山西大学

2012 届硕士学位论文

基于支持向量机的主动学习方法研究

作者姓名　白龙飞

指导教师　王文剑　教 授

学科专业　计算机软件与理论

研究方向　机器学习

培养单位　计算机与信息技术学院

学习年限　2009 年 9 月至 2012 年 6 月

二〇一二年六月

Thesis for Master's degree, Shanxi University, 2012

Research on Active Learning Approach Based on Support Vector Machines

Student Name	Long-fei Bai
Supervisor	Prof. Wen-jian Wang
Major	Computer Software and Theory
Specialty	Machine Learning
Department	School of Computer and Information Technology
Research Duration	Sep., 2009-June,2012

June, 2012

目 录

Contents

中 文 摘 要

支持向量机（Support Vector Machine，SVM）是一种性能优良的学习器，作为一种有监督的机器学习方法，为了获得较好的泛化能力，在训练 SVM 前，必须有足够多的带标记样本来建立训练集，但在一些实际应用中如垃圾邮件过滤、医学图像检测等，往往很难获取到足够多的带标记样本。主动学习（Active Learning）是一种完全利用未标记样本进行学习的方法，通过迭代将未标记样本中最有价值的部分样本交由专家标记，并由此获得训练集。将其与 SVM 结合后，能很好的解决上述问题。本文以 SVM 为基准学习器，采用主动学习方法，针对不同维度的数据提出了相应的 SVM 主动学习算法。本文的研究内容总结如下。

（1）针对低维数据提出基于距离的 SVM 主动学习策略，称为 Dix_SVMactive。此算法主要通过计算未标记样本与当前超平面的距离、与前几次迭代中获取的已标记样本的距离，来度量样本的价值，进而判定此未标记样例是否需要交由专家标注。

（2）针对高维数据提出基于向量余弦的 SVM 主动学习策略，称为 Cos_SVMactive。此算法主要通过计算当前分类间隔内的未标记样本与当前的已标记样本夹角的余弦值，来度量高维样本的价值，从而决定此样本是否需要交由专家标注。

（3）Dix_SVMactive 和 Cos_SVMactive 都是先利用聚类算法对给定的未标记样本集进行粒化，选取与各类的类中心相关度最大的样本初始化训练集，并训练得到初始分类器。然后通过两种方法定义的样本置信度度量来挑选最有价值样本进行人工标注，并在每次迭代中对训练集的平衡度进行调整，以获得更好的泛化能力。另外，Cos_SVMactive 算法中还采用了新的迭代停止条件。

（4）在多个 UCI 标准数据集上分别对上述两个算法进行了一系列实验。实验结果表明，与基于随机选样（Random Sample）的 SVMactive 和传统 SVMactive（Tong SVMactive）方法相比，Dix_SVMactive 可以提高算法的分类精度；Cos_SVMactive 不仅具有较好的泛化性能，而且具有很好的收敛性。另外在某些数据集上，Cos_SVMactive 在保证具有较高精度的同时可大大减少算法的训练时间。

主动学习是一个当今比较热门的研究领域，将 SVM 与主动学习结合起来能够解

决更多的实际问题。本文工作所取得的成果不仅可以丰富支持向量机的理论与应用研究，而且可以拓宽支持向量机和主动学习的应用范围，具有重要的理论意义和应用价值。

关键字：支持向量机；主动学习；样本置信度度量；平衡度调整

ABSTRACT

Support Vector Machine (SVM) is one of the excellent learners. As a supervised machine learning method, to obtain better generalization performance, sufficient training labeled samples should be created before training the SVM. However, it is often difficult to obtain enough labeled samples in some practical problems, such as spam filtering, medical image detection and so on. Active learning is a method which completely utilizes unlabeled samples to learn. Some "most valuable" samples will be labeled by experts iteratively, and then they will be regarded as training samples. Combining active learning with SVM, many practical problems can be solved efficiently. This thesis proposes two SVM active learning approaches for different dimensional data, employing SVM as a basic learner, and adopting active learning. The major research contents are summarized as follows:

(1) For the low-dimensional data, an SVM active learning strategy based on the distance is proposed, called Dix_SVMactive. This algorithm measured the value of the samples by two distances, which between the unlabeled samples and the current hyper plane, the unlabeled samples and the labeled samples. Then the unlabeled most valuable samples will be labeled by experts.

(2) For high-dimensional data, an SVM active learning strategy based

on vector cosine is presented, named Cos_SVMactive. This algorithm measured the value of high-dimensional samples by calculating the cosine between unlabeled and the labeled samples. Then, some unlabeled samples which are most valuable will be labeled by experts.

(3) Both Dix_SVMactive and Cos_SVMactive are all used clustering algorithm to granulate a given set of unlabeled samples. The initialize training set includes some samples which have the highest relation degrees with the class centers of various types of samples. Then the SVM is trained and the initial classifier is obtained. The most valuable samples are selected for manually labeling based on sample confidence levels defined by the two approaches, and the balance of the training set is adjusted to get a better generalization performance after per iteration. In addition, a new iteration stop condition is also provided by the Cos_SVMactive algorithm.

(4) A series of experiments on UCI standard data sets are completed by these two algorithms, respectively. Experiment results demonstrated that, comparing with the traditional SVMactive (Tong SVMactive) and the SVMactive approach based on random sampling, the two proposed algorithms can improve the classification accuracy. The Cos_SVMactive approach has not only a good generalization performance, but also a good convergence. Additionally, for some date sets, the Cos_SVMactive can ensure a higher precision, and at the same time reduce the training time of the algorithm significantly.

At present, the active learning has become a hot research issue. Combining the SVM with active learning will solve more practical problems. The results achieved by this work can not only enrich the theory and application research of SVM, but also expend the application range of the SVM and active learning. Therefore, the work possesses important theoretical and practical values.

Key words: Support Vector Machine; Active Learning; Measure of Sample Value; Balance Adjusting

第一章 引言

本章主要说明研究工作的背景、研究的目的和意义，对相关领域国内外研究的现状作了简要介绍，同时提出本文要解决的问题，阐述了研究的方法。

1.1 研究背景

SVM 是基于统计学习理论（Statistical Learning Theory）[1]提出来的一种学习方法，作为一种成熟的学习器，因其稳定优良的学习性能而广泛地应用于各个领域，利用它来处理多数二分类问题时都能得到很好的效果。随着使用范围的不断扩大，SVM 面对的实际问题也越来越多样化，当使用传统的 SVM 处理某些问题时，往往无法获得足够的带标记样本，如自动过滤垃圾邮件是许多用户希望邮箱具备的一个功能，要想使用传统的 SVM 来过滤垃圾邮件，首先需要大量的已标注样本来建立一个训练集，而要想建立起这样一个训练集，不仅需要搜集成千上万的邮件，还要利用人力分别对其中的垃圾邮件和正常邮件一一进行标注，显然，这项工作所花费的代价是巨大的。因此，如何完全利用未标记样本来训练学习器具有重要的应用价值。

目前利用未标记样本进行学习的方法有很多，主动学习便是主流方法之一。与传统的学习方法不同，它在学习的初始阶段不需要任何带标记样本，是一种完全利用未标记样例来训练学习器的学习方法。“主动”体现在它可以基于某种“选样策略”，自动地从给定的未标记样本集中选出“最有价值”的样本，提供给专家进行标注，然后将其加入训练集训练学习器，如此循环往复，在多次迭代中不断改善学习器的性能。一般只要采用的选样策略合理，那么就能在达到同样或更好的学习效果的前提下，使训练集的规模更小，从而有效地减少人工标记样本所耗费的代价。

1.2 国内外研究现状

Vapnik 提出的 SVM 遵循结构风险最小化原则[2]，因而具有坚实的理论基础和良好的泛化能力。通过将原空间中的非线性可分问题映射到高维空间，并用核函数代替高维空间中的内积运算，对非线性分类问题可获得较好处理效果。核函数的使用，将 SVM 成功地推广到了回归问题、密度函数估计和函数拟合等领域。近年来 SVM 方法已经在很多领域如手写数字识别[3]、时间序列预测[4]等得到成功的应用，显示了它的优势。

在标准 SVM 的应用中已有一些研究成果。Osuna 等人改进了传统的 SVM 方法，使其对海量数据分类问题也能有很好的处理效果，最后文章通过在人脸图像识别[5]

数据库上的实验验证了算法的有效性。传统 SVM 寻找最优的分类超平面要通过求解一个二次规划问题来实现，但由于这样的二次型是相当稠密的，所以实际计算时对计算机内存的要求会随着样本集中样本个数的增多而呈平方级增长，因而带来的计算复杂度和时间复杂度是相当高的。Osuna 等人提出的分解算法，不仅能够保证得到的解为全局最优，而且大大减小了算法的计算代价和时间代价，使算法在处理数量较大的样本集时，同样能够取得很好的训练效果。这种分解算法的核心思想分为两部分：第一，对原问题进行分解，将其转变为若干个子问题，然后通过迭代解决子问题来找到原问题的解决方案；第二，提出对最优条件的评价，从而在迭代过程中不断地改进子问题的输出值，保证算法的收敛性。除此之外，算法还专门设置了停止准则，加强了算法运行的稳定性。在包含 50000 个样本的人脸检测数据库上进行了实验验证，并给出了一系列的实验结果，通过这些实验说明了 SVM 训练算法的可行性和优越性。

此外，卢增祥等人也在交互式支持向量机[6]方面提出了自己的方法。还有一些学者利用 SVM 压缩信息、分析样本、筛选因子和修复数据，同样取得了很好的效果，验证了 SVM 的有效性。

但不难发现，目前大部分 SVM 算法中，都需要有足够的带标记样本来保证其分类性能，而现实生活中往往很难做到这一点，这是 SVM 应用中不可避免的一个问题。主动学习[7]是一种区别于传统方法的学习思想，它利用的是更容易收集的未标记样本，主动学习领域的研究工作自从此概念提出后就获得了飞速的进展，涌现出了许多经典的算法。

Seung、Opper 和 Sompolinsky 提出了 Query by Committee 算法[8]，它包含两个子算法，即训练算法（training algorithm ）和询问算法（query algorithm）。训练算法负责使用专家标记过的样本训练学习器，询问算法负责从给定的未标记样本集中选择最有价值的样例作为下一个输入，将其交由专家标记并加入训练集。此算法先建立一个委员会，委员会中有 2k 个委员，每一个委员都是一个分类器，而且它们都是由同一个训练集训练产生的。询问算法认为最有价值的未标记样本 x 满足这样的条件，即有 k 个委员认为 x 属于正类样本，有另外 k 个委员认为 x 属于负类样本。也就是说，询问算法选择的是委员会最有争议的样本。当询问的次数趋于无穷时，算法收敛于一个稳定的值，从而使算法的泛化误差随样本数的增大呈指数级减小，实验说明这种方法具有很好的分类精度。类似的采用委员会对未标记样本的价值进行评判

的方法还有很多[9]。

Nguyen 和 Smeulders 在 2004 年发表的论文中，提出了主动学习与预聚类相结合的算法[10]，对二分类问题进行研究。传统的主动学习方法都是选择与分类边界间距离最短的未标记样本，Nguyen 等人经过分析发现，如果能将未标记样本集的先验分布知识加入到训练过程中，那么与传统算法相比，这种算法将有更好的学习效果。文中算法的核心思想就是通过对给定的未标记样本集进行聚类，来获取它的先验分布信息。算法主要包括两个步骤：首先对未标记样本集进行聚类，找出每个类的代表，将其加入训练集，训练分类器；然后用得到的分类器为剩余未标记样本做标记，从而确定当前数据集的噪声模型。这种选样方法不仅能选择到最有代表性的样本，而且避免了在不知道未标记样本集先验分布知识的情况下，选择同一类别中的样本加入了训练集。值得一提的是，算法在聚类步骤中，采用了由粗到细的聚类策略，这样既延续了大集群聚类在获取未标记样本集先验知识方面的优势，又兼顾到了对原未标记样本集特征的正确表示。作者在图像数据集上进行的实验表明他们提出的算法在各项指标上都优于其它算法，具有良好的泛化性能。

Lughofer 在 2012 年发表的论文中就减少人工标注未标记样本的工作量这一问题进行了深入研究，提出了一种混合主动学习模型[11]。这种模型的核心思想在于通过将无监督聚类方法和有监督增量学习方法整合到一起，来减少专家标注未标记样本所花费的代价。算法分为两个部分，第一部分通过无监督聚类找到信息量最大即最有价值的未标记样本交由专家标注，用来建立初始分类器；第二部分则在第一部分的基础上，通过在线学习模式，对于新输入的样本，使用基于置信度的主动学习方法度量其信息量，然后通过预先选定的增量学习法更新分类器。实验结果表明，与使用全部未标记样本进行训练的方法相比，此算法并没有损失分类精度；与随机选样的主动学习方法相比，该算法的学习效果和泛化性能更好。

除此之外，主动学习作为一种通用的机器学习思想，还可以与其它学习器结合。中科院的孙功星、戴贵亮[12]等人将主动学习应用到神经网络中，使其在样本集规模较大、信息冗余严重时仍能有很好的学习效果。北京科技大学的赵悦、穆志纯[13]也在这方面做了深入的研究，他们改进了传统的 QBC 算法，同时使用 KL-D 度量与投票熵度量样本的价值，同等训练条件下，改进后的 QBC 算法既选择了投票不一致的例子，又具有不低于被动学习的精度。

关于主动学习与 SVM 相结合（SVMactive）的研究，许多学者也做了很多工作。

Tong 和 Koller 使用一种新的主动学习方法与 SVM 相结合，来处理文本分类问题[14]。这种方法与以往的随机选择未标记样本训练学习器的方法不同，它首先定义了一个版本空间，其中的假设定义如下：

$$H = \{f \mid f(x) = (w \cdot \Phi(x))/\|w\| \quad \text{其中 } w \in W\} \tag{1.1}$$

上式中的参数集合$\mathcal{W}$等价于特征空间$\mathcal{F}$。所以版本空间$\mathcal{V}$定义如下：

$$\mathcal{V} = \{f \in \mathcal{H} | \forall i \in \{1 \cdots n\}\ y_i f(x_i) > 0\} \tag{1.2}$$

其中，$\mathcal{H}$是一系列分类超平面的集合，因为有

$$f(x) = w \cdot \Phi(x_i) \tag{1.3}$$

所以，式(1.2)也可写为

$$\mathcal{V} = \{w \in \mathcal{W} | \|w\| = 1,\ y_i(w \cdot \Phi(x_i)) > 0,\ i = 1 \cdots n\} \tag{1.4}$$

当下一个样本x_{i+1}输入后，有

$$\mathcal{V}_i^- = \mathcal{V}_i \cap \{w \in \mathcal{W} | -(w \cdot \Phi(x_{i+1})) > 0\} \tag{1.5}$$

$$\mathcal{V}_i^+ = \mathcal{V}_i \cap \{w \in \mathcal{W} | +(w \cdot \Phi(x_{i+1})) > 0\} \tag{1.6}$$

$\mathcal{V}_i^-$和$\mathcal{V}_i^+$分别表示下一个输入的样本x_{i+1}被标记为“-”和标记为“+”后生成的版本空间。算法要寻找的下一个输入就是能最快的缩减版本空间规模的未标记样本，最理想的情况就是下一个输入的样本总能二分当前版本空间，从而使算法的收敛速度最快，泛化性能最优。他们对提出的算法进行了有效性验证，实验结果表明该算法对文本分类问题有很好的效果。

中南大学和湖南公安高等专科学校的段丹青、陈松乔、杨卫平[15]在网络入侵检测中使用到了 SVM 主动学习方法，他们提出的算法每次都选择一个离超平面最近的样本进行人工标注，然后加入训练集，改善分类器性能。

另外，Mukerjee[16]、Schohn[17]、Lewis 与 William[18]、Vlachos[19]、周艳丽[20]、张健沛[21]、解洪胜[22]、Wu 与 Huang[23]等许多科研工作者都在这个领域做了深入和广泛的研究。总之，由于 SVMactive 具有很好的性能，它已广泛应用各个实际领域。

1.3 本文的研究内容

如 1.2 节所述，现有的 SVM 主动学习算法主要存在如下问题。

（1）大部分算法仅根据未标记样例与超平面间的距离来度量样本价值，这样有可能选到孤立点，降低算法的分类精度。

（2）使用距离来度量维度较高的样本的价值时，往往无法选到最合适的未标记样本，减弱了算法的泛化性能。

（3）主动学习在每一次迭代中都会生成对应的训练集，新生成的训练集与已有的训练集可能存在较为严重的信息冗余，影响了算法的收敛速度。

本文就以上三个问题进行研究。工作的关键在于设置一个合适的置信度来衡量样本的“价值”，目的在于既能使训练得到的学习器性能不弱于传统 SVM 和经典的 SVM 主动学习算法，又能减小训练集规模。

文中针对低维和高维数据分别提出了对应的算法。这两种算法先利用聚类粒化样本集，然后，

（1）对于低维数据，通过度量未标记样本与超平面的距离以及未标记样本与已有的已标记样本的距离，来判定此样本是否能够加入训练集。

（2）对于高维数据，通过计算与当前分类间隔内的未标记样本与已有的已标记样本夹角的余弦值，来度量该样本的价值。

经过数次实验发现，在训练集构造过程中，有时会出现样本偏斜的情况，此时分类器的性能会有所下降，本文通过减小多类数据的规模来解决此问题，具体实施时，采用基于聚类和基于向量余弦两种策略从多类数据中选择合适的样本加入训练集。

1.4 本文的结构安排

本文主要针对主动学习与 SVM 的结合进行初步的探讨和研究。对现有的 SVM 主动学习方法存在的问题提出解决方案，并提出新的 SVM 主动学习方法。本文结构安排如下。

第一章主要介绍本文工作的研究背景、国内外研究现状、本文结构以及文章研究的主要内容。

第二章主要介绍 SVM 以及主动学习的基础知识，包括 SVM 与主动学习的基本理论、现有的主要算法及应用领域等等。

第三章着重介绍一种新的基于 SVM 的主动学习策略。新策略采用了一种新的距离度量，并且对样本集合之间的信息冗余提出了解决方案，减少人工标注工作量的同时，提高了算法的精度。新方法的有效性也在多个 UCI 标准数据集上进行实验验证。

第四章详细介绍一种针对高维数据的 SVM 主动学习策略。这种策略将未标记样本看作向量，计算了未标记样本与已有的已标记样本夹角的余弦值，然后与未标记样本和当前超平面间的距离相结合，共同判定未标记样本的价值。这种方法可增强

SVM 主动学习方法的适用性，并在多个 UCI 标准数据集上进行实验，验证算法的收敛性和有效性。

第五章总结了本文所做的工作，提出了后续工作的重点和值得深入研究的方向，展望了 SVM 主动学习的前景。

第二章 预备知识

本章将对后续章节中涉及到的相关知识做简要介绍，包括 SVM 和主动学习的理论基础、原理和典型算法等等。

2.1 支持向量机

传统的统计模式识别方法有一个重要的前提，就是样本规模趋于无穷大，但实际条件下往往无法完全满足这样的条件。另外它单纯的强调经验风险最小化，这样极有可能产生过学习问题，即某些时候，如果过分最小化学习器的训练误差反而会使其推广能力大幅下降。因此它得出的模型对新的输入做出的预测可能是不正确的。统计学习理论就是针对这个问题提出来的。这就是 SVM 的理论基础。

SVM 是由 Cortes 和 Vapnik 于 1995 年提出的一种学习器。它与传统的模式识别学习方法不同，其研究对象是有限样本前提下的学习问题，而且统计学习理论背景下学习器的实际风险分为两部分，即置信范围和经验风险。后者是传统模式识别理论一贯强调的，前者才是 SVM 的优化目标，即 SVM 是一种基于结构风险最小化准则的学习器。显然，从理论层面上来说，SVM 的学习效果和推广能力要比传统方法好很多。

现实中的分类问题大致可分为二分类和多分类问题，对于这两类问题，SVM 都有很好的学习效果。此处以二分类问题为例，解释 SVM 的学习原理。

线性可分和线性不可分是二分类问题的两种类型。以线性可分的问题为例，给定训练集$X = \{x_1 \cdots x_n\}$, $X \in R^d$。对应的标记为$\{y_1 \cdots y_n\}$，$y_i \in \{1, -1\}$。设给定空间的维度为d，并设其线性判别函数为$g(x) = wx + b$。我们可以通过归一化使$\{x_1 \cdots x_n\}$满足$g(x) \geq 1$，此时，分类间隔等于 $2/\|w\|$ 。所以，当

$$y_i[wx_i + b] - 1 \geq 0, i = 1, \cdots, n \tag{2.1}$$

成立时，所有样本都通过此分类器获得了正确的类别标记。显然，当$\|w\|$最小时，分类间隔最大。所以，最优分类超平面必须同时满足式(2.1)和使$\|w\|$最小。支持向量就是令式(2.1)中等式成立的样本。

如上所述，求这样一个最优分类面的问题等价于求以下的约束优化问题：

$$\min \quad \|w\|^2/2 \tag{2.2}$$

$$\text{s.t.} \quad y_i[wx_i + b] - 1 \geq 0, i = 1, \cdots, n \tag{2.3}$$

显然式(2.2)和式(2.3)都是凸函数，这样就将 SVM 的求解最后转化成二次凸规划问题的求解，因此从理论上来说 SVM 的解是全局最小解，也是全局唯一的最优解。通

过 Lagrange（拉格朗日）乘子法可知：

$$f(x) = \mathrm{sgn}(w^* \cdot x + b^*) \tag{2.4}$$

即

$$f(x) = \mathrm{sgn}\left(\sum_{i=1}^{k} \alpha_i^* \cdot y_i \cdot (x_i \cdot x) + b^*\right) \tag{2.5}$$

式(2.5)中，$(x_i \cdot x)$表示两向量的内积，α^*和b^*为确定分类超平面的参数。

这就是对于线性分类问题的分类超平面求解函数。对于非线性分类问题，SVM 通过核函数对当前空间进行非线性变换，这样既不显著增加计算复杂度，又可以在变换后的空间中对原空间的非线性问题实现线性分类。这样，求解分类问题的函数成为：

$$f(x) = \mathrm{sgn}\left(\sum_{i=1}^{k} \alpha_i^* \cdot y_i \cdot K(x_i \cdot x) + b^*\right) \tag{2.6}$$

式(2.6)就是形式化描述下的 SVM。常用的核函数有以下几种：

（1）线性核函数：

$$K(x,\ x_i) = (x_i \cdot x); \tag{2.7}$$

（2）多项式核函数：

$$K(x,\ x_i) = [p(x_i \cdot x) + s]^q; \tag{2.8}$$

（3）Sigmoid 核函数：

$$K(x,\ x_i) = \tanh\left(\mu(x_i \cdot x) + c\right); \tag{2.9}$$

（4）径向基核函数(Radical Basis Function,RBF)：

$$K(x,\ x_i) = \exp\left(-\gamma|x - x_i|^2\right); \tag{2.10}$$

不同的分类问题中，可以采用不同的分类核函数，以期达到好的分类效果。

对于线性不可分的情况，在式(2.3)中添加松弛项$\xi \geq 0$，原优化问题从而变为：

$$\min \quad \|w\|^2/2 + C\sum_{i=1}^{n} \xi_i \tag{2.11}$$

$$\text{s.t.} \quad y_i[wx_i + b] - 1 + \xi_i \geq 0, i = 1, \cdots, n \tag{2.12}$$

其中，$C \geq 0$为一常数，称作惩罚参数。这样，优化目标就由原来的最大化分类间隔变为现在的折中考虑最大化分类间隔和最小化错分样本数。

由于 SVM 理论上具有优异的学习性能，因而 SVM 在中文信息处理、图像识别、样本分析和知识挖掘等许多领域有了许多成功的应用。

2.2 主动学习

Simon 于 1974 年最先提出主动学习这个概念[24]。它的核心是“自动”地选出最有价值的未标记样本，由专家对其进行标注，然后加入训练集训练分类器，最后通

过多次循环迭代来改善学习器的性能，相对而言，传统的学习就可以称为在已标注样本集上的被动学习。与传统学习方法不同的是，主动学习定义了样本价值的度量方法，如熵、不确定性等等，在迭代选样的过程中能够选择到对改善学习器性能贡献最大的未标记样例。因此主动学习不仅具有较高的分类精度，而且可减小训练集规模。

主动学习从功能结构上来看，有两大模块，即学习模块和选择模块。前者负责利用选择模块提供的样本对学习器进行训练；后者负责从众多的未标记样本中选出最有价值的样本，以供学习模块对给定数据进行学习。

从学习形式上来看主动学习是一个循环迭代的过程，其主要步骤为。

（1）随机或以某种策略从大量未标注样本中选择若干个样本进行人工标注，然后加入训练集，初始化分类器。需要注意的是，初始训练集中至少要有 1 个正类样本和 1 个负类样本。

（2）在上一步骤基础上，采用某种主动学习算法，从剩余的未标记样本中选出一部分“最有价值”的样本，交出专家标注，然后添加到训练集中，以便在下一次迭代中改善分类器性能。

（3）如果学习器的性能好于预期值或循环次数大于预先设置的阈值，则迭代停止；否则重复上述步骤。

按照未标注样例获取的方式不同，可以将现有的主动学习策略大致归为两大类。基于池的策略和基于流的策略[25]。基于池的策略又可以细分为以下四类。

（1）基于不确定度缩减的方法[25]

这种方法是主动学习中较为常用的未标记样例选择方法之一，它认为类别最不能确定的样例最有价值。这类方法中，衡量样本类别不确定度的方法有两种：第一种，从信息学角度看，样例的不确定度是与其信息熵的值成正比的，即样例的信息熵越大，其类别越不确定，反之亦然。而一个对象信息熵的大小与它所蕴含信息的多少成正比。所以依照此衡量标准选择的样本从理论上来说必然是蕴含信息量最大的样本。第二种；从几何学角度来看，离当前分类超平面越近的样本其标记越不确定。因为在超平面移动过程中，其附近样本的标记最有可能被改变，所以理论上样本到当前超平面的距离与其不确定度成正比。理论上的可行性使这类方法能够适用于各类学习器，诸如 SVM[14-23]、隐马尔可夫模型（Hidden Markov Model）[26]和逻辑回归[18]等等。

但是这类方法同样也存在着一些弊端。首先，如何合理的定义未标记样本的信息熵，是一个值得研究的问题；其次，如果只凭借未标记样本与当前超平面的距离来选择下一个输入的话，很有可能会选到孤立点；另外，当样本维度较高时，传统的欧氏距离往往不能正确的表示未标记样本与当前超平面间的几何关系，也无法与未标记样本的不确定程度正确对应。

南京理工大学的韩光、赵春霞、胡雪蕾[27]提出了一种新的 SVM 主动学习算法，使用其来处理障碍物检测问题。算法中对超平面位置的校正方法定义如下：

$$f'(x) = f(x) - f_0 \tag{2.13}$$

用 SVM 分类器标注所有未标记样本，并计算样本到当前超平面的距离值，并用正数表示正类样本对应的距离，负数表示负类样本对应的距离，最后依次按值从大到小排序，得到的新的样本序列如下所示：

$$x_1,\ x_2,\ \cdots,\ x_{s_{t-1}},\ x_{s_t},\ x_{s_{t+1}},\ \cdots,\ x_{n-1},\ x_n \tag{2.14}$$

其中，x_{s_t}对应的距离即为超平面校正幅度。在第 i 步迭代中计算式(2.13)中的f_0时，需要根据第 i-1 步迭代中专家对未标记样本给出的标记调整s_t的值。

$$s_t' = s_t + s_0 \tag{2.15}$$

$$s_0 = \Delta L = \sum_{j=1}^{n}\left(L_{expert}(x_j) - L_{SVM}(x_j)\right) \tag{2.16}$$

其中，$L_{expert}(x_j)$表示专家标记的结果，$L_{SVM}(x_j)$表示 SVM 标记的结果。

设$\{x_1 \ldots x_n\}$中$L_{expert}(x_j)$的值为正类而$L_{SVM}(x_j)$的值为负类的样本个数为a，设$L_{expert}(x_j)$的值为负类而$L_{SVM}(x_j)$的值为正类的样本个数为 b。当$\Delta L < 0$时，$a > b$，说明负类样本中心与当前 SVM 超平面之间的距离较远，所以此时应该调整超平面的位置使其靠近负类样本的中心；同理，当$\Delta L < 0$时，$a < b$，应该对 SVM 超平面的位置进行调整使其靠近正类样本的中心。

为了保证算法的收敛性，最后超平面移动的幅度s_0如下式所示：

$$s_0 = |c_0\omega(t)\Delta L| = \left|c_0\omega(t)\sum_{j=1}^{n}\left(L_{expert}(x_j) - L_{SVM}(x_j)\right)\right| \tag{2.17}$$

其中c_0为可调参数，$\omega(t)$为关于 t 的可调项。如上所述，超平面每次迭代都将专家标注结果与 SVM 标注结果进行对比，得到超平面调整幅度f_0，然后对超平面位置进行调整。

（2）基于版本空间缩减的方法[26]

在介绍此类方法前，先对版本空间的相关基础知识作简单的介绍。

设非空集合 H_S 为一个假设空间，集合X中的元素为给定的样例，集合Y中的元

素为样例对应的标签，从 X 映射到 Y 的函数都为H_S中的元素，训练集初始化完成后开始训练学习器。对H_S中的所有元素进行搜索，如果$\exists f(x)$，$\forall x \in X$，都有$f(x) = c(x)$成立，且$f(x) \in H_S$，则$f(x)$就是算法所要寻找的假设。设版本空间$V_S_{H_S,Train} \subseteq H_S$，它表示训练集为Train的假设空间的子集，$V_S_{H_S,Train}$满足以下条件:

$$\forall h \in V\_S_{H\_S,Train}, \forall x \in Train, h(x) = c(x) \tag{2.18}$$

$$\forall h \notin V\_S_{H\_S,Train}, \exists x \in Train, h(x) \neq c(x) \tag{2.19}$$

其中，$c(x)$为真实函数。

基于版本空间缩减的方法认为，能够在每次迭代中大幅减小当前版本空间规模的样本为最有价值的样本。对于二分类问题而言，传统的基于版本空间缩减的方法总是以能够二分版本空间的未标记样本为最有价值的样本。这类方法中具有代表性的算法有 Query by Committee[8, 28]和 Selecting the Most Possibly Wrong-Predicted Instances[29]等等。

同样，在使用此类方法选择未标记样本时，也存在以下问题。第一，版本空间的规模不能太大，否则会增加搜索的复杂度，但也不能太小，否则无法包含最优的假设；第二，传统的基于版本空间缩减的方法认为，在二分类问题中，每次都能二分版本空间是最理想的效果，但在实际情况下，不一定每次选择的样本都能二分版本空间；第三，在多分类问题中，每次选择的样本将版本空间缩减到什么程度才算是最优，仍值得商榷。

（3）基于泛化误差缩减的方法[26]

使用此类方法选择未标记样例时，要先选定一个函数$f(x)$作为损失函数，$f(x)$用于估计每个未标记样本使学习器对未来输入的样本分类错误的错误程度。然后对于当前未标记样本集中的每一个未标记样本x_i都使用$f(x)$估计出其对应的错误率，即若选择x_i作为下一个输入，那么它对应的学习器的泛化误差为$f(x_i)$。最后选择$f(x_i)$最小的未标记样本作为下一个输入。

这类方法的思想近似于贪婪算法，理论上具有好的学习效果，但实际情况下，第一，当未标记样本集规模很大时，算法的时间复杂度和计算复杂度是巨大的；第二，不同的基准学习器要选择不同的损失函数，同一种基准学习器也有不同的损失函数可供选择，所以，损失函数本身的精度以及对学习器泛化错误率估计的准确度直接影响了最后的学习效果，如果损失函数选择的不恰当，那么在未标记样本选择

的过程中就会出现很大偏差。

（4）其他方法[26]

还有一些常见的、较为典型的主动学习算法，诸如 COMB[30]算法、多视图学习[31]以及预聚类方法[32]等等。

COMB[30]算法是先选定三种不同的主动学习器，然后将其按照一定的方式组织在一起。在进行学习和训练时，实时监测各个学习器的学习效果，选中效果最好的那一个，然后快速转移到此学习器上进行学习。总而言之，COMB 算法总是在当前性能最优的主动学习器上进行学习。但这种方法最后的学习效果同样局限于所选的三种子学习器的学习性能。

在介绍多视图学习[31]前，先介绍一下视图的概念。所谓视图，就是指当前未标记样本集中能够对未来输入的未标记样本做出类别判定的特征集合。多视图学习就是针对此类多视图的问题设计的，它每次总选择这样的未标记样本 x，即多个视图对 x 的类别标注分歧是最大的。这种方法学习性能的优劣一定程度上取决于未标记样本集中包含的视图的多寡，如果较多，那么选到的样本会更合理一些，反之，选到的样本可能并不是最有价值的。所以，这种方法在低维数据集上的效果并不是很好。

预聚类方法[10]的关键在于先对未标记样本进行聚类，然后优先考虑类内中心和类间分界线附近的样本。这样做的好处是兼顾了未标记样本集原始的分布规律，缺陷是选择到的未标记样本是否最有价值一定程度上受制于所选择的聚类算法是否恰当。

基于流的方法与基于池的方法最大的不同之处在于，它要处理的未标记样本是逐次逐个提供的，也就是说在基于流的方法中，我们无法对未标记样本的价值进行整体的衡量和相互的比较，设计算法时只能先设定一个阈值，当选择模块接收到一个未标记样本时，先用某种方法估计出此样本的价值，然后与阈值比较，将达到条件的未标记样本提交给专家进行标注后供学习模块训练，将不符合条件的未标记样本丢弃。这类方法的缺陷很明显，首先，阈值的设置要合理，否则会直接影响学习器的性能；其次，不同的数据，不同的问题，要选择不同的阈值，哪怕是同一类问题，当数据的特性发生变化时，当前的阈值应马上做出调整，而对于如何调整阈值现在还没有成熟的理论和方法。所以，基于流的方法的应用还不是太广泛。

由上述可知，主动学习不论从理论上还是实际应用上，都具有独特的解决问题的思路和优良的学习性能，所以，其在障碍物检测[32]、图像检索[23]规则训练[33]等众

多领域都有广泛的应用。

2.3 本章小结

本章对 SVM 和主动学习各自的相关基础知识、基本原理、主流算法以及应用领域都做了介绍。主动学习近年来一直流行于机器学习领域，与 SVM 的结合也一直是一个热门的研究方向。前者是一种思想，后者是一种工具，将二者结合后既延续了主动学习的独特思路，又借助了 SVM 的优良性能，拓宽 SVM 应用领域的同时，使主动学习的功能得到了更大的发挥。

此页不缺内容

第三章 基于距离的 SVM 主动学习策略

Tong 较早就提出了 SVMactive 算法[7]，其采样策略是每次迭代选取一个离分类超平面最近的样本。通常认为，这些样本的类别最不确定，也最有可能被分错，因此信息量最大，所以它们最有可能改变超平面的位置，而远离超平面的样本对其位置的改善作用不大[34]。但如果仅以此为标准，可能会存在两个问题：①第 n 次迭代中选择的最有价值样本很可能会与第 n-1 次迭代中选择的样本产生信息冗余；②每次只考虑一个离超平面最近的样本，使得训练集的规模过小，无法及时获取无标记样本集的总体特征，从而影响收敛速度。虽然 SVMactive 算法所采用的一次标注一个样本的策略对文本分类问题能获得最好的分类性能[14]，但在其它应用问题中，由专家对样本进行逐次逐个标注是不现实的。上述这些问题都会影响训练过程的收敛速度以及训练效果。

本章提出了基于距离的 SVM 主动学习方法 Dix_SVMactive，其选样策略的核心在于通过未标记样本与当前超平面的距离和未标记样本与当前已标记样本的距离共同度量未标记样例的价值，并且对迭代过程中可能出现的训练集偏斜问题提出了解决方案。最后在多个 UCI 标准数据集上进行验证，并且与基于随机选样（Random Sample）的 SVMactive 和传统 SVMactive（Tong SVMactive）方法进行比较。

3.1 基于距离的 SVM 主动学习算法

算法将 SVM 分类器的训练过程看作一个迭代过程，每次迭代都从未标记样本中通过定义的置信度度量寻找最有价值的样本进行人工标注，然后加入 SVM 的训练集，一直迭代直到分类器的精度或循环次数达到某一阈值时停止。

3.1.1 基于距离的置信度度量方法

一般认为，样本距离超平面越近信息量越大，所以也越有价值。但主动学习是一个迭代的过程，因此收敛速度也是一个需要兼顾的指标。假设 L 和 U 分别表示已标记样本集与未标记样本集，本算法对每个未标记样本x_i的置信度定义如下：

$$c(x_i) = \overline{d^n}(x_i, x_j)/D^n(x_i)，\ x_i \in U, x_j \in L \tag{3.1}$$

其中 $\overline{d^n}(x_i, x_j)$ 表示第 n 次迭代中，未标记样本x_i与每个已标记样本x_j的欧氏距离的平均值，$D^n(x_i)$ 表示第 n 次迭代中未标记样本x_i与当前分类超平面的距离。显然，样本的价值与 $D^n(x_i)$ 的值成反比，同时由于样本信息可能会产生冗余，算法用 $\overline{d^n}(x_i, x_j)$ 衡量样本的冗余程度，即 $\overline{d^n}(x_i, x_j)$越大，冗余程度越小，此样本也就越有价值。

综上所述，$c(x_i)$ 越大，样本越有价值。每次迭代中，对所有未标记样本都计算对应的置信度$c(x_i)$，然后按降序排列，取前 m 个样本进行人工标记，然后加入训练集。

3.1.2 基于聚类的训练集平衡度调整策略

在每步迭代后得到的带标记样本集都有可能是非平衡的，即超平面可能与一类样本的中心距离较远，与另一类样本中心距离较近，此时，依照本文所提出的置信度度量选择的样本中，样本中心离超平面近的类中样本个数可能会大于另一类样本，如果不对数据集进行处理，则会使算法的学习能力下降。

为避免出现选择最有价值样本导致的数据不平衡现象，Dix_SVMactive 算法在每次迭代后都会检测样本集的平衡度 b，其定义如下：

$$b = \begin{cases} num^+/num^- & 若 num^+ \le num^- \\ num^-/num^+ & 否则 \end{cases} \tag{3.2}$$

其中，num^+表示正类样本的个数，num^-表示负类样本的个数。当 b 的值不大于ε（ε为一预先设定的参数）时，认为此集合是非平衡的，这时算法对多数类数据进行聚类（聚类个数为少数类样本数），然后仅将与聚类中心最靠近的多数类样本与少数类样本加入训练集，而将多数类数据中的其它样本删去，以此来消除集合的不平衡现象。

3.1.3 Dix-SVMactive 算法

假设已标记样本集用L表示，未标记样本集用U表示，初始化时令$L = \emptyset$，将所有的未标记样本$\{x_1 \dots x_n\}$加入集合U，即令$U = \{x_1 \dots x_n\}$；集合 Need_label 中的元素为每次迭代结束后选出的最有价值样本，其初始值为∅，且在每次迭代前都要清空；Train 中的元素为所有人工标记过的样本，用作 SVM 训练集，其初始值同样为∅。Wrong_label 中的元素为每次迭代中被分错的样本，其初始值也为∅，同样在每次迭代前都要清空。

基于以上假设，本章算法的主要步骤总结如下。

Step1 初始化。

将U中所有样本聚为k类，对应的类中心为$c_1,\dots,c_k$；然后将$c_1,\dots,c_k$交由专家标记，若$c_1,\dots,c_k$中分别含正负类，则令$Train = \{c_1,\dots,c_k\} \cup Train$，否则对原始数据聚k+1类。重复这一过程，直到类中心分别包含正负类样本为止。令$U = U - Train$。

Step2 循环t步，执行以下步骤。

Step2.1 用Train训练SVM，并对U中样本的类别进行预测。

Step2.2 对每个$x_i(x_i \in U)$，按公式(3.1)计算$c(x_i)$。然后按$c(x_i)$的值对U中样本进行降序排列，取前m个样本加入Need_label，并将Need_label中的样本交由专家进行标记。

Step2.3 将Need_label中各样本的标记结果与Step2.1中对应样本的标签进行对比，若二者不同，则将其放入Wrong_label。

Step2.4 按公式(3.2)计算当前Wrong_label集对应的b，若$b \le \varepsilon$，则按3.1.2节中的方法对Wrong_label的平衡度进行调整。

Step2.5 令$Train = Wrong_label \cup Train$，$U = U - Wrong\_label$。

Step2.6 如果循环次数达到预设值，则转Step3，否则，继续循环。

Step3 算法结束。

3.2 实验结果与分析

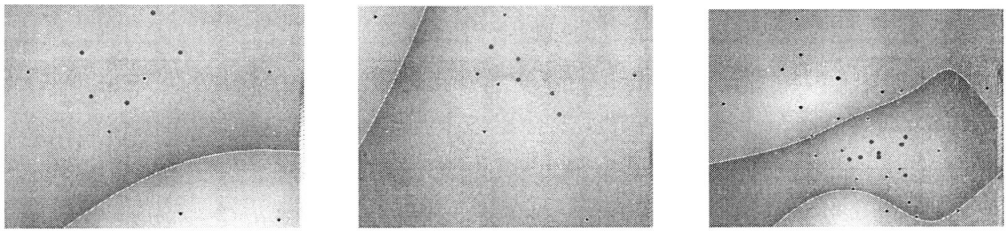
为了验证本章算法的有效性，在6个UCI标准数据集（见表3.1）上进行了实验。实验中采用高斯核函数，C取值为1000，σ取值为1.0。

表3.1 实验采用的数据集

数据集	训练集个数	测试集个数	数据维数
Banana	8800	1000	2
Thyroid	2800	1500	5
Diabetis	4680	3000	8
Breast_Cancer	2000	770	9
Image	6500	5050	18
German	3500	1500	20

3.2.1 Dix-SVMactive 的有效性验证及实验结果分析

在同等规模的训练集下，以 Thyroid 为例，分别对基于随机选样方法（Random Sample SVMactive）、传统 SVMactive（Tong SVMactive）以及本文算法（Dix_SVMactive）进行了比较。由于 Random Sample SVMactive 算法运行结果很不稳定，故在训练集规模相等的情况下，连续运行 10 次，将与平均值最接近的那次作为最终结果。三种方法得到的分类超平面如图 3.1 所示，测试结果比较见表 3.2。实验结果表明，本文提出的 Dix_SVMactive 算法具有较高的预测精度，且能保持较高的学习效率。在其它几个数据集上的实验也得到了类似的结论。



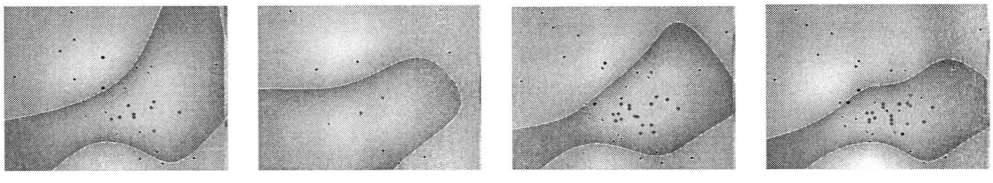
(a) Random Sample SVMactive　(b) Tong SVMactive　(c) Dix_SVMactive

图 3.1 三种方法在 Thyroid 数据集上得到的分类面

表 3.2 三种方法在 Thyroid 数据集上实验结果比较

方法	准确率(%)	运行时间(秒)
Random Sample SVMactive	82.6	1.672
Tong SVMactive	81.3	0.016
Dix_SVMactive	96.3	0.047

Dix_SVMactive 中有一个可调参数 m，它的值越大，意味着能选到更多的有价值样本，算法的精度也越大。从实验结果来看，当 m 的值增大到一定程度时，算法的精度不再有大幅度的提高。在 Thyroid 集上 m 取不同值时得到的分类超平面如图 3.2 所示，表 3.3 列出了 m 取不同值时对泛化能力的影响结果；其中#SV 表示最后一次迭代后获得的支持向量个数。



(a) m=10　(b) m=20　(c) m=30　(d) m=40

图 3.2 m 取不同值时本文算法在 Thyroid 集上得到的超平面

表 3.3 m值对算法结果的影响（Thyroid集）

m	10	20	30	40
#SV	9	20	23	32
准确率(%)	93.3	95.5	97.2	97.9

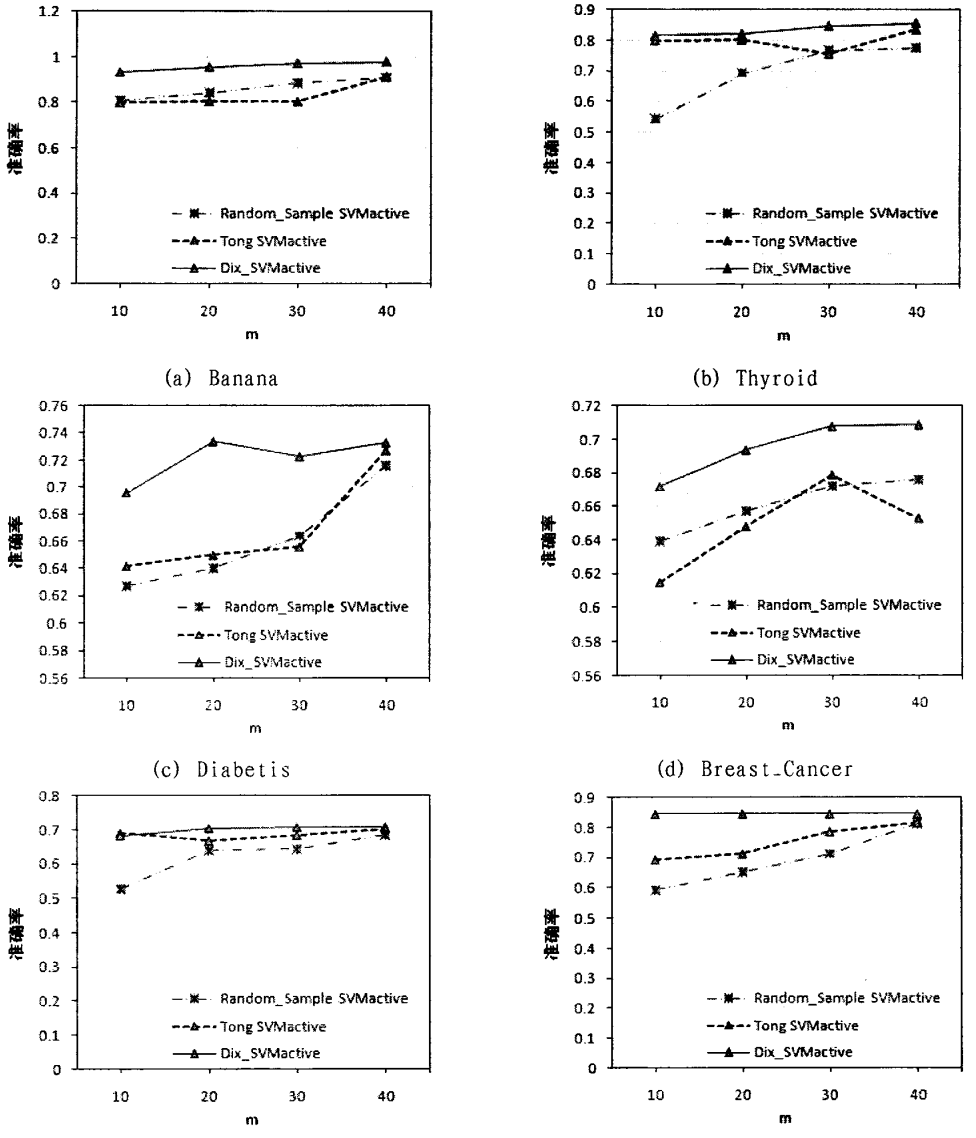
表 3.4 是 Dix_SVMactive 算法在其他数据集上进行测试的结果。

表 3.4 Dix_SVMactive 在其他数据集上的测试结果（ε=0.5）

数据集	测试结果	*m*			
		10	20	30	40
Banana	#SV	19	19	24	25
	准确率(%)	81.6	82.2	84.8	85.7
	训练时间(秒)	0.078	0.0938	0.0781	0.0513
Diabetis	#SV	7	15	24	30
	准确率(%)	67.2	69.4	70.8	70.9
	训练时间(秒)	0.0156	0.0312	0.0625	0.0703
Breast_Cancer	#SV	9	18	19	28
	准确率(%)	69.6	73.4	72.3	73.3
	训练时间(秒)	0.0469	0.0781	0.0625	0.1094
Image	#SV	9	19	19	39
	准确率(%)	84.5	84.6	84.5	84.5
	训练时间(秒)	0.0313	0.0469	0.0562	0.0625
German	#SV	5	12	20	37
	准确率(%)	68.1	70.3	70.6	70.6
	训练时间(秒)	0.0312	0.0469	0.0781	0.1094

从表 3.4 可以看出，Dix_SVMactive 算法在 Diabetis 集与 German 集上得到的分类精度在 70%左右，在其它数据集上分类精度都能达到 80%以上。这与数据集本身的分布特点、核函数参数的选择以及相关参数的设置有关，如果结合一定的核函数及相关参数选择方法，则会取得更好的泛化效果。

图 3.3 分别给出三种算法在实验数据集上的测试结果比较。



(a) Banana (b) Thyroid

(c) Diabetis (d) Breast_Cancer

(e) Image (f) German

图 3.3 三种方法在实验数据集上的测试结果比较

从图 3.3 中可以看出，Dix_SVMactive 的准确率曲线总体看来均在 Random Sample SVMactive 和 Tong SVMactive 之上，说明本章算法的泛化性能较好。

3.2.2 Dix-SVMactive 训练集平衡度调整实验分析

当 Need_label 为非平衡时，本章算法通过 Step2.4 对训练集平衡度进行了调整。以 Thyroid 为例，当$\varepsilon = 0.5$时，图 3.4 给出了 Dix_SVMactive 对训练集平衡度调整前与调整后的测试结果比较。从图中可以看出，调整了训练集平衡度的算法具有更好的性能。

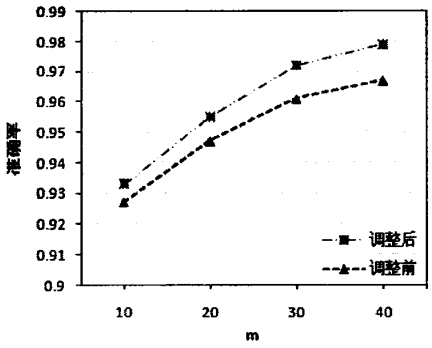


图 3.4 训练集平衡度调整前后泛化结果比较（Thyroid 集）

3.3 本章小结

作为一种日趋成熟的分类器，SVM 越来越广泛地应用于各个领域，与主动学习相结合可获得更好的泛化性能，尤其是当标记大量样本所耗费的代价很大的情况下，只利用未标记样本训练分类器，也能得出很好的分类结果。本章提出的 Dix_SVMactive 算法针对样本价值衡量标准进行了改进，一定程度上增强了算法的泛化能力。而且，当迭代中产生非平衡的训练集时，本算法也能保证较好的分类精度。

由于本算法所采用的样本价值置信度度量方法只考虑了距离因素，因此其性能难免受到数据集分布和样本特征的影响，另外在调整训练集平衡度时，仅对多数类数据进行聚类可能会丢掉一些有价值的样本。如何在距离度量基础上定义更合适的数据置信度度量，并在调整训练集平衡度时将这一度量作为参考，还需要继续深入研究。

此页不缺内容

第四章 基于向量余弦的SVM主动学习策略

本章提出了基于向量余弦的SVM主动学习方法Cos_SVMactive，此方法主要是针对维度较高的未标记样例提出的，其选样策略的核心在于通过未标记样本与当前已标记样本的余弦值来度量未标记样例的价值，而且同样对训练过程中可能出现的训练集偏斜问题提出了解决方法。最后在UCI标准数据集上对本章算法进行了有效性验证，并且与Random Sample SVMactive和Tong SVMactive做了比较。

4.1 基于向量余弦的SVM主动学习算法

本节将详细介绍Cos_SVMactive算法的核心思想，包括置信度定义、训练集偏斜度的调整和停止准则等等。

4.1.1 基于向量余弦的置信度度量方法

由上一章的实验结果看出，Dix_SVMactive在处理低维数据时有较好的实验效果，这充分说明离超平面最近的样本不一定就是最有价值的样本，必须考虑信息冗余的情况。但是当处理维度较高的数据时，传统的欧式距离已经不能正确反应样本间的相关程度。所以本章将样本看作向量，通过向量间夹角的余弦值来判断样本间的相关性，进而度量样本的价值。

同样假设L和U分别表示已标记样本集与未标记样本集，本算法对每个未标记样本x_i的置信度重新定义如下：

$$c = (x_i \cdot \bar{x_j})/(\|x_i\| * \|\bar{x_j}\|) \qquad x_i \in U, x_j \in L \tag{4.1}$$

其中 $\bar{x_j}$表示当前已有的已标记样本x_j各维相加后的平均值，即

$$\bar{x_j} = (\sum_{j=1}^{m} x_j)/m \qquad m = |L| \tag{4.2}$$

$\|x_i\|$和$\|\bar{x_j}\|$分别表示第未标记样本x_i与$\bar{x_j}$的模，（$x_i \cdot \bar{x_j}$）表示x_i和$\bar{x_j}$的内积。

在此置信度中，$\bar{x_j}$度量了当前迭代中已有的已标记样本的平均水平，即将$\bar{x_j}$作为当前已标记样本的代表，然后通过未标记样例x_i与$\bar{x_j}$的余弦值来度量x_i与当前的已标记样例集的相关程度。在$0\sim\pi/2$范围内，余弦值越小，两向量夹角越大，二者的相关度就越小，反之亦然。若x_i与$\bar{x_j}$夹角的余弦值越小，说明x_i与当前已标记样本集包含的信息差异就越大，就越有价值。

本算法中仅对每个与当前超平面的距离小于当前最大分类间隔的未标记样本x_i计算对应的$c(x_i)$。这样既降低了计算复杂度，又避免选到孤立点，可保证算法快速收敛。

综上所述，$c(x_i)$ 的大小直接可反应样本价值的大小。每次迭代中，对当前分类

间隔中的每个未标记样本x_i都计算对应的置信度$c(x_i)$，对其求绝对值后按升序排列，取前 m 个样本进行人工标记，然后加入训练集。

4.1.2 基于向量余弦的训练集平衡度调整策略

同 Dix_SVMactive 一样，Cos_SVMactive 在每步迭代后得到的超平面可能与一类样本的中心距离较远，而与另一类样本中心距离较近，这样，在选取需要计算置信度的未标记样本时可能会选到较多的一类未标记样本和较少的另一类未标记样本，若按置信度选择并交由专家标注过后，得到的已标注样本集有可能会成为非平衡集合，即样本中心离超平面近的类中样本个数可能会大于另一类样本，如果不对数据集中样本的偏斜情况做出处理，将会影响算法的泛化能力。

为避免出现上述的数据不平衡现象，Cos_SVMactive 在每次迭代后同样会检测样本集的平衡度 b，其定义与第三章中式(3.2)相同。

针对高维数据的特性，本算法采用了基于向量余弦的平衡度调整策略。当检测到目标样本集出现偏斜现象时，对多数类样本集中的每个样本x_i，计算其与集合中其它样本夹角的余弦值，取绝对值后计算其平均值记作cos_i，用于度量样本x_i与集合中其它样本的相关性。然后将x_i按对应的cos_i值升序排列如下：

$$x_1,\ x_2,\ \cdots,\ x_t,\ \cdots,\ x_{n-1},\ x_n \tag{4.3}$$

若少数类样本数为 t，则取序列(4.3)中前 t 个样本加入训练集。

4.1.3 Cos-SVMactive 算法

已有研究表明，对改善超平面位置贡献最大的未标记样本常常集中在这样的区域，即此区域中的未标记样本与当前超平面的距离均小于当前最大分类间隔。每次选择这些样本中的某几个交由专家标记并加入训练集，能更快的达到更好的分类精度，从而获得更优的泛化性能，否则将降低算法的收敛速度，甚至使算法的精度趋于发散。所以本算法采用了双停止条件，即①若迭代次数达到预设值，则停止迭代；②若此时与当前超平面的距离小于当前最大分类间隔的未标记样本都被专家标记完毕，则停止迭代。这样既在一定程度上保证了算法的收敛性，又增强了算法与使用者的交互程度，让使用者在获得满足自己需要的分类精度的同时，将算法的运行时间保持在可控的范围内。

假设已标记样本集仍用L表示，未标记样本集用U表示，初始化时令$L = \emptyset$，将所有的未标记样本$\{x_1 \ldots x_n\}$加入集合U，即令$U = \{x_1 \ldots x_n\}$；集合 Need_label 中的元素为每次迭代结束后选出的最有价值样本，其初始值为$\emptyset$，且在每次迭代前都要清空。

Train 中的元素为所有人工标记过的样本，用作 SVM 训练集，其初始值同样为∅。Wrong_label 中的元素为每次迭代中被分错的样本，其初始值也为∅，同样在每次迭代前都要清空。ULx_near_svm 中的元素为每次迭代中与当前超平面的距离小于当前最大分类间隔的未标记样本，其初始值为∅，在每次迭代前同样都要清空。max_w 表示当前超平面的最大分类间隔。

基于以上假设，本文算法的主要步骤总结如下。

Step1 初始化。

将 U 中所有样本聚为 k 类，对应的类中心为$c_1,\dots,c_k$；然后将$c_1,\dots,c_k$交出专家标记，若$c_1,\dots,c_k$中分别含正负类，则令$\text{Train} = \{c_1,\dots,c_k\} \cup \text{Train}$，否则对原始数据聚 k+1 类，重复这一过程，直到类中心分别包含正负类样本为止。令$U = U - \text{Train}$。

Step2 循环 t 步，执行以下步骤。

Step2.1 用 Train 训练 SVM，并对 U 中样本的类别进行预测。

Step2.2 对每个$x_i(x_i \in U)$，计算其与当前超平面的距离$d(x_i)$，若$d(x_i) \leq \text{max_w}$，则将x_i放入 ULx_near_svm 中。

Step2.3 如果没有找到与当前超平面的距离小于当前最大分类间隔的样本，即$\forall x_i(x_i \in U)$，都有$d(x_i) > \text{max_w}$，即$\text{ULx_near_svm} = \emptyset$，那么转 Step3。

Step2.4 对 ULx_near_svm 中的样本按公式(4.1)计算$c(x_i)$。然后按$c(x_i)$的值对 ULx_near_svm 中样本进行升序排列，取前 m 个样本加入 Need_label，并将 Need_label 中的样本交由专家进行标记。

Step2.5 将 Need_label 中各样本的标记结果与 Step2.1 中对应样本的标签进行对比，若二者不同，则将其放入 Wrong_label。

Step2.6 按公式(3.2)计算当前 Wrong_label 集对应的平衡度 b，若$b \leqslant \varepsilon$，则按 4.1.2 节中的方法对 Wrong_label 的平衡度进行调整。

Step2.7 令$\text{Train} = \text{Wrong_label} \cup \text{Train}$，$U = U - \text{Wrong\_label}$。

Step2.8 如果循环次数达到预设值，则转 Step3，否则，继续循环。

Step3 算法结束。

4.2 实验结果与分析

为了验证算法的有效性，在 4 个 UCI 标准数据集（见表 4.1）上进行了实验。实

验中采用多项式核函数，C 取值为 1000，σ 取值为 1.0，迭代次数设为 10。

表 4.1 实验采用的数据集

数据集	训练集个数	测试集个数	数据维数
splice	5000	10875	60
sonar	133	90	60
Hill_Valley_with_noise	606	606	100
Hill_Valley_without_noise	606	606	100

4.2.1 Cos_SVMactive 的有效性验证及实验结果分析

在同等规模的训练集下，以 Hill_Valley_without_noise 为例，本章分别对 Random Sample SVMactive、Tong SVMactive 以及 Cos_SVMactive 进行了比较。由于 Random Sample SVMactive 算法运行结果很不稳定，在训练集规模相等的情况下，连续运行 10 次，将与平均值最接近的那次作为最终结果。三种算法的可调参数 m 都取 30。三种方法的测试结果比较见表 4.2。

表 4.2 三种方法在 Hill_Valley_without_noise 数据集上实验结果比较

方法	准确率(%)	运行时间(秒)
Random Sample SVMactive	72.61	≈ 0
Tong SVMactive	69.97	15.687
Cos_SVMactive	96.7	33.078

对实验结果进行分析，可得到如下结论。

（1）Cos_SVMactive 与 Random Sample SVMactive、Tong SVMactive 相比，在分类精度上有明显的优势。

（2）Tong SVMactive 的运行时间虽然比较短，但与本章算法相比，其分类精度很不理想。本章算法运行时间稍长是由于样本集聚类造成的。

（3）Random Sample SVMactive 的运行时间虽然比 Cos_SVMactive 短，但是其性能很不稳定。10 次运行中，准确率在 59.9%-79.21%之间浮动，而 Cos_SVMactive 在 10 次迭代过程中，准确率基本稳定且收敛。二者的准确率波动曲线对比见图 4.1。

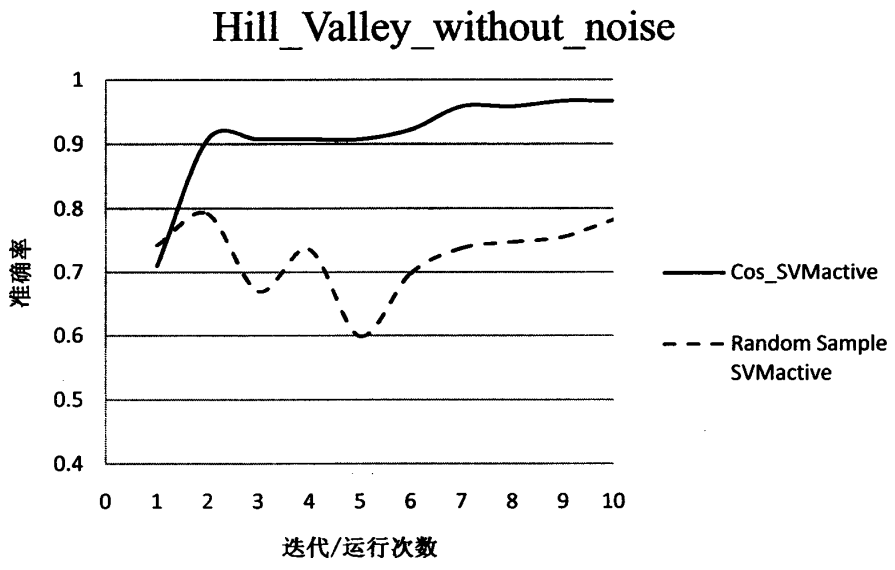


图 4.1 Random Sample SVMactive 和 Cos_SVMactive 的准确率波动曲线

本章算法中还设置了一个可调参数 m，理论上来说，它的值越大就能选到越多有价值的样本，算法的分类精度也越大。下表是 Cos_SVMactive 在 Hill_Valley_without_noise 集上当 m 取不同值时对泛化能力的影响结果，其中#SV 表示最后一次迭代获得的支持向量个数。

表 4.3 m 值对算法结果的影响（Hill_Valley_without_noise 集）

m	10	20	30	40
#SV	16	14	14	16
准确率(%)	94.22	95.54	96.7	92.57

从表中看出，当 m 从 10 增大到 30 时，算法的分类精度有显著提高，但当 m 从 30 增加到 40 时，分类精度反而有所下降。分析后认为造成这种情况的原因可能有以下几个。

（1）数据集本身分布特性所致。图 4.2 显示了当 m 取不同值时另外两种算法在此数据集上的分类精度变化曲线。从图中可以看出，Tong SVMactive 在 m 的值从 20 增加到 40 时也有大幅下降；Random Sample SVMactive 的精确率是取其运行 10 次的平均值，虽然其平均值在 m 值增大的过程中没有下降，但实际运行的结果并不理想。

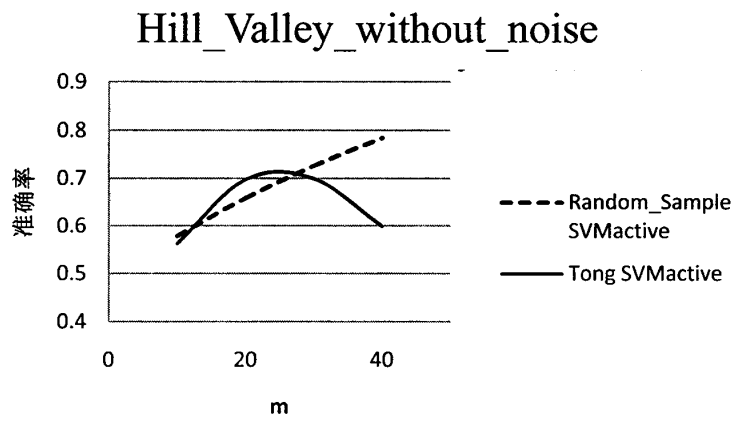


图 4.2 Tong SVMactive 和 Random Sample SVMactive 的准确率变化曲线

（2）当 m 增大到一定值时可能出现了过学习问题，导致分类精度下降，泛化性能减弱。为避免出现这种情况，可以增加迭代次数，算法经过一定的迭代步骤后可收敛于一个稳定值（见实验 4.2.3 部分）。

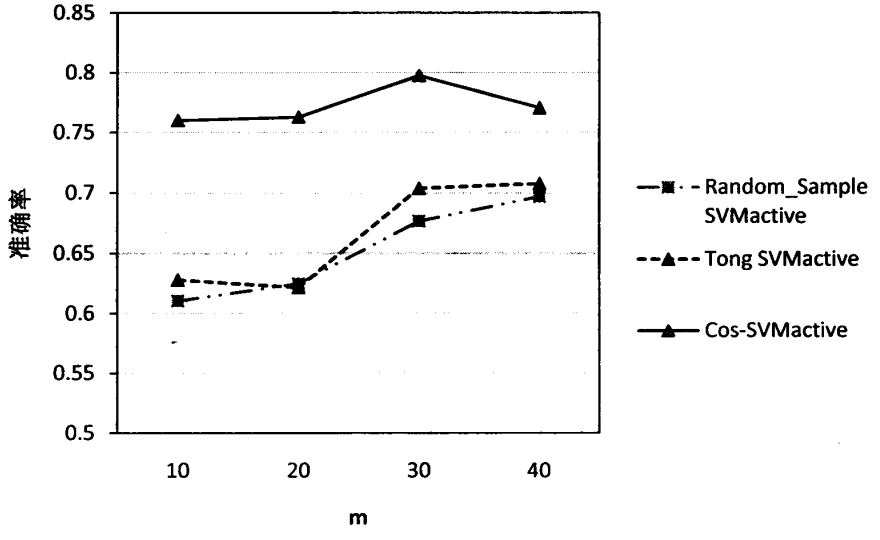
表 4.4 是本文算法在其它数据集上进行测试的结果。其中#SV 表示最后一次迭代获得的支持向量个数，训练时间取 10 次迭代中训练时间的平均值

表 4.4 Cos_SVMactive 在其他数据集上的测试结果（ε=0.5）

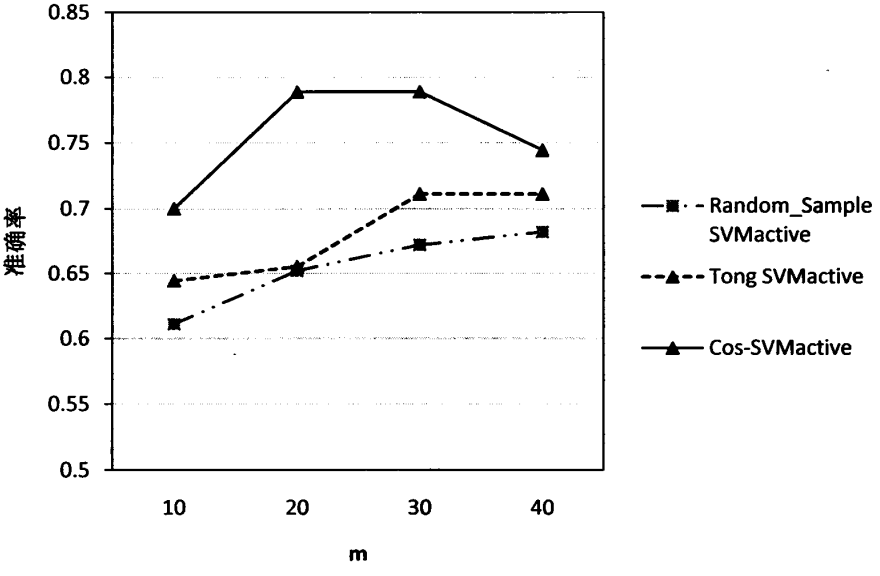
数据集	测试结果	m			
		10	20	30	40
Splice	#SV	19	19	24	25
	准确率(%)	76	76.29	79.77	77.06
	训练时间(秒)	0.078	0.0938	0.0781	0.0513
Sonar	#SV	17	8	8	11
	准确率(%)	70	78.89	78.89	74.44
	训练时间(秒)	0.0156	0.0312	0.0625	0.0703
Hill_Valley_with_noise	#SV	45	45	39	41
	准确率(%)	90.76	90.1	87.95	89.11
	训练时间(秒)	0.7344	0.5547	0.4906	0.9088
Hill_Valley_without_noise	#SV	16	14	14	16
	准确率(%)	94.22	95.54	96.7	92.57
	训练时间(秒)	0.375	0.3016	0.5234	0.9406

从表4.4可以看出，Cos_SVMactive算法在sonar集上得到的分类精度在70%-80%之间，在其它数据集上分类精度都能达到 80%或 90%以上。这与数据集本身的分布特点、核函数参数的选择以及相关参数的设置有关。

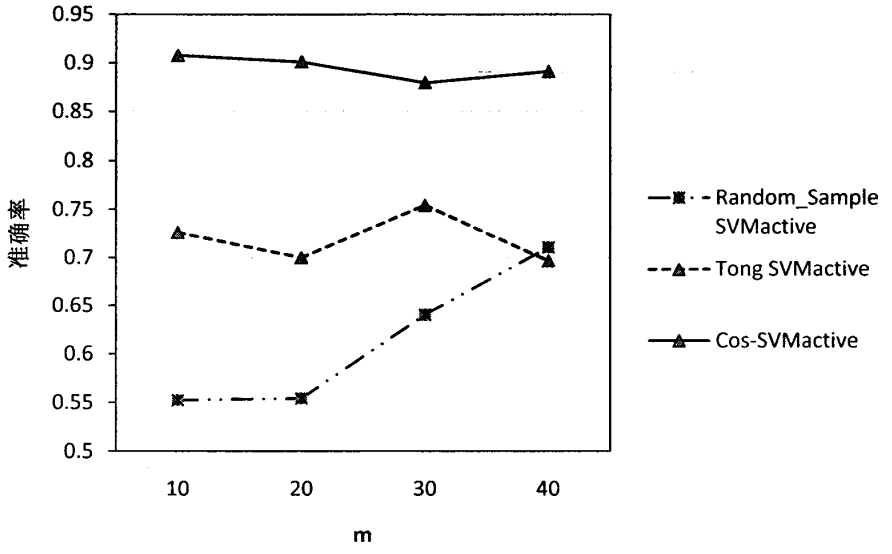
图 4.3 分别给出三种算法在实验数据集上的测试结果比较。



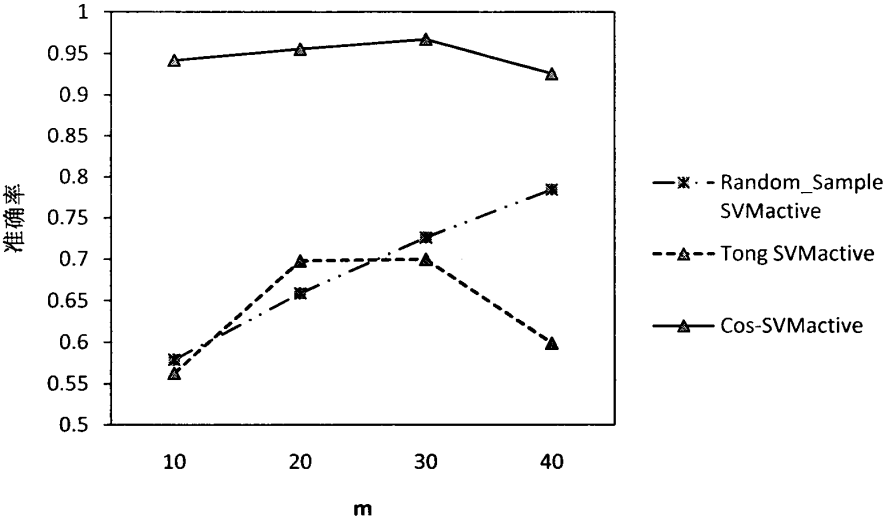
(a) Splice 集



(b) Sonar 集



(c) Hill_Valley_with_noise 集



(d) Hill_Valley_without_noise 集

图 4.3 三种方法在实验数据集上的测试结果比较

从图 4.3 中可以看出，本章算法的准确率曲线总体看来均在 Random Sample SVMactive 和 Tong SVMactive 之上，说明本文算法的学习性能较好。另外本章算法虽然准确率的整体水平较高，但是在部分数据集上，算法的精度会在 m 值从 30 增长到 40 的时候有所下降。

同样，当 Need_label 为非平衡时，本文算法通过 Step2.6 对训练集平衡度进行了

调整。以 Hill_Valley_without_noise 为例，当$\varepsilon = 0.5$时，图 4.4 给出了本文算法对训练集平衡度调整前与调整后的测试结果比较。从图中可以看出，调整了训练集平衡度的算法具有更好的性能。

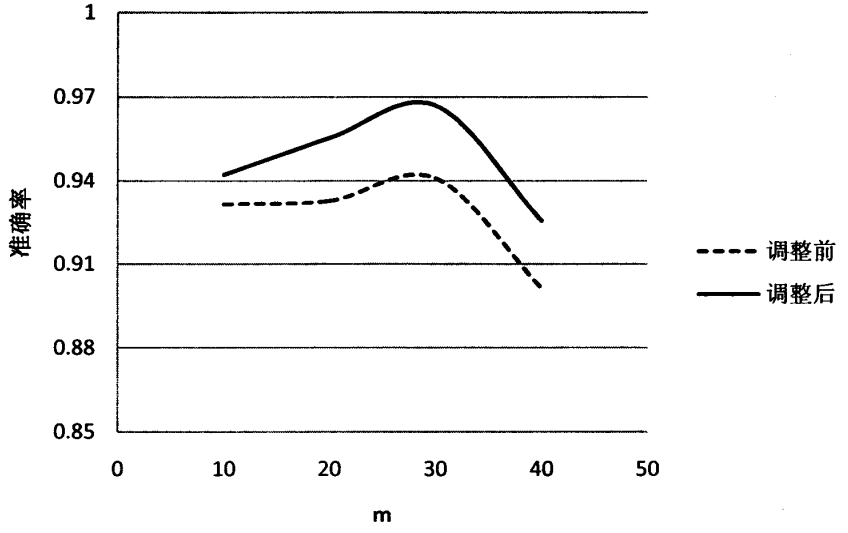


图 4.4 训练集平衡度调整前后泛化结果比较（Hill_Valley_without_noise 集）

4.2.2 Cos_SVMactive 在不同迭代停止条件下的实验分析

在 4.1.3 中，详细阐述了本章算法的具体步骤中 Step2.2 和 Step2.3 的必要性和重要性，以下称去掉 Step2.2 和 Step2.3 的 Cos_SVMactive 算法为 CosTest_SVMactive。本小节中，以数据集 Sonar 为例，比较 Cos_SVMactive 算法和 CosTest_SVMactive 算法的分类精度和泛化性能。实验中采用高斯核函数，C 取值为 1000，σ 取值为 1.0，$\varepsilon = 0.5$。

下表为以上两种算法在 m 取不同值时的分类精度对比。迭代次数为 10。

表 4.5 Cos_SVMactive 和 CosTest_SVMactive 在 m 取不同值时的分类精度对比

m	10	20	30	40
Cos_SVMactive	70%	78.89%	78.89%	74.44%
CosTest_SVMactive	73.33%	72.22%	72.22%	74.44%

图 4.5 为两种算法的分类精度曲线对比图。

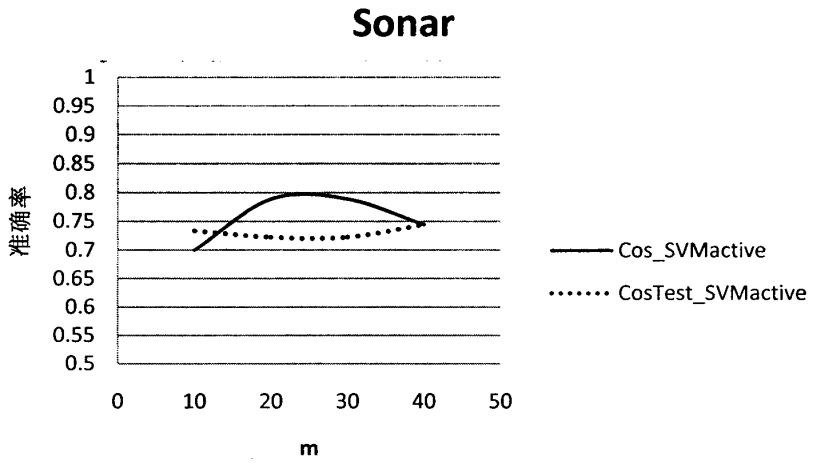


图 4.5 Cos_SVMactive 和 CosTest_SVMactive 在 m 取不同值时的分类精度曲线对比

从表 4.5 和图 4.5 可以看出，Cos_SVMactive 的分类精度在整体水平上要明显高于 CosTest_SVMactive，即经过平衡度调整的算法要好于未经过调整的算法。

Cos_SVMactive 在数据集 Sonar 上只经过了 1 次迭代就达到了表 4.5 中的精度，因此表 4.6 只列出在 Sonar 集上 CosTest_SVMactive 当 m 取不同值时 10 次迭代中的分类精度。

表 4.6 m 取不同值时 CosTest_SVMactive 10 次迭代中的分类精度

m \ 迭代次数 / 准确率/%	1	2	3	4	5	6	7	8	9	10
10	72.22	72.22	68.89	68.89	70	72.22	73.33	73.33	74.44	73.33
20	71.11	72.22	72.22	67.78	67.78	67.78	70	68.89	72.22	72.22
30	68.89	74.44	74.44	76.67	76.67	76.67	74.44	75.56	72.22	72.22
40	73.33	64.44	66.67	67.78	67.78	67.78	67.78	71.11	73.33	74.44

图 4.6 给出了当 m 取不同值时 CosTest_SVMactive 在 10 次迭代中的分类精度的变化曲线。从图 4.6 中可以看到，CosTest_SVMactive 的分类精度在 10 次迭代中变化很混乱，没有收敛的趋势，性能很不稳定。尽管它的迭代次数远大于 Cos_SVMactive 的迭代次数，但最后的分类精度仍低于 Cos_SVMactive。这充分说明 Cos_SVMactive 方法中的 Step2.2 和 Step2.3 是很必要的，即与当前超平面的距离小于当前最大分类间隔的未标记样本在改善超平面位置的过程中起着很重要的作用。

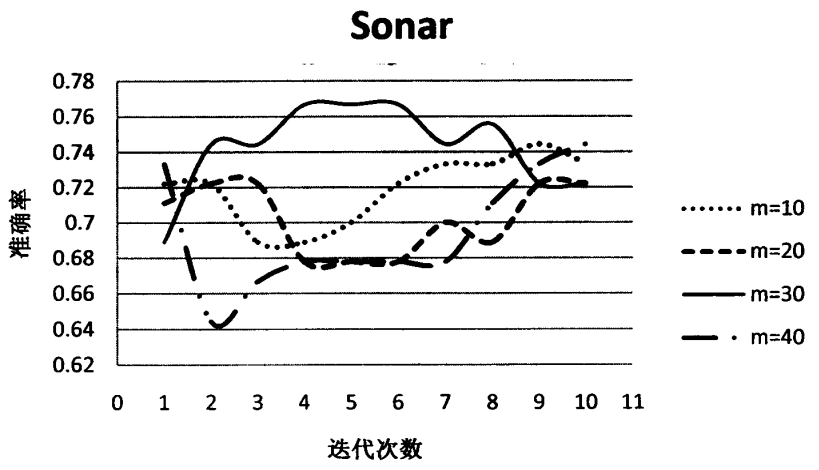


图 4.6 m 取不同值时 CosTest_SVMactive 在 10 次迭代中的分类精度变化曲线

4.2.3 Cos_SVMactive 的收敛性实验分析

本小节将在各个数据集上测试当 m 取不同值时 Cos_SVMactive 在多次迭代后的分类精度。

表 4.7-4.9 中列出了 m 为 10、20、30 和 40 时 Cos_SVMactive 在 Splice、Hill_Valley_with_noise 及 Hill_Valley_without_noise 三个数据集上 10 次迭代中的分类精度。

表 4.7 m 取不同值时 Cos_SVMactive 在 Splice 集上的分类精度

m \ 迭代次数 / 准确率/%	1	2	3	4	5	6	7	8	9	10
10	67.96	68.03	71.82	70.45	73.49	75.69	75.69	75.69	76	76
20	66.7	70.23	73.37	73.73	76.29	76.29	76.29	76.29	76.29	76.29
30	66.64	70.31	72.58	74.42	79.77	79.77	79.77	79.77	79.77	79.77
40	67.45	72.05	73.56	77.06	77.06	77.06	77.06	77.06	77.06	77.06

表 4.8 m 取不同值时 Cos_SVMactive 在 Hill_Valley_with_noise 集上的分类精度

m \ 迭代次数 / 准确率/%	1	2	3	4	5	6	7	8	9	10
10	50.17	59.41	68.65	88.94	88.78	89.93	90.43	90.59	90.76	90.76
20	53.8	78.05	91.09	91.42	91.42	91.42	91.09	91.25	90.1	90.1
30	58.09	85.31	85.97	86.14	87.29	87.95	88.28	88.61	88.78	87.95
40	58.75	85.97	87.95	88.28	88.12	89.77	88.61	88.61	89.27	89.11

表 4.9 m取不同值时 Cos_SVMactive 在 Hill_Valley_without_noise 集上的分类精度

m \ 迭代次数 / 准确率/%	1	2	3	4	5	6	7	8	9	10
10	60.01	81.52	90.92	94.06	94.22	94.22	94.22	94.22	94.22	94.22
20	67.16	89.60	91.09	95.87	95.87	95.87	95.21	95.21	95.54	95.54
30	70.96	90.76	90.76	90.76	90.76	92.24	95.87	95.87	96.7	96.7
40	73.93	90.43	90.43	90.26	91.42	91.75	91.75	91.75	93.07	92.57

图 4.7-4.9 中给出了 m 为 10、20、30 和 40 时 Cos_SVMactive 在 Splice、Hill_Valley_with_noise 及 Hill_Valley_without_noise 三个数据集上 10 次迭代中的分类精度变化曲线。

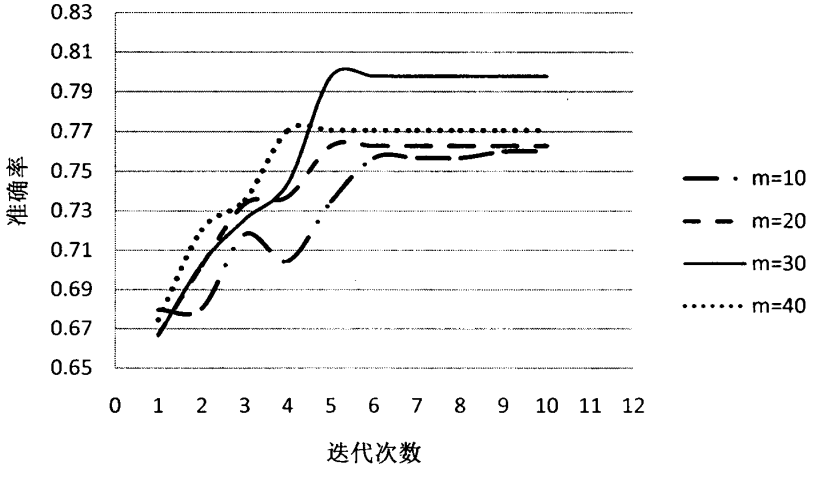


图 4.7 m取不同值时 Cos_SVMactive 在 Splice 集上的分类精度变化曲线

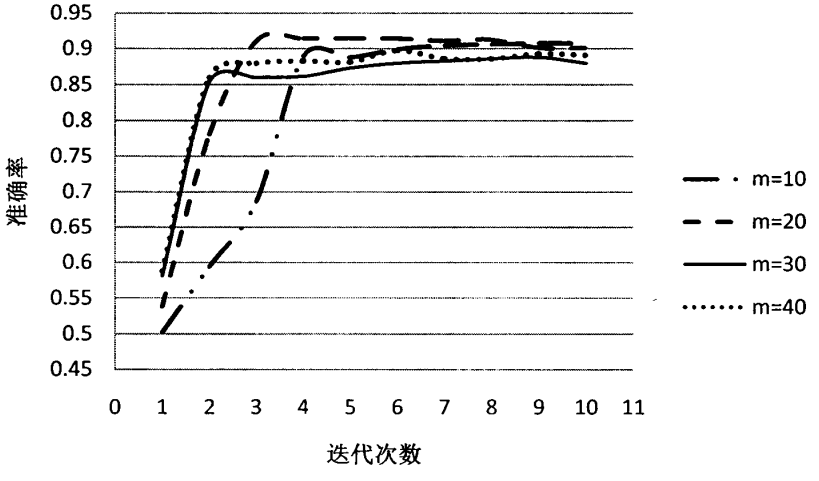


图 4.8 m取不同值时 Cos_SVMactive 在 Hill_Valley_with_noise 集上的分类精度变化曲线

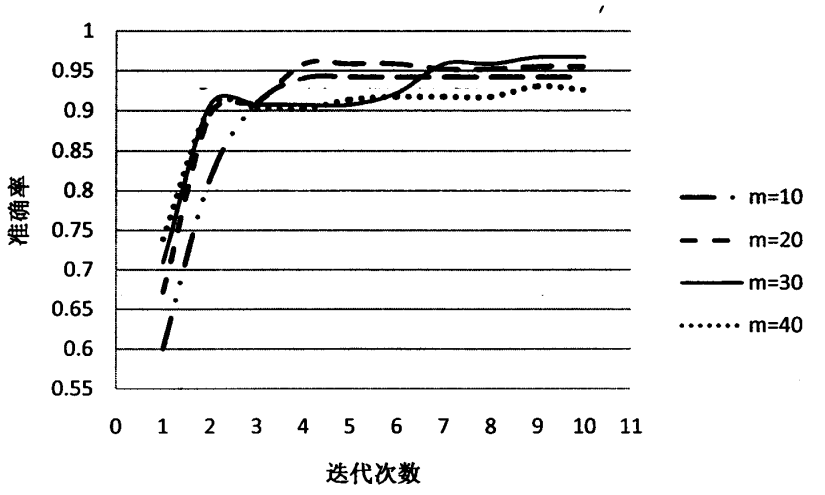


图 4.9 m 不同值时 Cos_SVMactive 在 Hill_Valley_without_noise 集上的分类精度变化曲线

Cos_SVMactive 在 Sonar 集上只经过 1 次迭代就达到了表 4.4 中所示的精度，本身就可以看出 Cos_SVMactive 的收敛性是很好的。另外，我们从以上图表中的数据和曲线可以看出，本章算法在其它三个数据集上的收敛性也很不错。虽然限于篇幅，以上图表只截取了 Cos_SVMactive 迭代 10 次、m 的值取 10、20、30 和 40 的实验结果，图表中所示的分类精度有可能还不是 Cos_SVMactive 能达到的最高精度，但从分类精度变化曲线的走势来看，本章算法的分类精度最后能够收敛到一个稳定的值。

4.3 本章小结

本章对新提出的基于向量余弦的SVM主动学习方法Cos_SVMactive进行了详尽的介绍，包括算法的核心思想、置信度的定义和具体步骤等等。从算法的原理来看 Cos_SVMactive 的处理对象主要是维度较高的数据，从 UCI 数据集上的实验结果上可以很清楚的看到，在分类精度、收敛性、运行时间和训练时间等方面，本章算法都取得了较好的效果。尤其是与基于随机选样（Random Sample）的 SVMactive 和传统 SVMactive（Tong SVMactive）方法相比，Cos_SVMactive 具有更优的泛化性能。

当然，在实验数据中我们也看到了一些不太理想的结果，例如在有的数据集上 Cos_SVMactive 的分类精度会随着 m 值的增加而有小幅下降，在有的数据集上的分类精度不太理想等等。分析认为，造成这些结果的原因可能包括数据集本身分布的特点、算法本身存在某些细节上的问题等等。这需要在以后的工作中进行改进。

此页不缺内容

第五章 结论与展望

SVM 本身具有强大的分类能力和优良的泛化性能，被广泛的应用于各个领域。但随着实际问题的多样化，SVM 将面对更多的无标记样本，而人工对样本进行标注所花费的代价往往又是巨大的。所以，本文从众多的利用未标记样本学习的方法中，选择了主动学习，并将其与 SVM 结合，以解决 SVM 遇到的上述问题。经过深入研究，本文取得了一定的成果，现总结如下。

（1）对现有的 SVM 主动学习策略进行了分析总结，提出现有的 SVM 主动学习算法中存在的主要问题，并针对这些问题开展研究。

（2）提出了基于距离的 Dix_SVMactive 方法。新方法采用了新的距离度量来衡量未标记样本的价值，既考虑了当前未标记样本与当前超平面间的距离，又考虑了当前未标记样本与当前训练集中已标记样本的距离。而且在训练过程中实时监测当前迭代中新生成的训练集的平衡度。从而解决了如下问题：避免了未来输入的未标记样本与当前的已标记样本产生信息冗余，影响收敛速度；避免了迭代过程中新生成的训练集出现偏斜现象，影响泛化性能。本算法从单个样本和集合整体两个角度保证了分类精度和收敛速度。

（3）提出了基于向量余弦的 Cos_SVMactive 方法。新方法针对维度较高的数据提出了新的未标记样本价值度量方法，先求出当前训练集中已标记样本各维相加的平均值，即“已标记样本均值”，然后通过当前未标记样本与当前已标记样本均值的夹角余弦值来衡量当前未标记样本的价值。另外，算法仅对当前分类间隔内的未标记样本求上述置信度，而不是对所有给定的未标记样本进行计算。算法还采用了新的双条件迭代停止准则，即若迭代次数达到预设值，则停止迭代；若与当前分类间隔内的未标记样本已全部被标记完毕，则停止迭代。通过上述方法，解决了以下问题：当样本维度较高时，传统的欧式距离或许不能准确的反映出样本间的相关度，仅凭距离来衡量高维样本的价值并不能选到真正有价值的未标记样本；如果对所有给定样本集计算置信度，势必会增加算法的计算复杂度和时间复杂度，影响算法的性能；只通过迭代次数判断是否停止迭代，既可能错过最佳的分类精度，又可能增加不必要的计算代价和运行时间。

（4）对提出的两种新的 SVM 主动学习算法在 UCI 标准数据集上进行了实验验证，取得了很好的效果。

本文仅对基于 SVM 的主动学习方法做了初步的探索和研究，以后将在以下几个

方面继续深入。

（1）针对不同特征的未标记样本集，研究一种自适应的数据置信度度量，从数据本身的特点出发，更准确的衡量样本的价值。

（2）改善算法的停止准则，替代人工设置参数，使其能自行在达到最优分类精度时停止迭代。

（3）从信息学角度度量未标记样本的价值仍是下一步工作的重点之一。

本文所进行的基于 SVM 的主动学习方法研究，既有成熟的理论基础，又有广阔的应用前景。文中所提出的新方法为这方面的工作提供了新的思路，也为 SVM 在处理大量未标记样本方面提供了方法层面的补充。

参考文献

[1]Vapnik V. Statistical Learning Theory[M]. New York, Wiley, 1998, 11-23.

[2]Vladimir N, Vapnik V. The Nature of Statistical Learning Theory[M]. New York, Springer, 2010, 123-179.

[3]Suykens J, Vandewa Lle . Least squares support vector machine classifiers[J]. Neural Processing Letters, 1999, 9(3), 293-300.

[4]Mukerjee S, Osuna E, Girosi F. Nonlinear prediction of chaotic time series using a support vector machine, Principle[J]. Proceedings of the 1997 IEEE Workshop on Neural Networks for Signal Processing, [S.I.]: IEEE Press, 1997, 1125-1132.

[5]Osuna E, Freund R, Girosi F. Training support vector machines: an application to face detection[C]. IEEE Computer Society Conference on Computer Vision and Patten Recognition, 1997, 130—136.

[6]卢增祥, 李衍达. 交互支持向量机学习算法及其应用[J]. 清华大学学报（自然科学版）, 1999, 39(7), 93-97.

[7]Tong S. Active learning: theory and application[R]. Stanford University, 2001, 1-168.

[8]Seung H S, Opper M, Sompolinsky H. Query by committee[C]. Proceedings of the 15th Annual ACM Workshop on Computational Learning Theory, California, 1992, 287-294.

[9]Dagan I, Engelson S. Committee-based sampling for training probabilistic classifiers[C]. Proceedings of the 12th Int. Conf. on Machine learning, 1995, 150-157.

[10]Nguyen H T. Active learning using pre-clustering[C]. The 21st Intl Conf on Machine Learning, Banff, Alberta, Canada, 2004, July 04-08, 79.

[11]Lughofer E. Hybrid active learning for reducing the annotation effort of operators in classification systems[J]. Pattern Recognition, 2012, 45, 884 - 896.

[12]孙功星, 戴贵亮. 神经网络主动学习的进化算法[J]. 计算机科学, 2002, 29(10), 61-63.

[13]赵悦，穆志纯．基于 QBC 的主动学习研究及其应用[J]．计算机工程，2006，32(24)，23-25.

[14]Tong S, Koller D. Support vector machine active learning with applications to text classification[J]. Journal of Machine Learning Research, 2002, 2, 45-66.

[15]段丹青,陈松乔,杨卫平．网络入侵检测中的支持向量机主动学习算法[J]．计算机工程与应用，2006，01，117-119.

[16]Mukerjee S, Osuna E, Girosi F. Nonlinear prediction of chaotic time series using a support vector machine[C]. Principle J, Giles L, Morgan N. Proceedings of the 1997 IEEE Workshop on Neural Networks for Signal Processing, [S.I.]: IEEE Press, 1997, 1125-1132.

[17]Schohn G, Cohn D. Less is more: active learning with support vector machines[C]. Proceedings of the 17th Int. Con. on Machine Learning, San Francisco: Morgan Kaufmann, 2000, 45-66.

[18]David D, Lewis William, Gale A. A sequential algorithm for training text classifiers (Uncertainty Sampling)[C]. Proceedings of Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, Springer-Verlag, London, 1994, 3-12.

[19]Vlachos A. Active learning with support vector machines[D]. Master of Science, School of Informatics, University of Edinburgh. 2004.

[20]周艳丽．基于主动学习SVM的智能车辆障碍物检测[D]．南京理工大学，2008，6.

[21]张健沛，徐华．支持向量机(SVM)主动学习方法研究与应用[J]．计算机应用，2004. 24(1)，1-3.

[22]解洪胜，张虹．基于支持向量机的图像检索主动学习方法[J]．山东师范大学学报，2007，22(4)，46-48.

[23]Wu H Y, Huang C H. Multi-path QOS routing in TDMA/CDMA ad hoc wireless networks[J]. Lecture Notes in Computer Science, 2004, 3252, 609-616.

[24]Simon H A, Lea G. Problem solving and rule education: a unified view knowledge and organization[J]. Knowledge and Cognition, 1974, 15(2), 63-73.

[25]龙军，殷建平，祝恩，赵文涛.主动学习研究综述[J].计算机研究与发展，2008，

45, 300-304.

[26]Scheffer T, Wrobel S. Active learning of partially hidden markov models[C]. In:Proc of the ECML/PKDD, 2001, Berlin:Springer, 2001, 1-15.

[27]韩光, 赵春霞, 胡雪蕾. 一种新的 SVM 主动学习算法及其在障碍物检测中的应用[J]. 计算机研究与发展, 2009, 46(11), 1934-1941.

[28]Freund Y, Seung H S, Samir E, et al. Selective sampling using the query by committee algorithm[J]. Machine Learning, 1997, 28(23), 133-168.

[29]龙军, 殷建平, 祝恩, 蔡志平. 选取最大可能预测错误样例的主动学习算法[J]. 计算机研究与发展, 2008, 45, 472-478.

[30]Baram Y, El-Yaniv R, Luz K. Online choice of active learning algorithm[C]. In: Proc of the 20th Intl Conf on Machine Learning San Francisco: Morgan Kaufmann, 2004, 5, 255-291.

[31]Muslea I, Minton S, Craig A, Knoblock. Active learning with multiple views[J]. Journal of Artificial Intelligence Research, 2006, 27, 203-233.

[32]韩光, 赵春霞, 胡雪蕾. 一种新的 SVM 主动学习算法及其在障碍物检测中的应用[J]. 计算机研究与发展, 2009, 46(11) ,15-20.

[33]Ranganathan K., Iamnitchi A., Foster I. Improving data availability through dynamic model-driven replication in large peer-to-peer communities[C]. CCGrid, 2002, 5, 376- 381.

此页不缺内容

攻读学位期间取得的研究成果

[1] 白龙飞, 王文剑, 郭虎升. 一种新的 SVM 主动学习策略[J]. 南京大学学报（自然科学版）.2012. 2, 60-67

此页不缺内容

致 谢

时光荏苒，岁月如梭，三年前毕业的场景还未从脑海中散去，又一个毕业季向我走来。在这个熟悉的城市，在这个熟悉的学校，我又走过了三年，三年前新生报到的时候，我曾和同学戏谑："三年呐，还有三年呐！"可话音还没落，就到了离开的时候。时间虽然走得飞快，但留下的东西却太多，闲暇时看着眼前熟悉的一切，总会想起在这个教室里，曾聆听过老师的教导；在这条小路上，曾和同学朋友打闹；在这条小路上，曾接过远方的家人打来的电话。

在这三年里，我最先也最想要感谢的是我的导师王文剑教授。自从听过王老师讲授的课以后，我就觉得王老师是一位既认真严格又宽容、有责任感的老师，所以在上研选导师的时候，我选择加入王老师的团队。研一时，王老师就根据我的自身情况为我制订了培养计划，研二在撰写小论文前后，王老师不仅悉心帮我解决实验过程中的问题，而且对论文中诸如遣词造句等极小的细节做了细致的修改。在研三这一年里，王老师也时刻关注着我找工作的进展，并且结合我的实际情况针对就业方向为我提出了许多建议。在这三年里，王老师教会了我如何学习，如何解决问题；在这三年里，王老师更教会了我做人要谦虚平和、勤勤恳恳，做事要认真负责，果断谨慎。所有这些，都对我以后的工作、生活有着莫大的意义和作用。

在这三年里，另外一个我想要感谢的老师就是我的班主任阎老师。记得刚入学时，因为我无故缺席集体活动，而且事后没有及时认识到自己的错误，受到了阎老师的批评，当时觉得阎老师很严厉。但在后来的三年里，在生活上我一次次的感受到了阎老师的关怀，尤其在党性教育上，阎老师不止一次地用实际行动告诉我如何做一个合格的共产党员。在最后的这一年里，我更是在阎老师的鼓励和帮助下加入了中国共产党。这莫大的恩情我将没齿难忘。

在这里，我还要感谢团队里的师兄师姐师弟师妹们，尤其是门昌骞老师、郭虎升师兄和王亚贝师姐，在你们的帮助下，我从入学时的茫无头绪开始，一步一步地成长，一直到现在。三年的生活中，我就像生活在一个大家庭里，无时不刻都能感受到你们带给我的温暖。

另外，我还要感谢和我相处了三年的同班同学们。研一刚报到时的茫然，研二发小论文时的紧张，研三四处奔波找工作时的劳累，每一段日子里都有你们的身影，每一次经历都有你们的出现。还有研一与我同宿舍的师兄们、研二与我同宿舍的师弟们，谢谢兄弟朋友和所有人在三年里给予我的一切。

最后，我要感谢我的奶奶、爸爸、妈妈，还有我去世的爷爷，还有我其他所有的家人们，谢谢爷爷在十几年里对我的抚养，谢谢奶奶这些年来为我的付出，谢谢爸爸为我做的所有牺牲，谢谢妈妈在养育我的这二十几年里付出的心血，谢谢你们在我十八年的求学路上为我做的一切，谢谢你们对我的包容，谢谢你们对我的鼓励，谢谢你们对我的关怀！我知道不管我做什么都无法报答你们对我的恩情，但我会努力工作，认真生活，让你们因为有我而骄傲！谢谢，我可爱的亲人们，我永远爱你们！

个人简况及联系方式

个人简况：

姓名：白龙飞

性别：男

籍贯：山西省忻州市偏关县

个人简历：

2009.9- 至今　山西大学计算机与信息技术学院　计算机软件与理论　硕士

2005.9-2009.7　山西大学计算机与信息技术学院　计算机科学与技术　学士

联系方式：

电话：

电子信箱：liangansandiblf@126.com

等：

此页不缺内容

承 诺 书

本人郑重声明：所呈交的学位论文，是在导师指导下独立完成的，学位论文的知识产权属于山西大学。如果今后以其他单位名义发表与在读期间学位论文相关的内容，将承担法律责任。除文中已经注明引用的文献资料外，本学位论文不包括任何其他个人或集体已经发表或撰写过的成果。

作者签名：

2012年5月17日

此页不缺内容

学位论文使用授权声明

本人完全了解山西大学有关保留、使用学位论文的规定，即：学校有权保留并向国家有关机关或机构送交论文的复印件和电子文档，允许论文被查阅和借阅，可以采用影印、缩印或扫描等手段保存、汇编学位论文。同意山西大学可以用不同方式在不同媒体上发表、传播论文的全部或部分内容。

保密的学位论文在解密后遵守此协议。

作者签名：

导师签名：

2012年5月17日