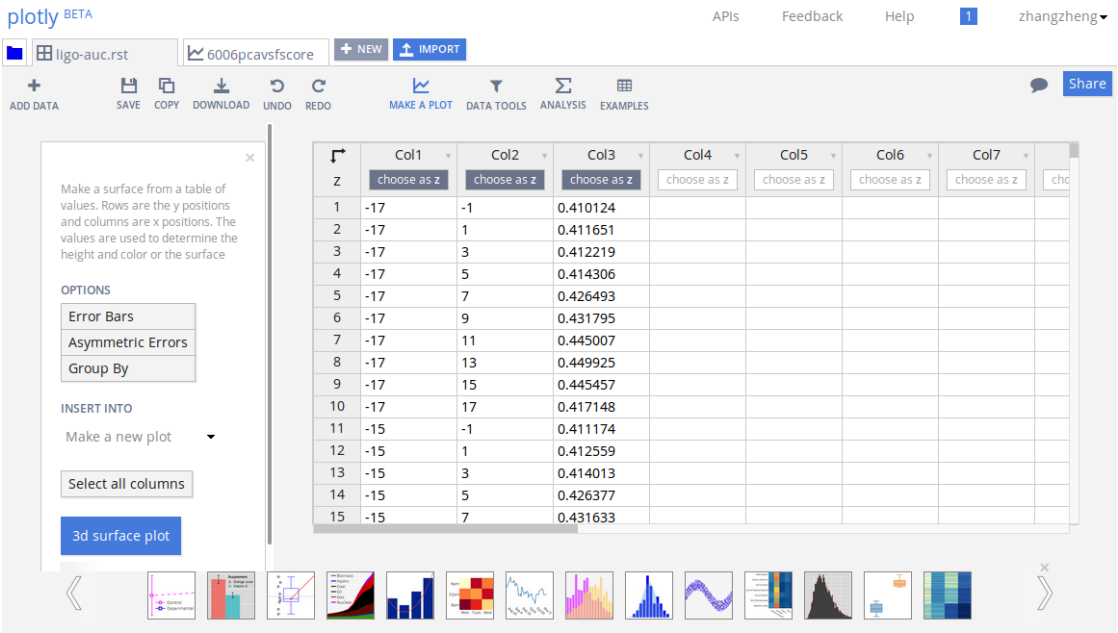


图 3.9 plotly 网页接口

第五章的搜索图就是通过 plotly 画出来的。我们可以使用它的 API 来操作，在 github 上有很多基于官方 API 实现的 python 接口的程序，例如 arduino-api,raspberrypi。生成图片时你可以选择这些图片是否是公开的，这取决于参数。按照默认习惯，每个人都可以通过唯一的 URL 来访问自己的图片，如果需要设成私有的，那么将变量 world_readable 设为 false。

```
plotly graph = plotly("your_plotly_username", "your_plotly_api_key",
streaming_tokens, "your_plotly_filename", num_traces);
graph.world_readable = false;
```

时间戳默认 X 是毫秒级的，并且在服务器上自动将 X 以"America/Montreal"为基准转换为实时的时间戳。为了取消默认行为，需要如下操作。

```
plotly graph = plotly("your_plotly_username", "your_plotly_api_key",
streaming_tokens, "your_plotly_filename", num_traces);
```

```
void setup(){
  graph.convertTimestamp = false;
}
```

为了改变时区应该设置 timezone

```
void setup(){
  graph.timezone = "Africa/Abidjan";
}
```

为了改变画图时最大的点的数量

```
void setup(){
  graph.maxpoints = 200;
}
```

默认你初始化图片（`graph.init();`）的时候，会覆盖你原来存在的图片。这对

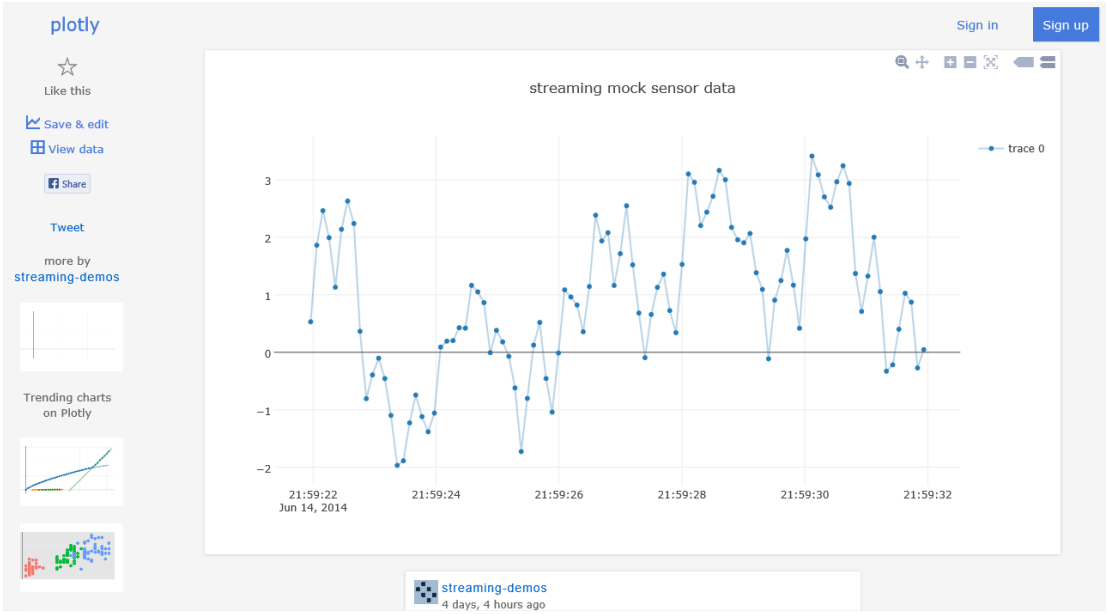


图 3.10 Arduino stream 演示图

于开发而言无疑是一个好的选项。但是假设你长时间运行，那么你会发现 Arduino 会复位自己并且移除所有的数据。为了防止这种事情发生，你可以使用 `fileopt "extend"`，这将会将你的新的数据附加在已经存在的数据上而不是移除或者是覆盖。因此你需要在 `setup(loop)`中添加 `graph.fileopt = extend`。

```
void setup() {
  Serial.begin(9600);

  startEthernet();

  bool success;
```

```
    graph.maxpoints = 500;
    graph.fileopt = "extend"; // Remove this if you want the graph to be
overwritten
    success = graph.init();
    if(!success){while(true){}}
    graph.openStream();
  }
```

参数 log_level 设置调试信息的打印方式。对于寻找 bug 时需要设置为 0。如果不想打印调试信息那么设置成 4。

```
  void setup(){
    Serial.begin(9600);
    startEthernet();

    graph.log_level = 0;

    success = graph.init();
    if(!success){while(true){}}
    graph.openStream();
  }
```

```
class plotly(char *username, char *api_key, char* stream_tokens[], char
*filename, int nTraces);
```

Public Member Functions（公共函数）

- `bool plotly.init()`

在你的 plotly 账户中建立一个空的图。这通过对 plotly 的 REST 服务的 API 调用实现的。如果初始化成功那么返回 `true`，否则返回 `false`。

- `void plotly.openStream()`

打开一个 plotly 的流式服务的 TCP 连接。这个流由 `stream_tokens` 唯一标识。

- `void plotly.closeStream()`

关闭 plotly 的流连接。

- `void plotly.reconnectStream()`

重新打开 plotly 的流式连接，如果连接没打开的话。

- `void plot(unsigned long x, int y, char *token)`

将(x, y)画在流图中

- `void plot(unsigned long x, float y, char *token)`

将(x, y)画在流图中

Public Member Parameters

- `int plotly.log_level` (默认 `2`)

  决定在序列中调试信息的打印方式:

  - `0`: Debugging
  - `1`: Informational
  - `2`: Status
  - `3`: Errors
  - `4`: Quiet
- `bool plotly.dry_run`

如果 `True`，那么将对 Plotly 的服务器没有调用。

- `int plotly.maxpoints` (默认 `30`)

决定一次最大打印点的数量，从 1 到 200000。

- `bool plotly.convertTimestamp` (默认 `true`)

如果 `true`，那么 Plotly 假设 `x` 是毫秒级的，因此程序开始时执行 (`millis()`) 并且自动转换时间戳。

- `char *plotly.timeZone` (默认 `"America/Montreal"`)

如果 `plotly.convertTimestamp=true`，那么将转化时区。

- `bool plotly.world_readable` (默认 `true`)

如果 true，那么你的图片将是公开的并且可以被唯一的 URL 对应发现，否则只有你个人能看到。

- char *plotly.fileopt (默认 "overwrite")

  "extend" 或者"overwrite"。

如果"overwrite"，那么当图片初始化的时候(在 plotly.init()时) ，已经存在的图片将会被新的覆盖。这意味这原来的图片会被移除。这个选项对于开发应用而言是一个不错的选择，因为你每次运行脚本总是想要刷新图片。

如果 "extend"，那么那么当图片初始化的时候(在 plotly.init()时) ，已经存在的图片将会保留。并且新的数据将会加在存在的数据后面。这个选项对于当需要长时间运行设备的时候是有好处的。因为 Arduino 会复位，通常可能是每几个小时，那么已经存在的数据就不会被移除，如果使用这个选项的话。

第4章 实验内容

4.1 对 LIGO S6 10 组数据的 pca 特性分析

每个数据集都有全 0 的列，这对机器学习而言这些特征是没有贡献的，为此

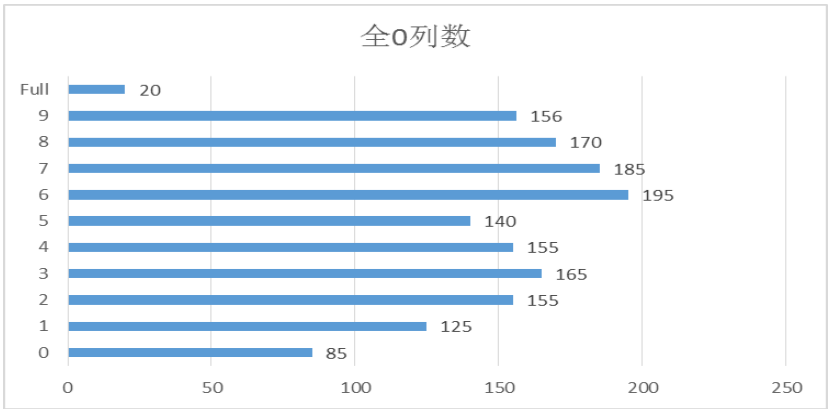


图 4.2 全零列在数据集中锋分布

在进行训练之前，先将这些数据的全 0 特征的 ID 提取出来，观察它们的数量和布。可以看到对于 10 个数据集一起作为 pca 模型的模型只需要去掉 20 个特征，

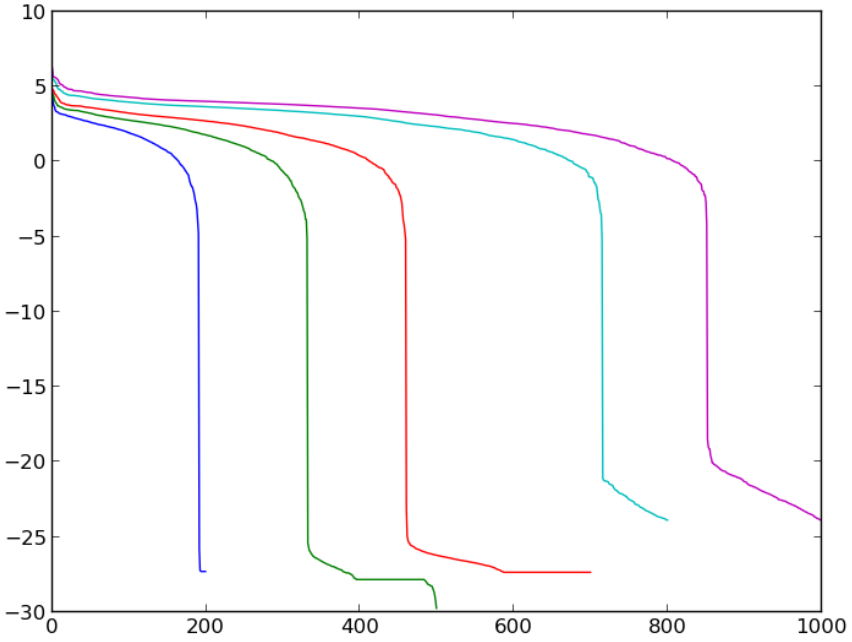


图 4.1 不同数量的样本对 ligo 数据的 pca 分布的影响

这样可以大大的减少特征的损失，所以用更多的数据集会更加的可靠。

同样，如果我们取得的样本的数量不够多的话，那么不但是我们会误删除更多的特征列，同时我们的 pca 的分布也会改变，这会导致 pca 模型不对，从而影响后面的训练的结果。对于不同样本数量我做了测试。

从左到右依次取的样本数量是 200，500，1000，2000，10268。这个分布图是对应数据集 0 的，从而我们可以看出对于 pca 训练，我们需要足够多的数据才能保证分布稳定。

对于 10 个数据集我画出了它们之间的 pca 分布图的对比（见图 3.4）。一个不错的问题是我们如果不去掉全 0 列那么是不是样本的数量的影响就会下降呢？如果我们不去掉全 0 的，我使用了 numpy.linalg.eigh 函数来进行 pca 分解，我这时得到的曲线图变得非常的陡峭（见图 3.5），将前面的特征值的重要性提升过多，这对机器学习而言我们在相同阈值下会舍去更多的特征。并且存在全 0 列对于判断 pca 主方向和机器学习而言都有影响。

另外我们可以看出的是，除了尾巴上的特性有点不同，前面的部分大致吻合，我们会舍去后面的尾巴，因此实际上这 10 组数据的 pca 模型是非常接近的。如果

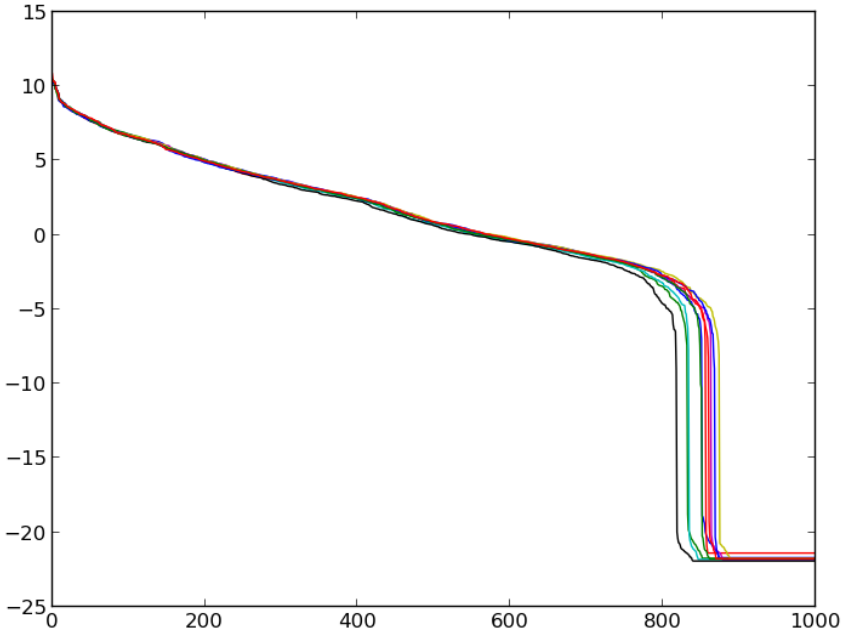


图 4.3　10 个集合含 0 列的 pca 分布对比图

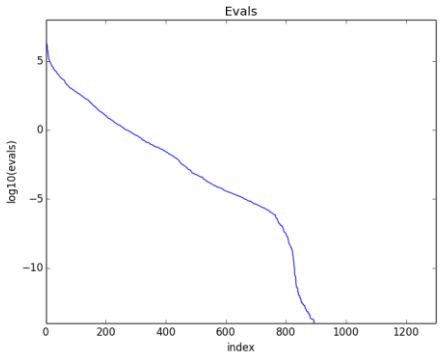
我们没有进行正规化，那么会怎样影响 pca 的分布呢？理论上没有进行正规化，

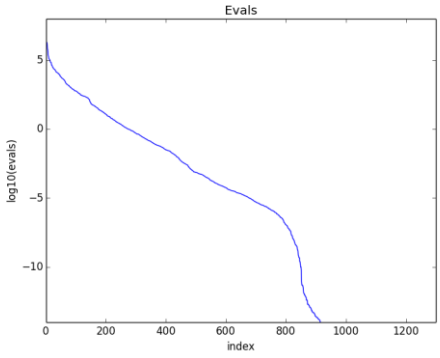
那么量纲的不同会导致量级差的很大，就好比 100000g 和 1m 进行对比，最终的结果是严重夸大某些量并且容易忽视某些量的作用。因此我们在进行 pca 分析的时候一定要进行正规化。这些统计量存储在 pca 模型文件中，随时可以调用。从图中可以看出正规化能使得 pca 分布曲线图变得更加的平滑，能反映主成分真实的贡献。我使用的正规化的方法是：

$$\frac{x - \bar{x}}{\sigma}$$

其中分母是标准偏差，先中心化再规范化，不过这个公式归一化到 n 的平方根。下面这幅图表示了正规化的作用，它能使得曲线更加的平滑，平摊了由于每个频道量纲的不同导致的实际分布的偏离。

如果没有正规化，那么 10 组数据的特征值的表现是否比较相似?所以我做了另外一组对比实验。可以看出区别也不是很大，说明这些数据集分布非常的均匀。使用这 10 组数据任何一组作为 pca 的模型实际上相差不会很大。我们出于上面的讨论还是使用完整的 10 万的数据集作为模型文件。





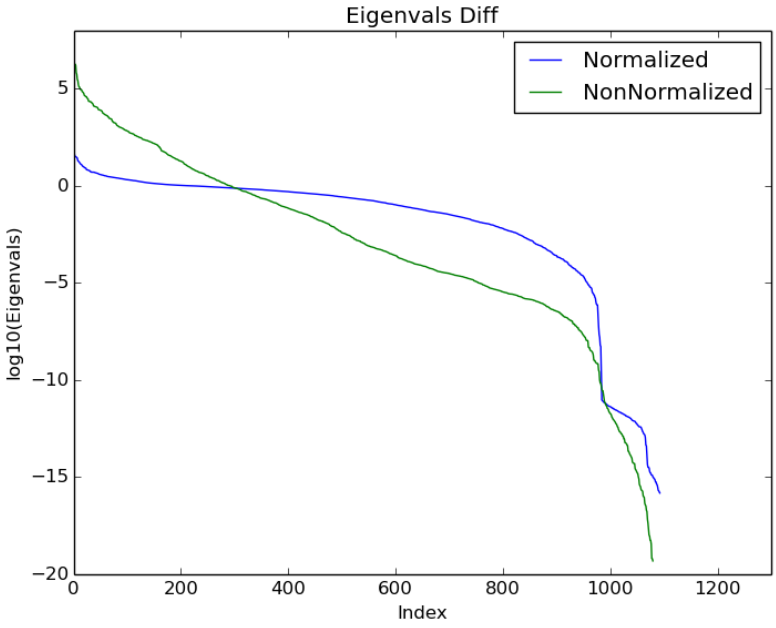
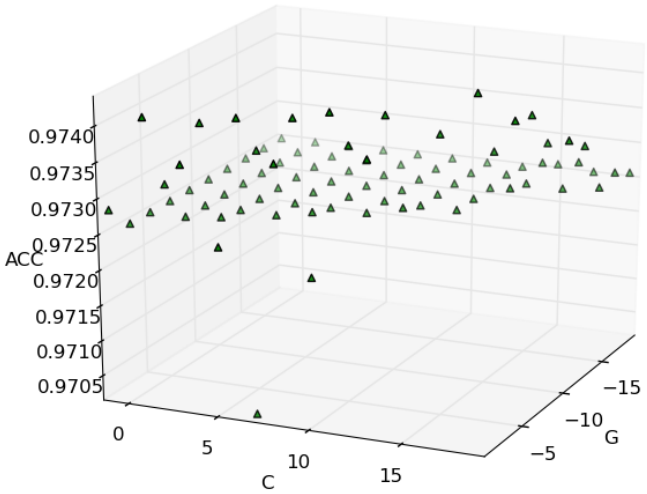


图 4.4　有无正规化对 pca 分布的影响和无正规化的数据集 0（左）和全集的 pca（右）分布对比

4.2 Libsvm 基本测试实验

首先是没有做正规化下的 pca 模型，选取了 120 个主成分作为最终训练特征



维数 ROC 曲线见图 3.8。计算花费 1 天 3 小时。通过对结果文件 acc 和 ligo-auc（通过对计算在 0.001-0.01 处的面积，这是 ligo 关心的致信区间）分析，最大值和最小值如下表，图中颜色深浅表示值的高低。

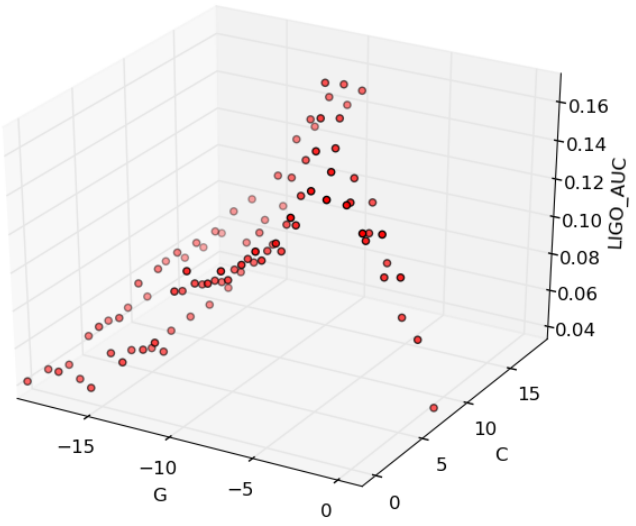


图 4.5　acc 和 ligo-auc 结果 3D 图(120pca)

表 4-1　120pca 结果

gamma	cost	acc
-9	-15	0.974322
gamma	cost	ligo-auc
-1	-1	0.171641

下面是 600 维的数据:

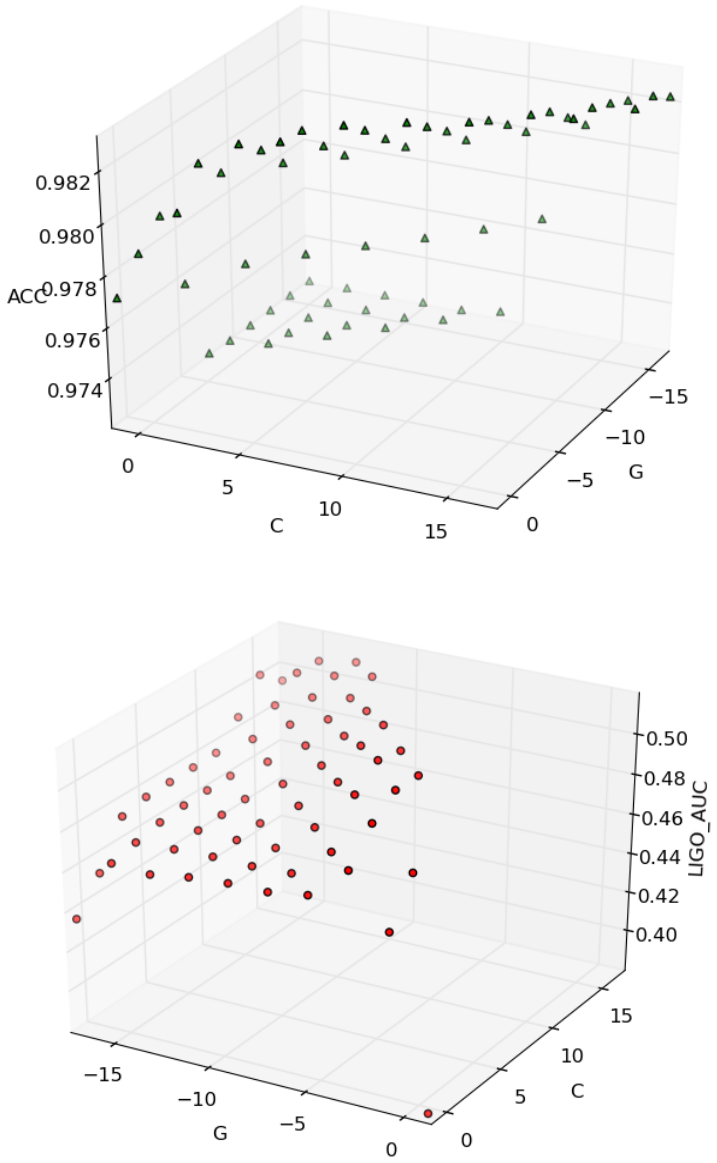


图 4.6 acc 和 ligo-auc 结果 3D 图(600pca)

表 4-2 600pca 结果

gamma	cost	acc
-5	7	0.983097
gamma	cost	ligo-auc
-5	5	0.517408

800 维数据图像

表 4-3　800pca 结果

gamma	cost	acc
-5	7	0.983097
gamma	cost	ligo-auc
-1	5	0.517408

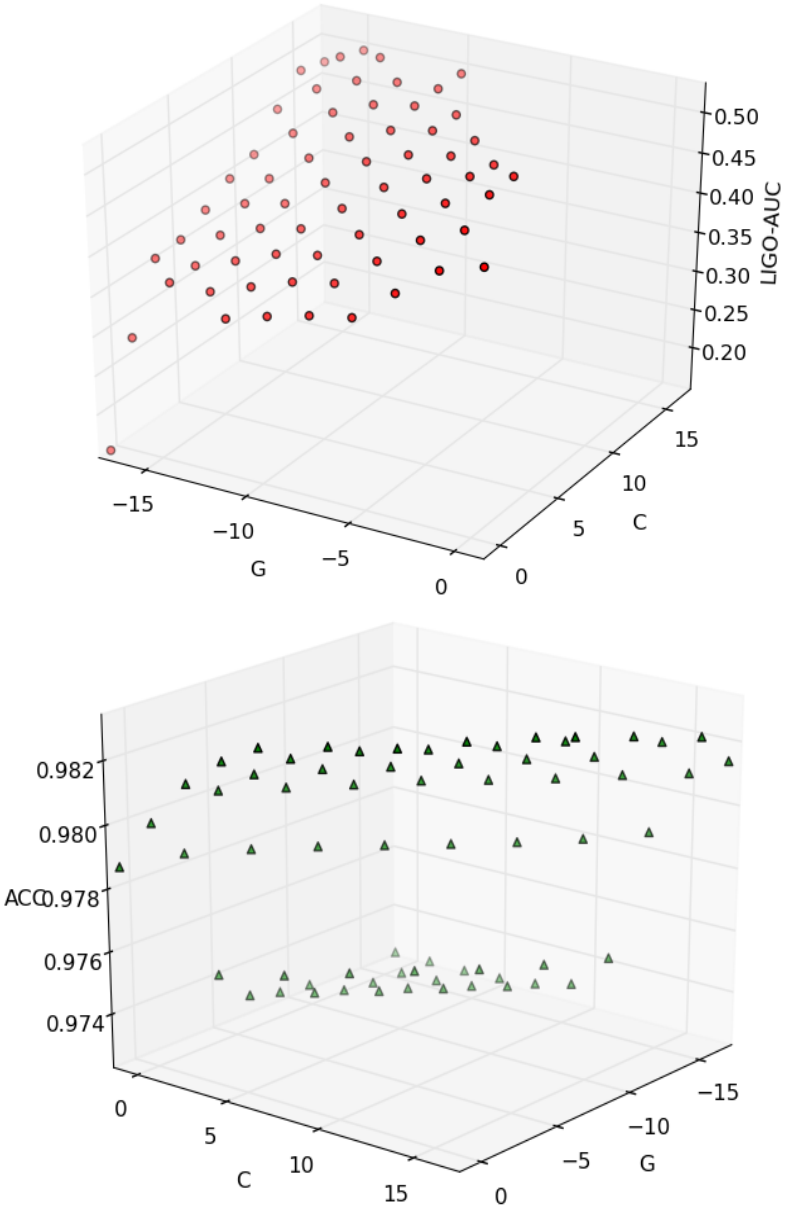


图 4.7　acc 和 ligo-auc 结果 3D 图(800pca)

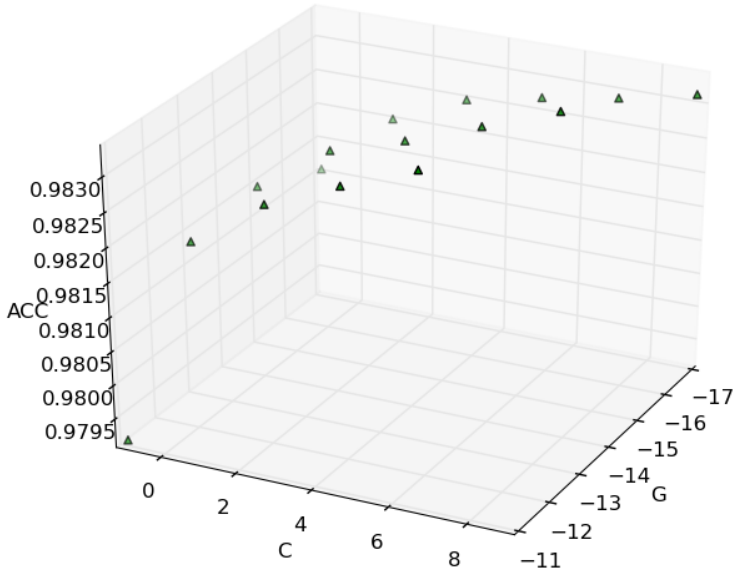
4.3 尝试可能改进 libsvm 结果方法

4.3.1 在 libsvm 环节去掉 scaling

考虑到做过 pca 后，数据已经 scaling 了，这时如果我们在 libsvm 训练的时候取消它的 scaling，那么我们可能保存更多 pca 压缩的信息，花了 3 天只计算 16 个点。看来速度上受到了影响。颜色深浅表示值的高低。

表 4-4　600pca no scaling 结果

gamma	cost	acc
-5	7	0.983358
gamma	cost	ligo-auc
-13	3	0.527026



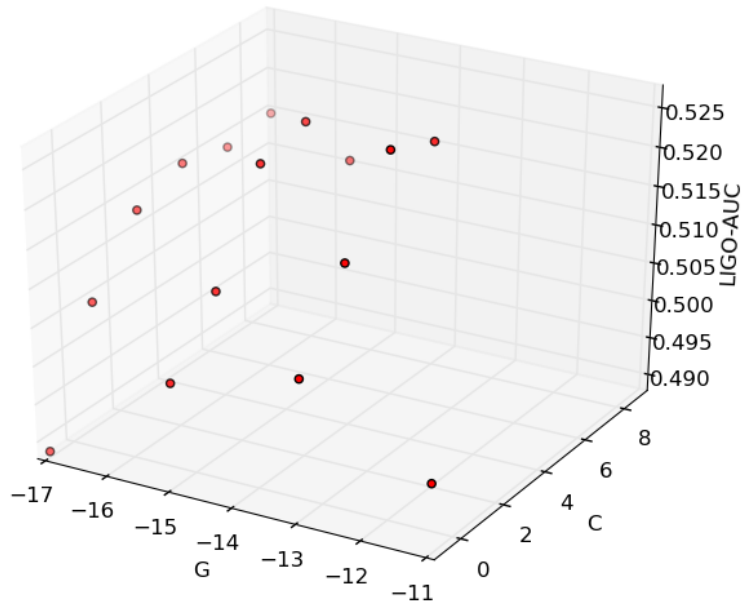


图 4.8　acc 和 ligo-auc 结果 3D 图(600pca no scale)

4.3.2 加入 label 实现有监督的学习

Pca 是一种非监督模式的分类方式，因此我们可以加入 label 一起进行主成分分析，然后再按照一定规则进行删除特征，历时 2 天 4 小时.做法如下：

1) 准备数据，然后加入 label，形成 N+1 维的矩阵，然后按照同样的办法进行正规化。应用 pca，得到 N*N 的变换矩阵 W
2) 压缩小的权重

   （a）对于每一列都求绝对值均值 $a_i=\sum|w_{ij}|$

   （b）取定一个小常数 α，如果 $|w_{ij}|<\alpha a_i$，那么就将 w_{ij} 设为 0
3) 提取特征和选择特征

   去掉 W 的 label 列，然后应用在原始数据（正规化后的 N 维矩阵）上，得到 N+1 维新矩阵。如果 W 相应的列为 0，那么就删除对应的特征。

表 4-5　600pca with label 结果

gamma	cost	acc
-5	7	0.983358
gamma	cost	ligo-auc
-13	3	0.527026

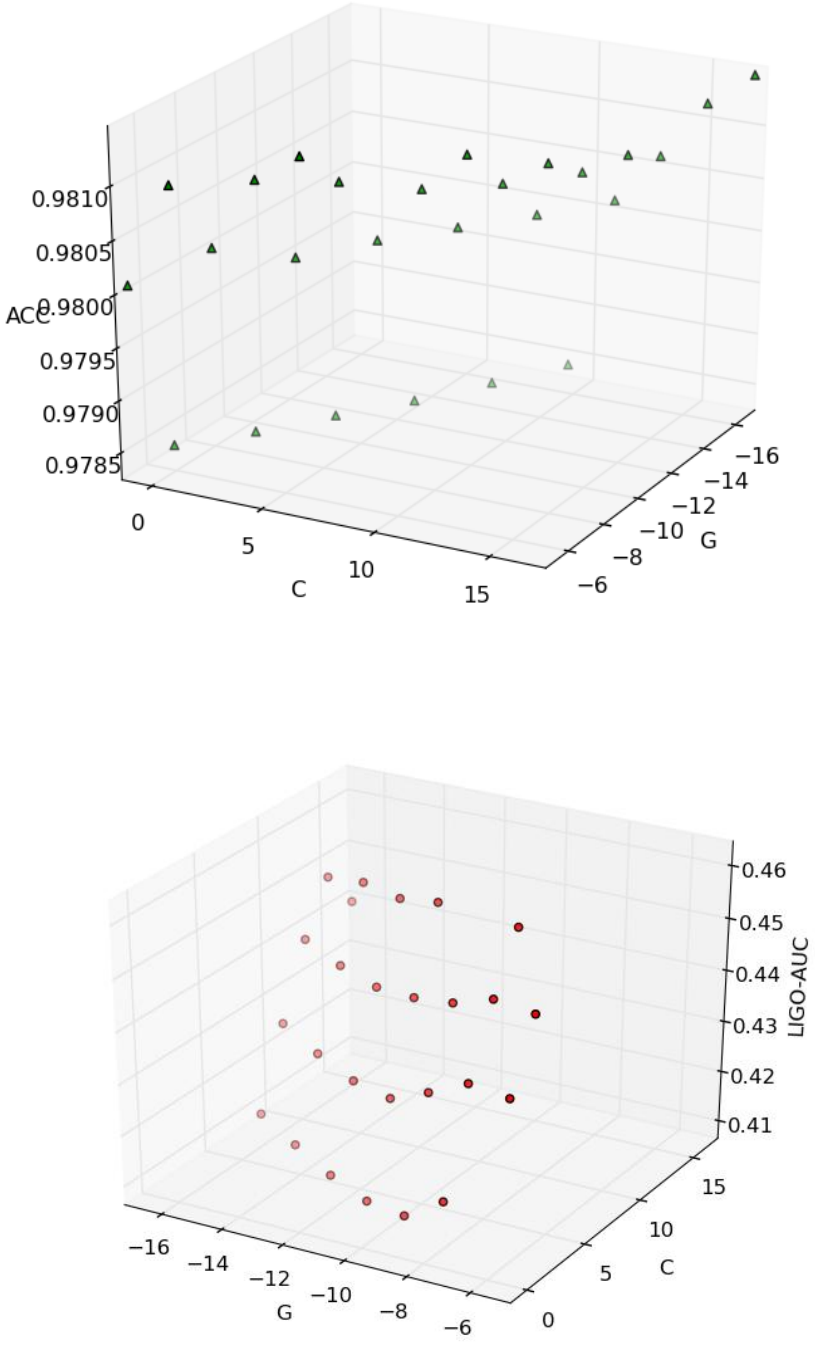


图 4.9　acc 和 ligo-auc 结果 3D 图(600pca with label)

4.3.3 使用 fscore 去除噪声再进行 pca

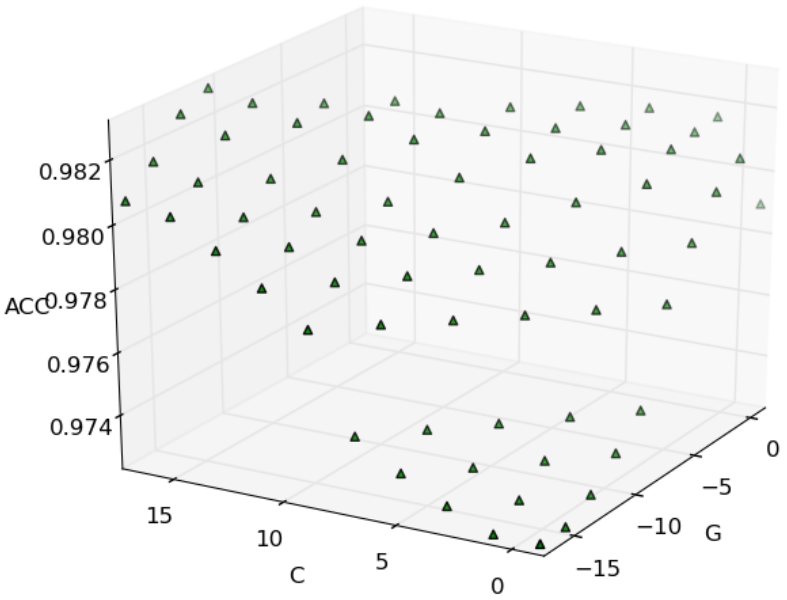
Fscore 是特征选择使用的方法，于是我猜想如果能够先使用 fscore 滤过一些特征再使用 pca，对结果是否会有帮助。Fscore 是纯统计学方法，使用的公式如下：

事先舍去了低于 0.00001 的特征，得到 1032 维数据，在这个基础上运行 pca，于是有如下结果：

表 4-6 600pca with fscore 结果

gamma	cost	acc
-5	7	0.982988
gamma	cost	ligo-auc
-9	13	0.495573

$$F(i) \equiv \frac{(\bar{x}_i^{(+)} - \bar{x}_i)^2 + (\bar{x}^{(-)} - \bar{x}_i)^2}{\frac{1}{n_+ - 1}\sum_{k=1}^{n_+}(x_{k,i}^{(+)} - \bar{x}_i^{(+)})^2 + \frac{1}{n_- - 1}\sum_{k=1}^{n_-}(x_{k,i}^{(-)} - \bar{x}_i^{(-)})^2}$$



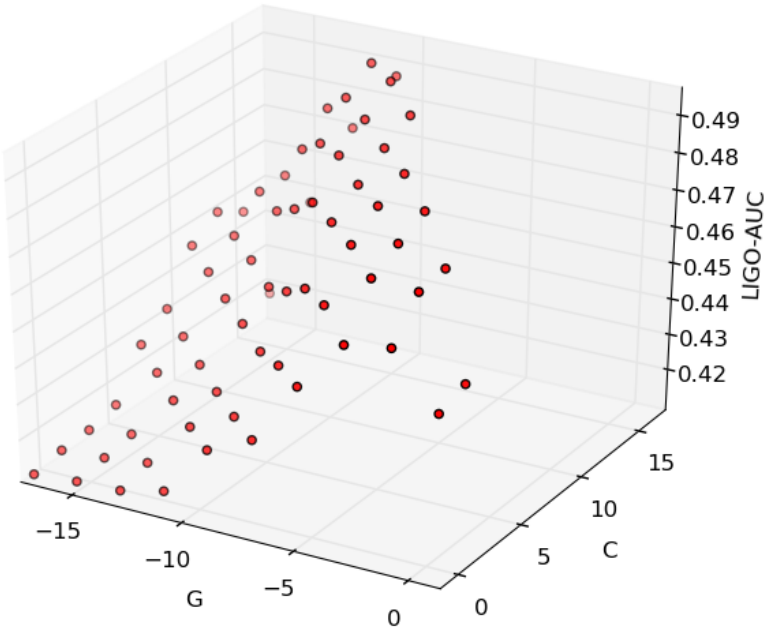


图 4.10 acc 和 ligo-auc 结果 3D 图(600pca with fscore)

第5章 结论与展望

下面的是 fscore 在 10 万个 ligo 数据上的测试结果，从图中可以看出大部分数据的可分性不是很高。我从中选取了 1032 个新的特征做 pca，猜测可能会降低噪声。

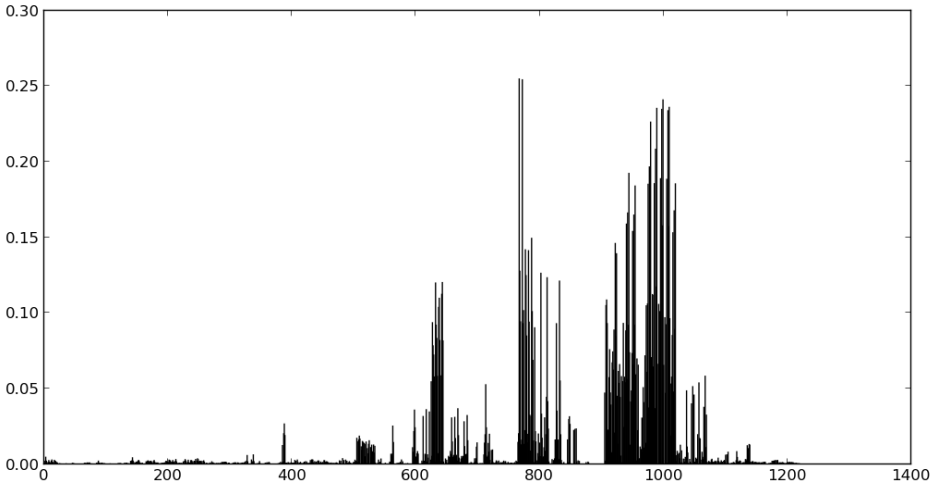


图 5.1 10 万个数据的 fscore 分布图

很多对于学习而言可能成为干扰能够与服务器进行数据交换。之后学习了 pca 的算法描述，并且使用 python 进行模块化实现，实现代码的重用。然后学习了有关了机器学习相关知识和着重查阅了 libsvm 的使用文档，了解基本的学习过程: 先准备 svm 数据集，然后 scaling，得到训练模型，之后用训练模型进行预测。由于数据集庞大，开始的时候决定使用多少组样本做 pca 才合适，于是就探究了一下 pca 分布与样本数量的关系。

ROC 曲线实验分几个过程进行，首先尝试 120 维数据的效果，由 ROC 曲线图不难发现和原始的学习效果比起来差太多，于是猜测可能是由于压缩过多的信息导致的，就选择 600 维数据，发现表现良好。另外尝试了三种可能使得最后结果变好的方法 1）去掉机器学习 scaling 的计算 2）使用加上 label 使之变成无监督的学习 3）使用 fscore 去掉可能是噪声的特征。实验结果使用 ROC 曲线表示，主要是看 0.001-0.01 之间的面积来决定效果的优劣。从结果上看，去掉 scaling 后运行速度大大降低，但是最接近原始数据，fscore 方法速度变快，但是精度不如去掉 scaling 的方法。加上 label 后运行速度也降低，搜索出来的 grid 也少，效果不是很理想。但是这几种方法都能减少运行时间，在精度要求区别不是很大的情况下我们可以使用。

实验中碰到了很多问题，例如 python 可以使用 csv 这样的库效率可能会更高，我自己写的文本处理模块效率太低，导致自己跑实验效率不高。其次在做加 label 实验的时候，alpha 取值比较大的时候没发现有全 0 列，后面 debug 发现求平均值求成行的，没注意到。另外做 pca 的时候没考虑代码重用，使得每次跑都要重新做一次 pca，非常的慢，尤其是对于 10 万个数据，后面生成模型文件存储信息，只需要一次运行就可以。试验中有很多需要改进的地方比如说需要更多维数的测试(400，300 等维度)这样我们可以知道这些方法是不是和维度也密切相关的，可能在其他情况运行的更好。另外搜索的范围和精度也有可能有很大提升空间。

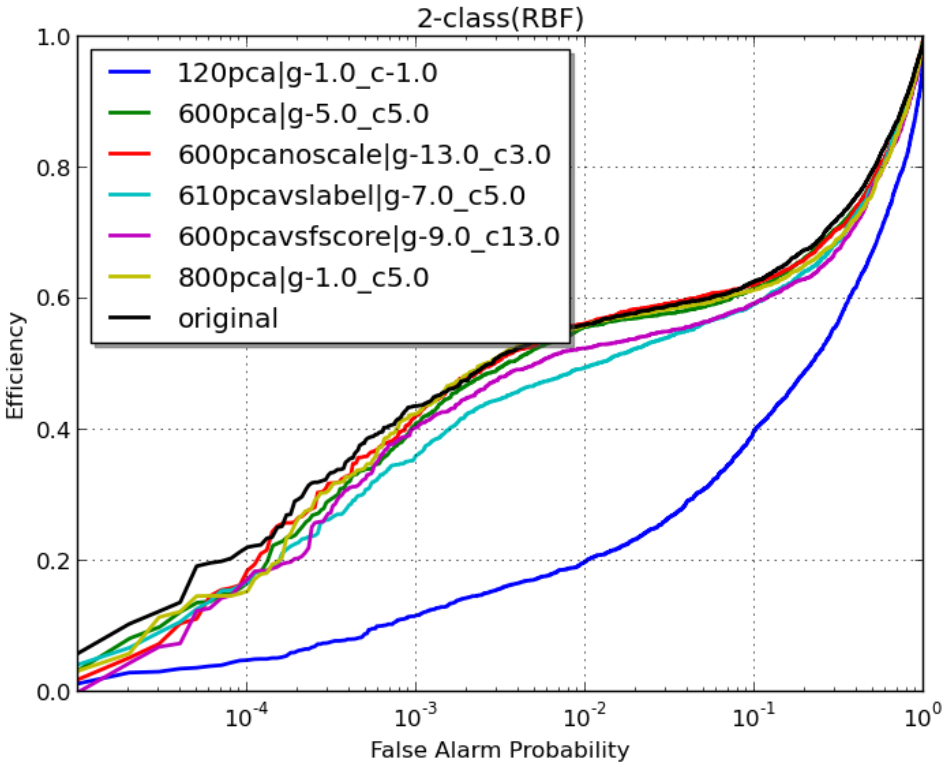


图 5.2　所有实验结果 ROC 曲线图

下面是搜索面的立体图，可以看出有些搜索达到了 g（-17~1）c（-1~17）搜索边界，因此可以对算法的搜索半径和精度提高。

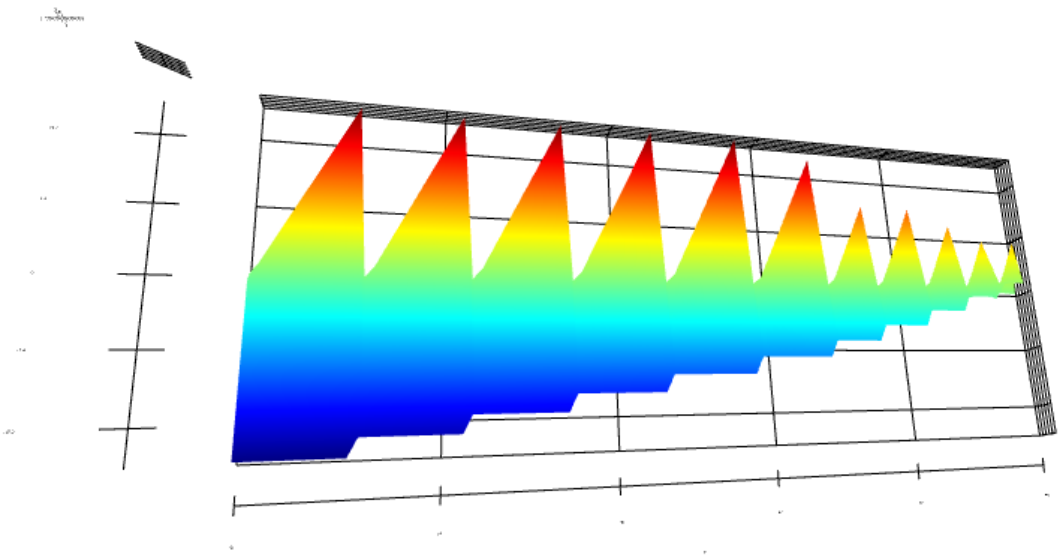


图 5.3　120pca 搜索图

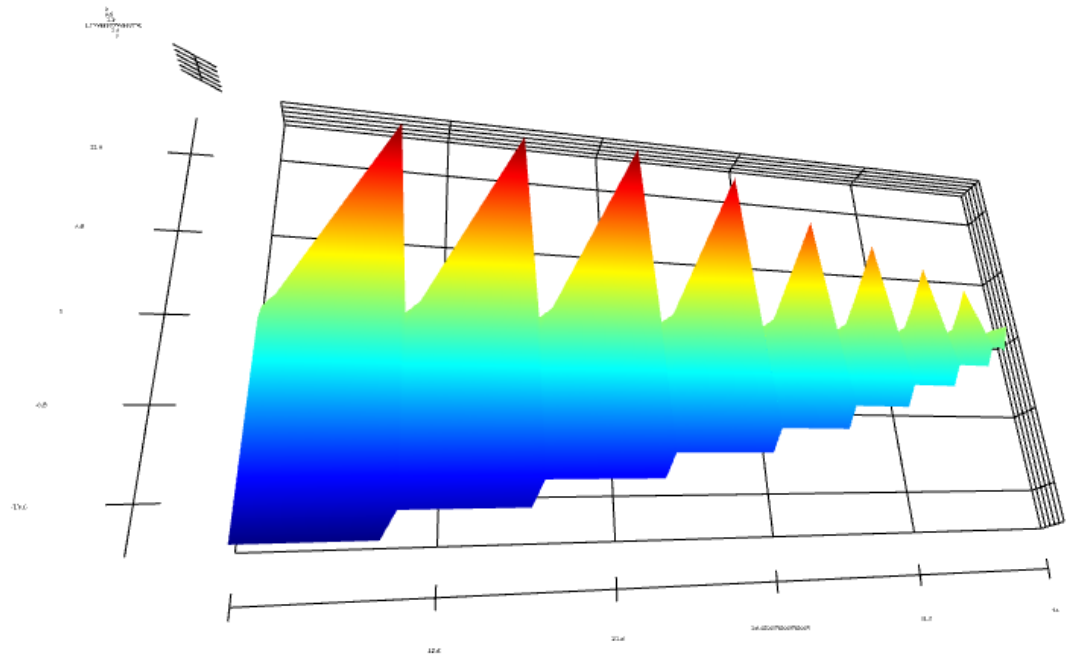


图 5.4　600pca 搜索图

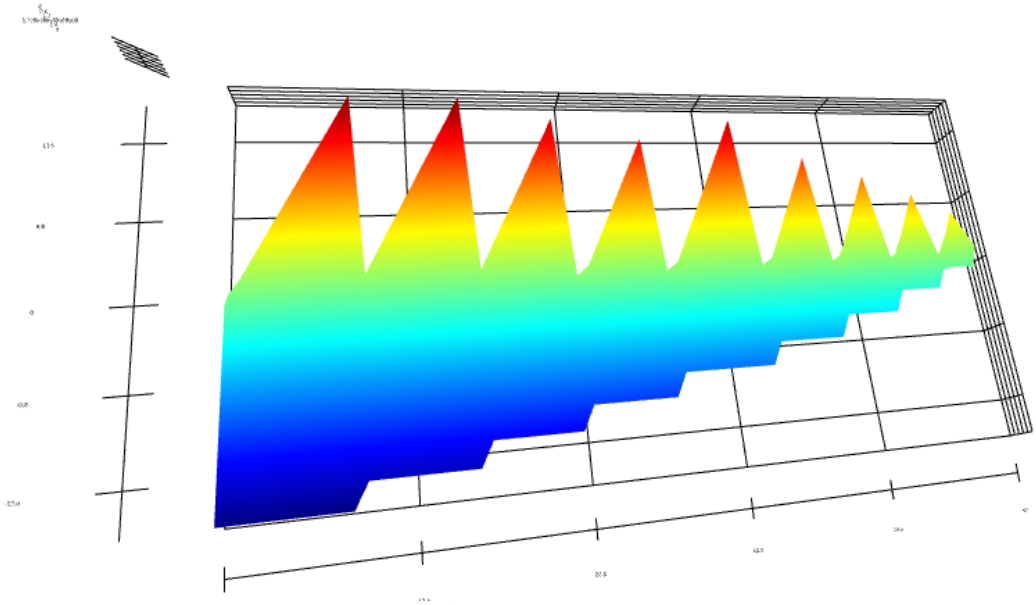


图 5.5　800pca 搜索图

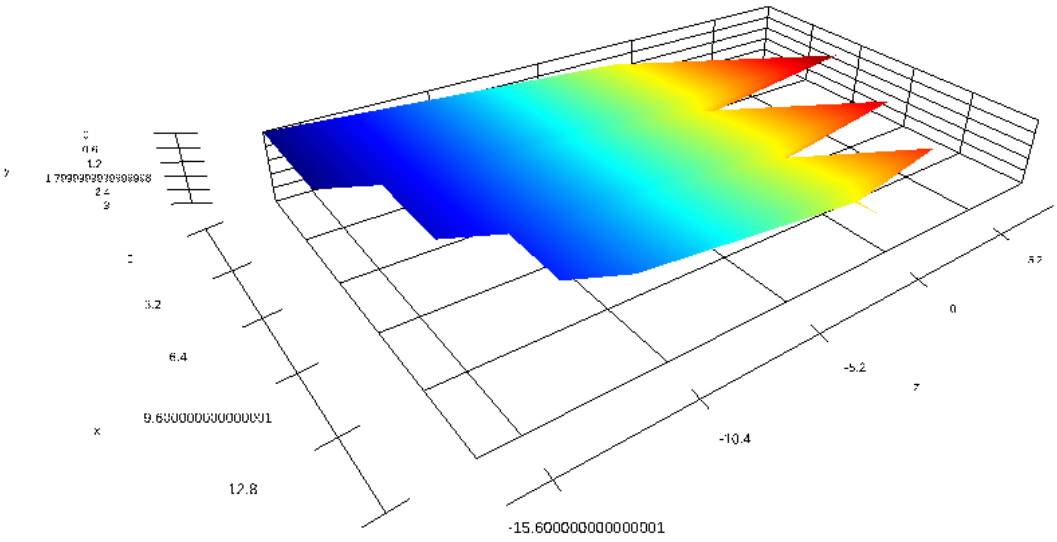


图 5.6　600pca no scaling 搜索图

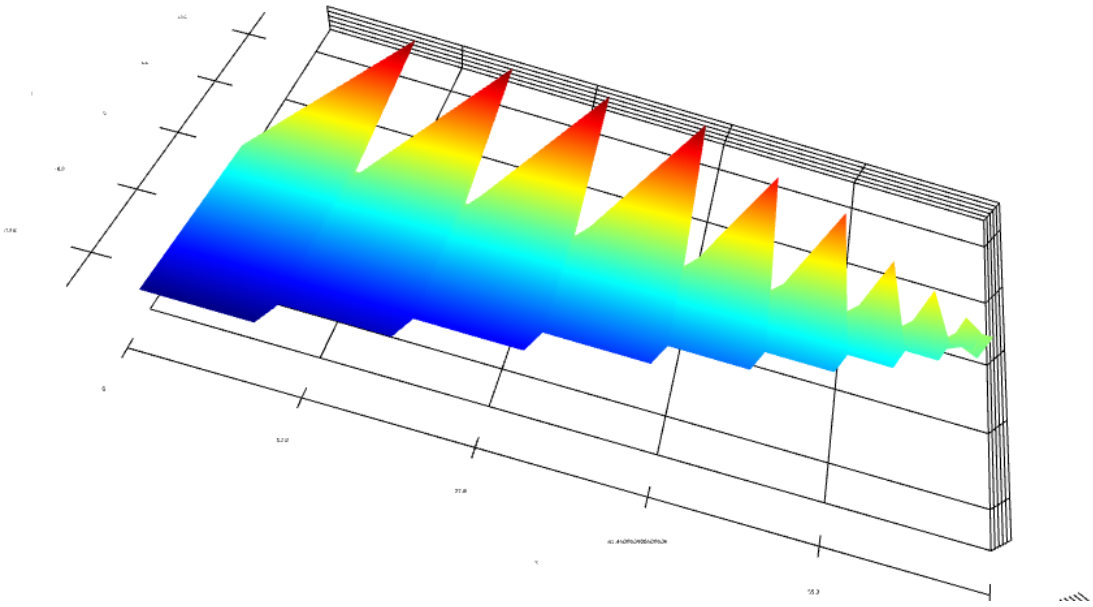


图 5.7　600pca fscore 搜索图

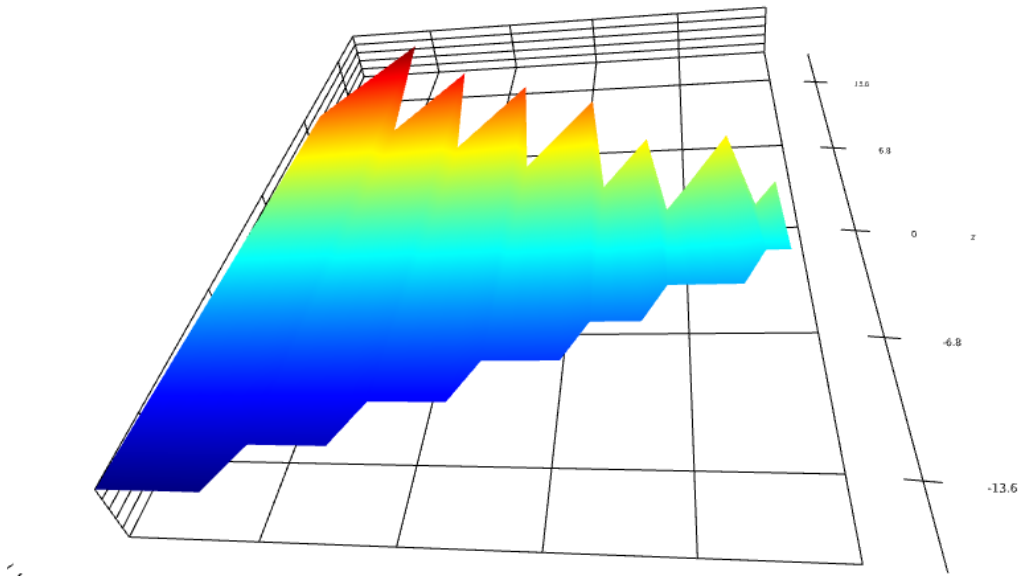


图 5.8　600 with label pca 搜索图

插图索引

表格索引

参考文献

[1] Clarke DW. Application of generalized predictive control to industrial processes. IEEE Control Systems Magazine 1988; 122:49–55

[2] Kyungnam Kim ,Face Recognition using Principle Component Analysis, Department of Computer Science University of Maryland, College Park MD 20742, USA

[3] BICEP2 2014 Release Papers http://bicepkeck.org/

[4] A.D. Back and T.P. Trappenberg, Input variable selection using independent component analysis, The 1999 Int'l Joint Conf. on Neural Networks, July 1999

[5] H. Liu and H. Motoda, Less is more, Feature Extraction Construction and Selection, pp. 3-11, Boston: Kluwer Academic Publishers, 1998

[6] LIGO page http://www.ligo.org/news/bicep-result.php

[7] AuxMVC wiki page[OL] https://wiki.ligo.org/foswiki/bin/view/DetChar/AuxMVC

[8] University of Nebraska Medical Center. ROC wikipage[OL]. http://gim.unmc.edu/dxtests/roc3.htm

[9] Abramovici A, el al. LIGO: the laser interferometer gravitional-wave observatory[J]. Science, 1992: 256:325~333

[10] Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. Journal of Educational Psychology, 24, 417-441, and 498-520

[11] Deco & Obradovic (1996). An Information-Theoretic Approach to Neural Computing. New York, NY: Springer

[12] Isogai T and the LIGO Scientific Collaboration and the Virgo Collaboration, Used percentage veto for LIGO and Virgo binary inspiral searches. Journal of Physics: Conference Series, 2010, 243:012005

[13] Yang ZR, Biological applications of support vector machines[J]. Briefings in bioinformatics. 2004 Dec;5(4)328-38

[14] Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines[J]. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011

[15] Warmuth, M. K.; Kuzmin, D. (2008). "Randomized online PCA algorithms with regret bounds that are logarithmic in the dimension". Journal of Machine Learning Research **9**: 2287–2320

[16] Shaw P.J.A. (2003) Multivariate statistics for the Environmental Sciences, Hodder-Arnold. ISBN 0-340-80763-6

[17] M. Andrecut. Parallel GPU Implementation of Iterative PCA Algorithms. Journal of Computational Biology, 16(11), Nov. 2009

[18] Chris Ding,Xiaofeng He, *K*-means clustering via principal component analysis, ICML '04 Proceedings of the twenty-first international conference on Machine learning, Page 29

[19] Timothy A. Brown. Confirmatory Factor Analysis for Applied Research Methodology in the social sciences. Guilford Press, 2006

[20] Greenacre, Michael (1983). Theory and Applications of Correspondence Analysis. London: Academic Press. ISBN 0-12-299050-1

[21] Wiki page http://en.wikipedia.org/wiki/Machine_learning

[22] 图片 http://pic.baike.soso.com/p/20100214/20100214111425-447445318.jpg

[23] 图片 http://pic.baike.soso.com/p/20101018/20101018142630-1627827228.jpg

[24] 图片 http://withfriendship.com/images/c/14253/Principal-component-analysis-image.gif

[25] 图片 http://tech.ddvip.com/2013-07/1374003291199149.html

[26] 图片 http://www.trendcomputing.de/wp-content/uploads/2010/12/grid1.jpg

[27] Condor and the Grid, Douglas Thain, Todd Tannenbaum, and Miron Livny, Computer Sciences Department, University of Wisconsin-Madison 1210 West Dayton Street, Madison WI 53706

[28] http://www.cnblogs.com/heaad/archive/2011/01/02/1924088.html

致 谢

首先应该感谢导师曹军威老师和王小鸽老师，曹老师给我一个能够接触 LIGO 这个科技前沿的项目的机会，同时感谢王老师耐心的指导，不仅教会我很多学术上的事物，而且在对待科学的态度方面也给予我很大的帮助。

这里也同时感谢季颖生博士在机器学习方面给予我的帮助和支持，并且感谢他给出的很多具有建设性意义的想法，这对我帮助很大。

感谢 Eric O.Lebigot 在工作中的指导，每次都能提出关键性的问题和新颖的想法，同时严谨的科学态度也使我获益匪浅。

在此感谢所有帮助和支持我的人，谢谢!

声 明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签　名：_____　　日　期：_____

附录 A 外文资料的调研阅读报告（或书面翻译）

调研阅读报告题目（或书面翻译题目）

写出至少 5000 外文印刷字符的调研阅读报告或者书面翻译 1-2 篇（不少于 2 万外文印刷符）。

参考文献（或书面翻译对应的原文索引）

[18] Chris Ding,Xiaofeng He, *K*-means clustering via principal component analysis, ICML '04 Proceedings of the twenty-first international conference on Machine learning, Page 29

通过主成份分析进行 K 均值聚类

Chris Ding

Xiaofeng He

Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720

摘要：

主成份分析是一个用来实现非监督的维数压缩而被广泛采用的统计技术。K 均值聚类是在进行非监督学习任务时的一种数据聚类方法。这里我们要证明主成份是 K 均值聚类离散聚类成员的连续解决方法。同样的，我们证明了通过聚类中心扩展的子空间是由数据协方差矩阵频谱的 K-1 截断部分给出的。这些结果表明，无监督降维和无监督学习是密切相关的。在降维方面，结果为基于 pca 的降维方法提供了新的见解，超越了常规降噪解释所得的效果。映射数据点到通过内核更高维空间中，我们得知核 k 均值是需要核 pca 来解决的。在学习方面，我们的结果指明了一些有用的技术。通过对基因组和互联网新闻的分析研究来阐明一些结果。实验表明新 k 均值的下界是目标是 0.5%-1.5%的最优值。

1.介绍

数据分析方法对于分析增长越来越快的高维数据规模而言变得非常重要。一方面，聚类分析（Duda et al., 2000;Hastie et al., 2001; Jain & Dubes, 1988）试图通过数据传递迅速获得一阶知识 通过分区数据点成不相交的群体， 属于同一个群集的数据点是相似的，而属于不同的簇的数据点是不一样的。其中一个最有效和流行的方法之一是 k 均值聚类（ Hartigan & Wang, 1979; Lloyd, 1957; MacQueen, 1967 ）它使用以最小方差函数优化的中心点来代表分类。（关于详细信息 k 均值的详细信息可以参照（ Jain & Dubes, 1988 ），（Wallace，1989）。

另一方面，高维数据常用 pca（Jolliffe, 2002 ）来压缩到低维数据，这样便于观测相关的特征。这样的无监督降维是用在非常广泛的领域，如气象，图像处理，基因组分析和信息检索。PCA 和 K 均值用于将数据投影到低维子空间是很常见的，然后在该子空间（Zha et al., 2002）而定。在其他情况下，数据被嵌入在低维空间中，如本征空间的图拉普拉斯算子，然后再是应用 K 均值。基于 PCA 的降维的主要依据是该 PCA 找到具有最大方差的尺度。在数学上，这相当于找到的最好的低秩逼近（在 L2 范） 通过奇异值分解（SVD）的数据 （Eckart

& Young, 1936 ）。然而，这种噪声降低的方法不足以解释 pca 的有效性。

在本文中，我们探讨它们之间的连接两种广泛使用的方法。我们证明了主成份实际上是一个群集成员指标的 K 均值聚类方法的连续解决方案，即 PCA 降维自动 好的光盘根据该执行数据聚类的 K-表示的目标函数。这提供了一个重要的基于 PCA 的数据缩减的判据。

我们的研究结果也提供了有效的方法解决了 K-均值聚类问题。K-均值法使用 K 原型簇的形心，对数据进行表征。它们是通过最小化平方和来确定误差的，

$$J_K=\sum_{k=1}^{K}\sum_{i\in C_k}(x_i-m_k)^2$$

其中 $(x_{1,}x_{2,}..,x_n)=X$ 是数据矩阵，而 $\boldsymbol{m_k}=\sum_{i\in C_k}\boldsymbol{x_i}/n_k$ 是集合 C_k 的中心点，并且 n_k 是集合 C_k 中点的个数。标准的迭代解决方案的 k-means 患有众所周知的问题：随着迭代的进行，方法被困在由于算法更新的贪婪而导致的局部极小(Bradley & Fayyad, 1998; Grim et al., 1998; Moore, 1998) 。

一 些 关 于 PCA 的 记 录 。 X 表 示 原 始 的 数 据 矩 阵 ； $Y=(y_{1,}y_{2,}..,y_n),y_i=x_i-\bar{x}$，表示中心矩阵，其中 $\bar{x}=\sum_i\frac{x_i}{n}$ 。协方差矩阵（忽略 1/n 的因子）是 $\sum_i(x_i-\bar{x})(x_i-\bar{x})^T=YY^T$ 。主成份方向 u_k 和主成份 v_k 是特征向量满足：

$$YY^T\boldsymbol{u_k}=\lambda\boldsymbol{u_k},Y^T\boldsymbol{Y}\boldsymbol{v_k}=\lambda\boldsymbol{v_k},boldv_k=Y^T\boldsymbol{u_k}/\lambda_k^{1/2}\quad(1)$$

这些是 svd 的定义方程： $Y=\sum_k\lambda_k^{1/2}\boldsymbol{u_k}\boldsymbol{v_k}$ (Golub & Van Loan, 1996) ,其中 v_k 是在主成份方向 u_k 上的投影值。

2.两种方式的聚类

考虑 K=2 的情况。令

$$d(c_k,c_l)\equiv\sum_{i\in c_k}\sum_{j\in c_l}(x_i-x_j)^2$$

作为两个聚类 c_k,c_l 之间的距离。经过代数变换我们得到

$$J_k=\sum_{k=1}^{K}\sum_{i,j\in c_k}\frac{(x_i-x_j)^2}{2\mathrm{n}_k}=n\bar{y^2}-\frac{1}{2}J_D,(2)\ \text{并且}$$

$$J_D=\frac{n_1n_2}{n}[2\frac{d(C_{1,}C_2)}{n_1n_2}-\frac{d(C_{1,}C_2)}{n_1^2}-\frac{d(C_{1,}C_2)}{n_2^2}],(3)$$

其中 $\bar{y^2}=\sum_j\boldsymbol{y_j^T}\boldsymbol{y_j}/n,$ 是一个常数。这样求 min(J_K)就是相当于求 max(J_D)。另外，我们可以证明：

$$\frac{d(C_1,C_2)}{n_1 n_2}=\frac{d(C_1,C_2)}{n_1^2}+\frac{d(C_1,C_2)}{n_2^2}+(\boldsymbol{m_1}-\boldsymbol{m_2})^2,(4)$$

用方程 4 去替代方程 3，我们可以看到 J_D 永远是正数。我们在以下定理中总结这些。

定理 2.1 对于 K=2，最小化 K 均值分类目标函数 J_K 等同于最大化距离函数 J_D，它永远是正的。

备注：（1）在 J_D 中，第一项代表两个聚类的平均距离，需要最大化。这保证两个分类结果尽量的分开。（2）第二项和第三项表示平均类内距离，需要最小化；这保证每个类尽量的紧密。这在方程 2 中也十分明显。（3）n_1n_2 因子能让类均衡，而 $J_D>0$，$\max(J_D)$ 希望求 n_1n_2 的最大值，这就导致 $n_1=n_2=n/2$ 。

这些备注给予 k 均值的一些见解。然而最重要的部分是 J_D 能够通过 pca 求出一个解法。

定理 2.2 对于 K=2 的 k 均值聚类，群集指标向量的连续解是主成分 v_1, i.e.,聚类 C_1 和 C_2 可以由以下得出：

$$C_1=\{i\mid v_1(i)\leqslant 0\},C_2=\{i\mid v_1(i)\geqslant 0\},(5)$$

K 均值目标最优值满足边界：

$$n\bar{y}^2-\lambda_1<J_k=2<n\bar{y}^2,(6)$$

证明：

考虑一个方阵 $D=(d_{ij})$,其中 $d_{ij}=\|x_i-x_j\|^2$ 。让聚类指示向量为：

$$q(i)=\begin{cases}\sqrt{n_2/nn_1} & if\ i\in C_1\\ -\sqrt{n_1/nn_2} & if\ i\in C_2\end{cases},(7)$$

这个指示向量满足和为零并且是规范化的：$\sum_i q(i)=0,\sum_i q^2(i)=1$。 可以清晰看到 $q^TDq=-J_D$ 。如果我们放松约束，让 q 不用选取离散的两个值而是在[-1,1]之间取连续的值，那么 $J(q)=q^TDq/q^Tq$ 的最小解由对应 $Dz=\lambda z$ 最小特征值的特征向量。一个对离散指示向量 q 更好的放送约束的条件是使用中心矩阵 D，也就是减去列和行的均值，

$$\hat{d}_{ij}=d_{ij}-d_{i\cdot}/n-d_{\cdot j}/n+d_{\cdot\cdot}/n^2,(8)$$

其中

$$d_i=\sum_j d_{ij},d_{\cdot j}=\sum_i d_{ij},d_{\cdot\cdot}=\sum_{ij} d_{ij}$$ 。

现在我们有 $q^T\hat{D}q=q^TDq=-J_D$,既然公式 8 中第二三四项贡献在 q^TDq 为 0。所以 $J(q)=q^TDq/q^Tq$ 的最小解是对应 $\hat{D}z=\lambda z$ 最小特征值的特征向量。

通过构造，这个中心化的矩阵有一些非常好的性质，那就是每一行和列的和都为 0。这样 $e=(1,..,1)^T$ 就是 $\hat{D}$ 的 $\lambda=0$ 对应的特征值。既然它的所有特征向量都与 e 正交，那么它们也有和为 0 的性质， $\sum_i z(i)=0$,也就是初始指标向量的确定性质。相比之下， $Dz=\lambda z$ 没有这个性质。

经过代数计算，将 $d_i=nx_i^2+n\overline{x^2}-2nx_i^2\overline{x}, d_{..}=2n^2\overline{y^2}$ 带入公式 8，得到

$\hat{d}_{ij}=-2(x_i-\overline{x})^T(x_i-\overline{x})$ *或者是* $\hat{D}=-2Y^TY.$

所以堆指示向量的连续解是对应最大特征值的 gram 矩阵 Y^TY 的特征向量，也就是主成份向量。 $J_D<2\lambda_1$,其中 λ_1 是协方差矩阵的主要特征值。通过方程 2 我们得到 J_K 的边界。

图 1 展示了主成份在 K 均值分类怎么决定堆的成员。一旦 C1，C2 被主成份通过方程 5 确定下来，我们可以计算当前均值 m_k 和迭代 K 均值直到收敛。这将会得到分类的局部最优解。我们将它称为 PCA 指导的 K 均值聚类。

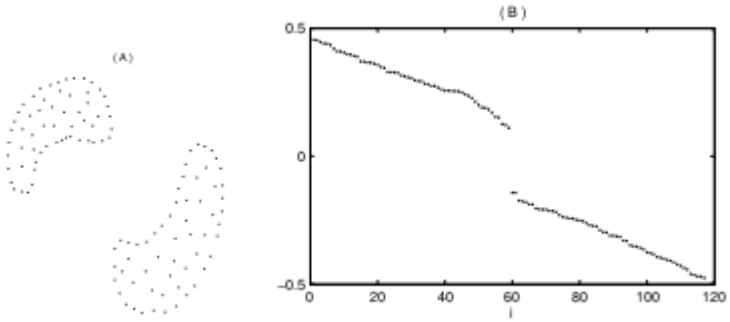


图 1.（A）两个在 2D 平面内的聚类（B）主成份，显示每个元素的值。

3. K 种聚类

之前我们讨论的是 K=2 的时候使用单一的指示向量。这里我们一般化到 K>2，使用 K-1 个指示向量。

规则化的减少约束

这个通用的方法第一次应用在(Zha et al., 2002)。这里我们提出一个更加具有扩展性和平滑的方案和一个连结性的分析。首先，观察方程 2， J_k 可以被写成

$$J_K=\sum_i x_i^2-\sum_k \frac{1}{n_k}\sum_{i,j\in C_k} x_i^Tx_j, \quad (9)$$

第一项是一个常数。第二项是 K 个 X^TX 的对角线元素表示聚类内部的相似程度。这个聚类的解可以用 K 个非负的指示向量：

$H_K=(h_{1,..},h_k)$, 其中 $h_k=(0,..,0,1,..,1,..,0)^T/n_k^{1/2}$,(10)

(在不是一般性的前提下，我们检索在聚类内的相互临近的数据点。）利用方程 9，它变成了 $J_K=Tr(X^TX)-Tr(H_k^TX^TXH_k)$,(11) 其中 $Tr(H_k^TX^TXH_k)=h_1^TX^TXh_1+...+h_k^TX^TXh_k$. H_K 中是有冗余的。例如，$\sum_{k=1}^{K}n_k^{1/2}h_k=e$ 这样其中一个 h_k 是其他分量的线性组合。我们通过进行线性变换来将 T 映射到 q_k ：

$Q_k=(q_{1,..},q_k)=H_KT$ 或者是 $q_l=\sum h_k t_{kl}$,(12)

其中 是一个 k*k 的正交矩阵： $T^TT=I$ 并且要求 T 最后一列是：

$t_n=(\sqrt{\frac{n_1}{n}}h_1+...+\sqrt{\frac{n_k}{n}}h_k)^T$,(13)

所以我们总有：

$q_k=\sqrt{\frac{n_1}{n}}h_1+...+\sqrt{\frac{n_k}{n}}h_k=\sqrt{\frac{1}{n}}e$

这个线性变换总是可能的（之后可以看见）。例如当 K=2，我们由

$T=\begin{pmatrix}\sqrt{n_2/n} & -\sqrt{n_1/n}\\ \sqrt{n_1/n} & \sqrt{n_2/n}\end{pmatrix}$,(14) 并且 $q_1=\sqrt{n_2/n}h_1-\sqrt{n_1/n}h_2$,正是方程 7 所说的指示向量。这个 K 聚类的算法是对 K=2 的一般化。

相互正交的 h_k 满足 $h_k^Th_l=\delta_{kl}$($\delta_{kl}=1$ 如果 $k=l$；其他 0）表明 $q_k^Tq_l=\sum_p h_p^Tt_{pk}\sum_s t_{sl}=\sum_p(T^TT)_{kl}=\delta_{kl}$ 让 $Q_k-1=(q_{1,..},q_k-1)$，那么上面的正交关系可以被表示为：

$Q_{k-1}^TQ_{k-1}=I_{k-1}$,(15)

$q_k^Te=0,k=1,...,K-1$,(16)

现在 k 均值目标函数可以被写为 $J_K=Tr(X^TX)-e^TX^TXe/n-Tr(Q_{k-1}X^TXQ_{k-1})$,(17)

注意到 J_k 不能区分原始数据和中心化过的数据。重复上面的推导我们由：

$J_K=Tr(X^TX)-Tr(Q_{k-1}X^TXQ_{k-1})$,(18) 第一项是常数。优化问题变为求 $\max Tr(Q_{k-1}^TY^TYQ_{k-1})$,(19)

满足 15，16 并且对于 q_k 满足 12。如果我们忽略最后一个限制，也就是让 h_k 可以取连续的值，期间还保持 15，16 的约束，最大化的问题可以通过以下结果解决。

定理 3.1. 当对 K 均值目标函数进行优化时，变换后的离散聚类成员指示向量的连续解 Q_{k-1} 是 k-1 主成份： $Q_{K-1}=(v_{1,..},v_{k-1})$。 J_K 满足上限和下限公式：

$$n\bar{y}^2-\sum_{k=1}^{K-1}\lambda_k<J_K<n\bar{y}^2,(20)$$

其中 $n\bar{y}^2$ 是所有的方差而 λ_k 是协方差矩阵的特征值。

注意到方程 16 的限制自动满足了，因为 e 是 Y^TY 的 $\lambda=0$ 对应的特征向量并且和其他的特征向量是正交的。这个结果对于任何 K 都是成立的。

这个定理的证明是 Ky Fan 理论的直接应用，为了优化问题 19。

定理 3.2 (Fan)让 A 是一个对称矩阵，有特征值 $\zeta_1\geqslant...\geqslant\zeta_n$ 并且对应特征向量是 $(v_1,..,v_n)$ 。那么 $Tr(Q^TAQ)$ 最大值满足限制 $Q^TQ=I_K$ ，有解 Q=$(v_1,..,v_k)$ R，其中 R 是一个任意的 k*k 的一个正交矩阵，并且 $max\ Tr(Q^TAQ)=\zeta_1+...+\zeta_K$ 。

方程 11 最先在（Godrdon & Henderson，1977）中以稍微有点不同的方式提出，并且很快被搁置了。它被独立发现于（Zha et all., 2002），其中光谱放松约束技术被应用在 11 而不是 18。导致 X^TX 的 K 个主成份特征向量是连续解。提出的方法由三个优点：（a）直接在 h_k 上对方程 18 进行放松并不和在方程 18 中的 q_k 上进行放松一样理想。这是因为 q_k 满足和为 0 的性质，这个通常的主成份是一样的，而 h_k 则没有这个很好的性质。离散的指示向量 q_k 的每个条目同时含有正数和负数，这个更加接近连续解。另一方面，离散指示向量 h_k 的每个条目只有一个符号，而这时所有 X^TX 的特征向量（除了 v1）同时含有正数和负数。换句话说， h_k 连续解会和离散形式的解由很大的出入，而 q_k 则会更加接近它的离散形式。（b）目前的方法和 K>2 和 K=2 情况都是一致的如果使用单一的指示向量。k=2 时，对于方程 11 的放松需要两个特征向量，这和在第二章中单个指示向量不一致。（c）方程 11 的放松使用原始数据，X^TX ，而现在使用的是中心化的矩阵 Y^TY 。使用这个中心化的矩阵使得方程 15，16 的正交性得以抱持一致性因为 e 是 Y^TY 的一个特征向量。另外，Y^TY 和 YY^T 是密切相关的。

群集质心子空间识别

假设我们已经找到了 K 聚类，每个聚类的中心是 m_k 。聚类间离差阵 $S_b=\sum_{k=1}^{K}n_k\boldsymbol{m_k}\boldsymbol{m_k^T}$ (总和为 0)，将任何向量 x 映射到 K 中心点张开的子空间中：$S_b^T\boldsymbol{x}=\sum_{k=1}^{K}n_k(\boldsymbol{m_k^Tx})\boldsymbol{m_k}$ 我们将这个子空间叫做中心群集子空间。从定理 3.1，我们有

定理 3.3 群集中心子空间被开始的 k-1 个主要方向张开，也就是

$S_b=\sum_{k=1}^{K-1}\lambda_k \boldsymbol{u_k u_k^T}$ 。

证明：一个聚类中心 m_k 可以用聚类指示向量来表示，$m_k=(1/n_k)\sum_{i\in C_k} y_i=n_k^{-1/2}\sum_i h_k(i)\boldsymbol{y_i}=n_k^{-1/2}Yh_k$ 这样 $S_b=\sum_{k=1}^{K} Yh_k h_k^T Y^T=Y\sum_{k=1}^{K} h_k h_k^T Y^T=Y\sum_{k=1}^{K} q_k q_k^T Y^T$ 现在，使用定理 3.1，q1，qk-1 用过 $v_{1,}\ldots,v_{k-1}$ 给出，并且 q_k 通过 $e_1/n^{1/2}$ 给出。这样 $\sum_{k=1}^{K} q_k q_k^T \rightarrow ee^T/n+\sum_{k=1}^{K-1} v_k v_k^T$ 。注意到 Ye=0 因为 Y 包含中心化的数据。使用方程 1，我们由 $Xv_k=\lambda_k^{1/2}u_k$ 。这就完成了证明。

定理 3.3 表明了 PCA 维数缩减自动会找到聚类中心子空间，也是差别最大的子空间。这个事实解释了 PCA 维数缩减是确实对 K 均值聚类有好处的因为在群集子空间中聚类过程比原始空间更加有效。

命题 3.4 在聚类子空间，聚类间的距离和原始空间中距离接近，而类内的距离减少了。

证明：将 y_i,y_j 之间的距离写成 $\|y_i-y_j\|_d^2=\|\hat{y}_i-\hat{y}_j\|_r^2+\|\bar{y}_i-\bar{y}_j\|_s^2$ 其中 $\bar{y}_i$ 是 r 维群集空间中的成员而 $\hat{y}_i$ 则是 s 维空间中的成员并且互不相关（$d=s+r$ ）我们希望证明

$$\frac{\|\bar{y}_i-\bar{y}_j\|_r}{\|y_i-y_j\|_d}\approx\begin{cases}1 & i\in C_k, j\in C_l\neq C_k \\ r/d & i,j\in C_k\end{cases} \quad (21)$$

如果 y_i,y_j 是在不同的类中，y_i-y_j 从一个类另一个类，或者是从一个中心点到另一个中心点。这样就几乎是在群集子空间的内部，也证明了 21 的第一个等式。如果 y_i,y_j 在相同群集，我们假设数据具有高斯分布。在概率为 r/d 时，y_i,y_j 指向群集子空间中的一个方向，在 pca 投影后会被保留。在概率为 s/d 时，$\bar{y}_i-\bar{y}_j$ 指向群集子空间外的一个方向，这会变成 0，即 $\|\hat{y}_i-\hat{y}_j\|\approx 0$ 。这证明了 21 中的第二个等式。

等式 21 说明了在群集子空间中，类间的距离保持为 1 个常数；而类内距离会缩减：群集变得更加密集。群集子空间的维数 r 越低，那么群集就越紧密，并且 K 均值聚类更加有效。

当映射到子空间时，子空间表达方式根据正交变换的矩阵 T 变换。因为 K 均值聚类方法是关于 T 不变的，所以我们不必关心 T 的具体形式。

总而言之，通过 pca 压缩维数群集子空间自动识别能保证 K 均值聚类在 PCA 子空间中表现效率非常的高。

核 K 均值聚类和核 PCA

从等式 9 中，K 均值聚类可以被看成使用 Gram 矩阵。这样它可以轻松地应用于其他的核（Zhang & Rudnicky, 2002）。它使用非线性变换将其映射到高维空间 $x_i \rightarrow \Phi(x_i)$

聚类目标函数在这种映射下，配合等式 9，可以写成

$$min\, J_K(\Phi)=\sum_i \|\Phi(x_i)\|^2-\sum_k \frac{1}{n_k}\sum_{i,j\in C_k}\Phi(x_i)^T\Phi(x_j), (22)$$

第一项是一个常数所以可以忽略。那么聚类问题可以变成求目标函数的最大值：

$$J_k^W=\sum_k \frac{1}{n_k}\sum_{i,j\in C_k} w_{ij}= Tr\, H^T W H= Tr\, Q^T W Q$$

其中 W= （w_{ij}） 是核矩阵， $w_{ij}=\Phi(x_i)^T\Phi(x_j)$

利用核均值聚类我们可以更加复杂的描诉数据的分布而不是单单的采用高斯分布。而不利的一面是不存在中心点因为仅仅只有成对的核或者是相似点。快速 n 阶局部修正不能再应用。

PCA 已经已经成功被应用在核矩阵中。非线性变换的一些好处已经体现出来了。K 均值聚类和 PCA 之间共同点在这里可以被扩展。注意到考虑一般性，一个核矩阵可以不用中心化因为数据已经中心化了。我们中心化的核用 $W \leftarrow PWP$，$P=I-ee^T/n$ 。因为等式 16， $Q^T WQ= Q^T PWPQ,$ 这样 23 等式抱持不变。中心化的核拥有所有特征向量满足 $q_k^T e=0$ 。我们假设数据和核是中心化。现在重复之前的分析，我们可以看到核 K 均值的解由核 PCA 提供。

定理 3.5 K 均值离散的聚类成员指示向量连续解是 k-1 核 PCA 主成份，并且 $J_K^W(opt)$ 满足下界： $J_K^W(opt)<\sum^{K-1}\zeta_k, (21)$

其中 ζ_k 是中心化的核 PCA 矩阵 W 的特征值。

恢复 K 聚类

一旦 K-1 个主成份 q_k 计算完了，怎么去恢复非负聚类指示向量 h_k ,然后聚类本身呢？

很明显，因为 $\sum_i q_{(i)}=0$,每个主成份都有很多负的元素，所以它们和非负的指示向量非常的不同。这样的话问题的关键就是计算正交变换矩阵 T。

一个 k*k 的正交变换等同于在 k 维空间中的一次旋转；由 k*k 个元素并且 k(k+1)/2 限制（Goldstein，1980）；剩下的 k(k-1)/2 自由度能很方便的表示为欧拉角。对于 K=2，一个简单的旋转 ψ 表示变换；对于 K=3，欧拉角 (φ,θ,ζ) 决

定了旋转。在 K 均值问题中，我们要求 T 的最后一列需要满足等式 13；所以真实的自由度是 $F_k=K(K-1)/2-1$ 。对于 K=2，Fk=0 并且解是固定的；由 14 给出。对于 K=3，Fk=2 我们需要在 2D 平面内搜索以找到最优解，也就是找到能够将 q_k 变换到非负指示向量 h_k 的 T 矩阵。

使用欧拉角来表示高维空间的正交旋转在 K>3 的时候有特殊约束是及其复杂的。这个问题可以通过如下陈述来解决。给定任意的 K(K-1)/2 正数 α_{ij} 和为 0，即 $\sum_{1\leqslant i\leqslant j\leqslant n}\alpha_{ij}=1$ 并且是对称的 $\alpha_{ij}=\alpha_{ji}$ 。那么自由度是 K(K-1)/2-1，和我们的问题的自由度是相同的。我们构造一个 K*K 的矩阵：

$$\Gamma=\Omega^{-1/2}\bar{\Gamma}\Omega^{-1/2},\Omega=diag(\sqrt{n_1},...,\sqrt{n_k})$$

其中 $\bar{\Gamma}_{ij}=-\alpha_{ij},i\neq j;\Gamma_{ii}=\sum_{j,j\neq i}\alpha_{ij},(25)$

对于任意的 $x=(x_{1,}...,x_K)^T$ 。这样对称矩阵 r 是半正定的，并且有实数特征向量和对应的非负的特征值。很明显，13 中的 t_n 是 Γ 0 特征值对应的特征向量。其他 K-1 个特征向量是相互正交的。在一般情况下，Z 是非奇异的并且 $Z^{-1}=Z^T$ 。这样 Z 是就是我们想要的正交变换矩阵 T。总结下来我们有：

定理 3.3 线性变换 T（12）由 Γ 的 K 个特征向量（25）组成。

这个结果表明 K 均值聚类将简化为 K(K-1)/2-1 参数的优化问题。

连结性分析

在上面分析中我们给出了 T 的结构。但是在我们知道聚类结果之前，我们不能计算 T 并且这样也就不能计算 h_k 。因此我们需要一个方法略过 T。

在定理 3.3 中，通过聚类中心点张开的聚类子空间由开始的 K-1 个主方向确定，也就是协方差矩阵 Y^TY 低维光谱表示。对于子空间的投影，相关系数 λ_k 是不重要的。增加一个常数矩阵 ee^T /n，我们有： $C=ee^T+\sum_{k=1}^{K-1}v_kv_k^T$

接着 3.3 的证明，C 可以被写成： $C=\sum_{k=1}^{K}q_kq_k^T=\sum_{k=1}^{K}h_kh_k^T$ ，而后面一项由明显对角结构，这就导致了相关性的解释：如果 $C_{ij}>0$ 那么 x_i,x_j 将会在同一个结构中，这样我们说它们是相连接的。我们进一步将 i，j 相连接的概率表示为 $p_{ij}=c_{ij}/c_{ii}^{1/2}c_{jj}^{1/2}=\delta_{ij}$ ，它取决于它们是否是相连接的。对角块结构是典型的聚类特征。

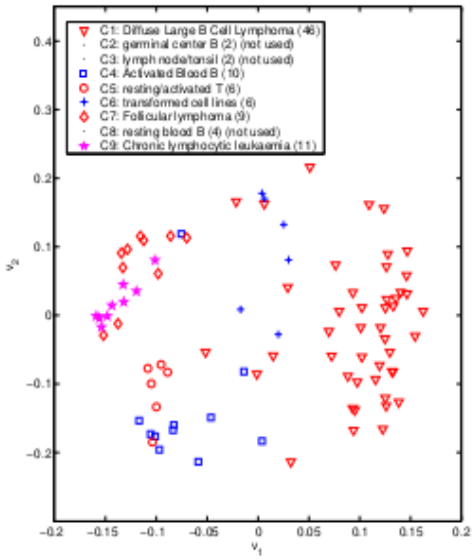
如果数据有明显的聚类结构，我们期望 C 拥有类似的对角块的结构，加上一些噪声，这是由于事实上主成份是离散指示向量的近似。例如，C 可以包含负数。这样我们将 $c_{ij}=0$ 如果 $c_{ij}<0$ 。并且，C 中的较小的正数表示很弱，可能是虚假的相连接的，这些应该被质疑。我们设定： $c_{ij}=0\,if\,p_{ij}<\beta,(26)$

其中 $0<\beta<1$ ，并且我们选择 $\beta=0.5$

一旦 C 计算过了，块结构就可以被观察到如果使用光谱排序（Ding & He, 2004）。通过对聚类交叉，群集分布在特定顺序上，一个 1D 的曲线展示了聚类结构。聚类可以通过线性化的赋值确定。

4.实验

基因表达



图像 2. 人类基因表达（Alizadeh et al., 2000）前两个主成份。

4029 个淋巴组织基因表达式从 Alizadeh et al.获得。使用生物和诊断专业技术，他们将其分为 9 个类别。因为大量的分类和非奇数数量的样本，它其实相对而言就是一个聚类问题。为了减少维数，4029 个基因中的 200 被选中基于 F 分布为了这次研究。我们专注于 6 个最大的类别，用了最少 6 个组织样本对于每个类别来表示每个类。类别 C2，C3 和 C8 因为样本数量太少而被忽略了。使用 PCA，我们划出了前两个主成份的图 2。

根据定理 3.1，聚类结构嵌在前 K-1=5 个主成份中。在这 5 维特征空间我们应用 K 均值聚类。聚类结果由以下混合矩阵给出：

$$B=\begin{bmatrix} 36 & . & . & . & . & . \\ 2 & 10 & . & . & . & 1 \\ 1 & . & 9 & . & . & . \\ . & . & . & 11 & . & . \\ . & . & . & . & 6 & . \\ 7 & . & . & . & . & 5 \end{bmatrix}$$

其中 b_{kl} 等于被分进类 K 的样本数量，但是正常应该属于类 L 的。分类准确度由式： $Q=\sum_k b_{kk}/N=0.875$ 计算得到，非常的合理对于这类困难的问题。为了给出这个结果的理解，我们进行 PCA 连结性的分析。聚类链接矩阵 P 在图 3 中显示。明显 5 个小的类有很强的连结性，最大的类和其他的 5 个由微妙的链接关系。这解释了分类的结果，B 的第一列。C1 被分为很多类别，另外 C5 的一个样本组织和 C4 有很大连结关系因此被分到 C4 中，B 的最后 1 列。

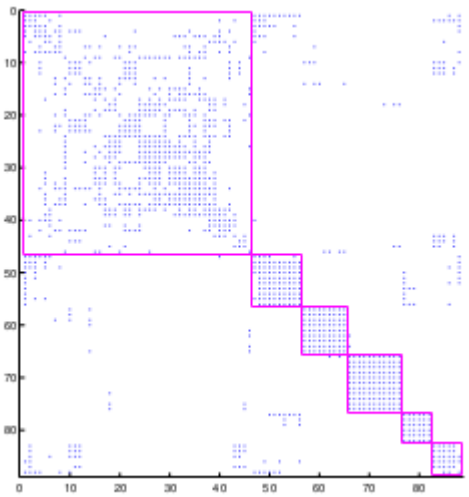


图 3 淋巴组织连结矩阵。6 个类别按顺序为 C1，C2，C3，C4，C5，C6

互联网新闻组

我们将 K 均值聚类方法应用在新闻组文章上。1 个 20 组数据集从 www.cs.cmu.edu/afs/cs/project/theo-11/www /naive-bayes.html. 获得。1000 个单词被选择根据单词和文档之间无监督的相互信息。使用标准的 tf.idf 术语加权。每个文档都规范化到 1。我们注意到在两个集合结合和 5 个集合结合。这个 4 个新闻组合如下表：

Dim	A5-B	A5-U	B5-B	B5-U
5	0.81/0.91	0.88/0.86	0.59/0.70	0.64/0.62
6	0.91/0.90	0.87/0.86	0.67/0.72	0.64/0.62
10	0.90/0.90	0.89/0.88	0.74/0.75	0.67/0.71
20	0.89	0.90	0.74	0.72
40	0.86	0.91	0.63	0.68
1000	0.75	0.77	0.56	0.57

```
NG1:  alt.atheism                NG18: talk.politics.mideast
NG2:  comp.graphics              NG19: talk.politics.misc
      A5:                              B5:
NG2:  comp.graphics              NG2:  comp.graphics
NG9:  rec.motorcycles            NG3:  comp.os.ms-windows
NG10: rec.sport.baseball         NG8:  rec.autos
NG15: sci.space                  NG13: sci.electronics
NG18: talk.politics.mideast      NG19: talk.politics.misc
```

表格 2. 聚类准确度，从原来的 1000 维进行 PCA 维数压缩

在 A2 和 A5，聚类中等程度的相互交叠。在 B2 和 B5，重叠的非常充分。为了累计足够多的统计数据，对于每个新闻组的结合，我们产生 10 组数据集，每一个都是随机的文档样本。细节如下，对于 A2 和 A4，每一个分类有 100 个文档随机从新闻组抽取；对于 A5 和 B5，我们让类大小变化来重建更多真实的数据集。对于平衡的情况，我们从每个新闻组抽取 100 个。对于不平衡的情况，我们分别取 200，400，120，100，60 个文档从不同的新闻组中。通过这种方式，我们产生了 60 个数据集，我们要在这上面作聚类分析。

我们首先通过获得下界（之前推导的）。对于每一个数据集，K 均值聚类我们作 20 次运作，每次从不同的随机开始，随机的选取数据点作为初始数据中心点。我们通过选取目标函数最小的聚类结果作为最终结果。对于每个数据集，我们都要计算核矩阵（中心化和非中心化的）主成份特征值。

表格 1 给出了 K 均值目标函数值和计算边界。从 km 开始的行是 J_k 对于每个数据样本的优化值。行 P2 和 P5 是下界。行 L2a，L2b 是早期工作的下界（Zha et al., 2002）L2a 是对于原始数据的，L2b 对应中心化的数据。最后一列是在下界和优化值之间的平均不同百分比。对于数据集 A2 和 B2，重新推导的下界比原来推导的更加接近优化的 K 均值。

从 60 个随机样本中新导出的下界一致地给出相近的 K 均值优化值。对于 K=2 的情况，下界是 0.6%在优化值之内。对于类的数量增多，下界会更加松散，但是会在 1.4%以内。

PCA 压缩和 K 均值

接下来我们在 PCA 子空间应用 K 均值分类。这里我们将数据从 1000 分别压缩到了 40，20，10，6，5 维。聚类精确度在 10 组样本中在表 2 中列出来了。为

了看到中心化和非中心化的微小的差别，表格左边是中心化的，右边是非中心化的。

两个观测点（1）从表格 2，明显可以看出维数被压缩了，结果是对称的并且显著提升了。例如对于数据集 A5 平衡的情况，分类精度达到 75%（1000 维）到 91%（5 维）（2）对于非常小的维数，PCA 基于中心化的数据能导致更好的结果。这些和之前的理论是相一致的。

讨论

传统的数据压缩观点都是从作为最好的一组双线性估计（SVD of Y）PCA 由来。最新的结果表明主成份是类内指示向量的连续解。这两种 PCA 的看法实际上是一致的因为数据分类也是数据压缩的一种形式。标准的数据压缩（SVD）发生在欧几里德空间，而分类是在分类空间中进行的数据压缩（数据点在同群集中被认为是数据同一类而反之亦然）这最好的解释了通过向量量子化在信号处理中的广泛使用而高维空间的信号特征向量被分解为 Voronoi cells 通过 K 均值算法。信号特征向量通过聚类中心点进行估计。PCA 在数据压缩和提供统一的方向扮演了重要的角色。

感谢

这个项目由 U.S.Department of Energy Office of Laboratory Policy 和 Infrastructure 支持

，通过 一个 LBNL LDRD，在协议 DE-AC03-76SF00098 下。

Table 1. K-means objective function values and theoretical bounds for 6 datasets.

	Datasets: A2										
Km	189.31	189.06	189.40	189.40	189.91	189.93	188.62	189.52	188.90	188.19	—
P2	188.30	188.14	188.57	188.56	189.10	188.89	187.85	188.54	187.91	187.25	0.48%
L2a	187.37	187.19	187.71	187.68	188.27	187.99	186.98	187.53	187.29	186.37	0.94%
L2b	185.09	184.88	185.63	185.33	186.25	185.44	185.00	185.56	184.75	184.02	2.13%
	Datasets: B2										
Km	185.20	187.68	187.31	186.47	187.08	186.12	187.12	187.36	185.51	185.50	—
P2	184.44	186.69	186.05	184.81	186.17	185.29	186.13	185.62	184.73	184.19	0.60%
L2a	183.22	185.51	184.97	183.67	185.02	184.19	184.88	184.50	183.55	183.08	1.22%
L2b	180.04	182.97	182.36	180.71	182.46	181.17	182.38	181.77	180.42	179.90	2.74%
	Datasets: A5 Balanced										
Km	459.68	462.18	461.32	463.50	461.71	462.70	460.11	463.24	463.83	463.54	—
P5	452.71	456.70	454.58	457.61	456.19	456.78	453.19	458.00	457.59	458.10	1.31%
	Datasets: A5 Unbalanced										
Km	575.21	575.89	576.56	578.29	576.10	579.12	579.77	574.57	576.28	573.41	—
P5	568.63	568.90	570.10	571.88	569.51	572.26	573.18	567.98	569.32	566.79	1.16%
	Datasets: B5 Balanced										
Km	464.86	464.00	466.21	463.15	463.58	464.70	464.45	465.57	466.04	463.91	—
P5	458.77	456.87	459.38	458.19	456.28	458.23	458.37	458.38	459.77	458.84	1.36%
	Datasets: B5 Unbalanced										
Km	580.14	581.11	580.76	582.32	578.62	581.22	582.63	578.93	578.27	578.30	—
P5	572.44	572.97	574.60	575.28	571.45	574.04	575.18	571.76	571.16	571.13	1.25%

表格 1 K 均值目标函数值；6 个数据集的理论下界

综合论文训练记录表

学生姓名	张政	**学号**	2010011426	**班级**	自 03
论文题目	主成分分析在引力波数据分析中的应用				
主要内容以及进度安排	**指导教师签字：** **考核组组长签字：** **年　　月　　日**				
中期考核意见	**考核组组长签字：** **年　　月　　日**				

指导教师评语	指导教师签字：_____ 年　　月　　日
评阅教师评语	评阅教师签字：_____ 年　　月　　日
答辩小组评语	答辩小组组长签字：_____ 年　　月　　日

总成绩：_____

教学负责人签字：_____

年　　月　　日