

基于 SVM 增量学习的用户适应性研究^{*}

彭彬彬 金翔宇 徐晓刚 孙正兴

(南京大学软件新技术国家重点实验室 计算机科学与技术系 南京210093)

Study of SVM-Based Incremental Learning for User Adaptation

PENG Bin-Bin JIN Xiang-Yu XU Xiao-Gang SUN Zheng-Xing

(State Key Laboratory for Novell Software Technology, Nanjing 210093)

Abstract User adaptation is a critical and important problem. For users' specialization, such as Handwriting, Voice, Drawing Styles, the system is hard to adapt to all users. SVM-based incremental learning can find the most basic feature of different users and cast away the special user's character, so this method can adapt the different users without over fitting. In this paper, the repetitive learning strategy and other two incremental learning algorithms are presented for comparison. Based on theoretical analysis and experimental results, we draw the conclusion that SVM-based incremental learning can solve the user conflict problem.

Keywords Support Vector Machine (SVM), Incremental learning, Intelligent human-computer interface, User conflict, User adaptation

1. 引言

人机交互技术(Human-Computer Interaction, HCI)是一门新兴的边缘学科,在近十几年内迅速形成并得到发展。人机交互是对于人、计算机技术以及它们相互影响方式的研究,其目的是为了使计算机技术更加适合于人^[1]。用户意图预测(User Intention Prediction)是智能化人机交互的关键所在。对一个多用户系统而言,不同的用户具有不同的特性或习惯,例如笔迹、口音、绘画习惯等。用户之间的意图往往是歧义的,甚至是矛盾的,因此建立一个通用的能够适合所有用户使用的系统从某种程度而言是不可能的。以在线图形识别问题(对用户绘制的基本图形的草图进行识别)为例^[2],不同用户输入的不同图形可能非常相似。如图1所示,用户1的目的是输入四边形(如图1(a)),用户2的目的是输入三角形(如图1(b))。由于用户2总是习惯将拐角画得比较圆,造成两个用户输入的图形非常相似,但是他们的意图却是不同的,这就为正确的意图预测带来了困难。因此,需要建立一种用户适应机制,能够通过用户的反馈,使得系统逐步适应用户特性或习惯,从而准确地预测用户的意图。

用户适应系统(User Adaptive Systems)是指以用户模式的获取和应用为基础适应各个个体用户的习惯的交互式系统,当然其中包含某种形式的学习、推理或者决策等^[3]。用户适应性是智能化人机接口的重要部分,它为人机接口的人性化、智能化和个性化提供支持。一般而言,用户适应性的解决方法有两种:基于规则的相关反馈方法和基于重复统计学习的方法。基于规则的相关反馈方法针对单用户而言比较有效,但当系统对多个用户开放时,就会出现矫枉过正的现象(ad hoc)。即系统对一个用户的适应性增强到一定的时候,对另一用户的适应性就会受到影响。基于统计学习理论的支持向量

机^[4](Support Vector Machine, SVM)是基于小样本的学习方法,它实现了结构风险最小化,避免了在学习过程中存在的过学习现象。然而经典的 SVM 理论不支持增量学习,每次新样本的引入都必须重复学习历史数据。在样本数比较小的时候,训练还能够在这段时间内完成;当样本数目膨胀的时候,学习效率就会急剧下降。



(a) 用户 1 的意图为四边形 (b) 用户 2 的意图为三角形

图1 尽管用户习惯的不同,不同的用户意图却有类似的形式

增量学习能有效地利用历史训练结果,从而在很小的时间空间代价下实现新样本的学习。支持向量机能在很小的样本集获得较好的效果,且不存在过学习的现象。基于 SVM 增量学习的方法,能从用户数据中找到根本的特性,而不会将可能造成用户冲突的、特定用户的习惯也记录下来,因此也就不会产生用户冲突。本文将首先介绍支持向量机理论和两种不同的增量学习方法,然后使用基于 SVM 增量学习的方法来解决用户适应性的问题,并构造了多值分类器。最后我们进行了实验,并对其进行分析比较。

2. 基于支持向量机的增量学习方法

2.1 支持向量机

支持向量机是统计学习理论中最年轻的内容,也是最实用的部分,其核心内容是在1992到1995年间提出的^[4,5]。支持向量机的主要思想就是通过构造一个非线性的核函数来将低

^{*} 本文得到国家自然科学基金项目(编号:69903006)资助。彭彬彬 硕士生,研究方向为机器学习,数据挖掘。金翔宇 硕士生,研究方向为图形识别,智能化人机接口。徐晓刚 硕士生,研究方向为图形识别,人机接口。孙正兴 副教授,博士后,研究方向为多媒体辅助工程、智能信息处理。

维的不可分的输入空间数据映射到高维可分的属性空间,然后再寻找一个满足分类要求的最优分割超平面,并使其在保证分类精度的同时最大化超平面两侧的空白区域。所以,这种方法是全局最优的,不存在过学习问题,而且从根本上来讲,它是用来解决二值分类问题的。

最优分类超平面^[4]可以表示为 $\omega \cdot x + b = 0$, 其中 ω 表示超平面的法线, b 表示超平面的位移。训练样本集可以表示为 $(x_1, y_1), \dots, (x_n, y_n)$, 其中 $x \in R^d, y_i \in \{-1, +1\}, i = 1, \dots, n$ 。根据 Kuhn-Tucker 条件知道, ω 可以用训练样本的线性组合来表示:

$$w = \sum_{i=1}^n a_i y_i x_i, \text{ 其中 } 0 < a_i \leq C \quad (1)$$

分类器可以表示为:

$$f(x) = \text{sgn} \left[\sum_{i=1}^n y_i a_i (x \cdot x_i) + b \right] \quad (2)$$

其中 x 表示测试样本, $x_i \in SV_S$

SV_S 是支持向量集合, 它是不等式 $y_i [x_i \cdot w - b] \geq 1, i = 1, \dots, n$ 中等式成立的点 x_i , 这些点代表了整个样本集的特征。以上都是对于线性可分的样本点适用, 在实际情况下, 绝大部分样本空间都是线性不可分的, 这就需要引进映射函数 $(\Phi(x))$, 来将样本集映射到一个线性可分的高维特征空间上。因此:

$$\begin{aligned} f(x) &= \text{sgn} \left[\sum_{i=1}^n y_i a_i (\Phi(x) \cdot \Phi(x_i)) + b \right] \\ &= \text{sgn} \left[\sum_{i=1}^n y_i a_i K(x, x_i) + b \right] \end{aligned} \quad (3)$$

其中 $K(x, x_i) = \Phi(x) \cdot \Phi(x_i), K(x, x_i)$ 就是核函数。

在进行机器学习的过程中, 虽然我们可以将所有数据都重新训练来进行学习, 但是这种方法的学习效率比较低。克服这个问题的一个方法是增量学习, 增量学习不把每一次训练看作一个完美的过程, 而是认为系统每通过一次学习效率就提高一点, 适应就增强一点, 系统就完善一点, 系统经过几次学习之后, 就可以达到比较稳定的状态。为了达到这个目的, 在每一次训练过程(记为 $Train(\mu)$)中, 都要记录下比较重要的样本点(μ_{sv}), 抛弃大量的不重要的数据点(μ_{non-sv})。那么在每次训练过程中的复杂度^[6]为:

$$O(\|\mu_{sv}\|^2 + \|\mu\| \times \|\mu_{sv}\|^2 + d \|\mu\| \times \|\mu_{sv}\|) \quad (4)$$

一般情况下: $\|\mu_{sv}\| \ll \|\mu\|$, 从而训练的复杂度可以表示为:

$$O(\|\mu\| \times \|\mu_{sv}\|^2) \quad (5)$$

然后用训练得到的分类器对测试数据集进行分类(记为 $Classify(V)$), 将得到为分类正确和分类错误的两个子集:

$$v_{ok} = \{(x, y) | (x, y) \in v, y, f(x) \geq 0\}$$

$$v_{error} = \{(x, y) | (x, y) \in v, y, f(x) < 0\}$$

那么分类的精度可以定义为:

$$p = \|v_{ok}\| / \|v\| \quad (6)$$

分类算法的复杂度则为:

$$O(\|v\| \times \|\mu_{sv}\|) \quad (7)$$

对一个 SVM 分类器 Γ , 设初始训练集为 IS , 增量训练集为 INS , 那么重复学习可以表示为 $Train(IS \cup INS)$, 增量学习可以表示为 $Train(Sub(IS) \cup INS)$, 其中 $Sub(IS) \subset IS$ 。进行增量学习一般要求分类边界没有重大变化, 由这个条件可知一般 $\|INS\| \ll \|IS\|$, 则

$$\|(IS \cup INS)_n\| \approx \|(IS_n \cup INS_n)\| \approx \|IS_n\| \quad (8)$$

重复学习的复杂度为:

$$O(\|IS \cup INS\| \times \|IS_n\|^2) = O(\|IS\| \|IS_n\|^2) \quad (9)$$

为了简化下面的讨论, 我们在下一节中的讨论是针对二值分类的, 而且这也很容易扩展到多值分类问题。为了下面形式化的需要, 我们也定义了下面常用的数据集合: WS 为当前工作集, BS 为后备数据集, NS 为一个新数据集, 而 DS 则为抛弃数据集。

2.2 基于 SVM 的增量学习算法

萧嵘等^[7]指出在 SVM 增量学习算法中两个主要关注的问题是: 1) 如何利用历史训练结果让再次训练分类时更快? 2) 如何在不损失分类精度的前提下抛弃样本点?

文[8]提出的增量学习的主要思想为: 增量训练由 SV 样本和新样本组成, 再训练只需要进行一次即可完成, 所有的非 SV 样本点都被抛弃。这种算法可以表示为下表的算法1, 其中 Step2和 Step3形成了增量训练过程。这种算法的复杂度为:

$$O(\|IS_n \cup INS\| \times \|IS_n\|^2) \quad (10)$$

文[7]提出的算法其结构比较复杂, 它构造了一个再分类-再训练循环, 每次总是把误分样本引入和 SV 样本一起进行训练, 直到误分样本比例小于系统设定的阈值。其具体的算法可以表示表1中的算法2。算法2中 β 为预设的分类精度。我们设 k 为 Step 3到 Step 7重复的次数。则该算法的计算复杂度为(其中 d 为样本集的维度):

$$O((k-1)\|IS_n + INS_{error}\| \|IS_n\|^2 + kd(1-\tau)\|IS\| \|IS_n\|) \quad (11)$$

表1 两种增量学习算法

<p>Algorithm 1: Syed et al.'s Algorithm Step 1. $\Gamma = Train(IS), WS = IS_{sv}$; Step 2. $WS = WS \cap INS$; Step 3. $\Gamma = Train(WS), WS = WS_{sv}$.</p> <p>Algorithm 2: Xiao et al.'s Algorithm Step 1. $\Gamma = Train(IS), WS = IS_{sv}, BS = IS_{non-sv}$ Step 2. $NS = INS, firstime = TRUE$ Step 3. $Classify(NS)$ by $\Gamma, WS = NS_{err} \cup WS, BS = NS_{ok} \cup BS$ Step 4. If $p > \beta$ then Stop Step 5. If $firstime$ then $DS = Discard(BS), BS = BS - DS, firstime = FALSE$ Step 6. $\Gamma = Train(WS), WS = WS_{sv}, NS = BS, BS = WS_{non-sv}$ Step 7. Goto Step3</p>
--

注: 算法1为文[8]提出的增量学习算法, 算法2为文[7]提出的增量学习算法。

3. 基于 SVM 增量学习的用户适应

3.1 特征抽取

在线图形识别中, 用户通过鼠标或绘图笔绘出草图, 我们记录下其笔画。为了简化研究的问题, 特对用户的草图做如下假设:

1. 图形由线条组成, 不存在实心图形。
2. 用户输入的是单笔的、连笔的图形, 即用户的草图是一笔画出来的。
3. 用户输入的图形是封闭图形, 而且是简单的凸多边形(如: 椭圆、三角形、四边形、五边形和六边形)。
4. 不存在起笔或落笔处有勾状的情况(如图2中(c))。
5. 不存在在两条线段的拐角处形成圆圈的情况(如图2中(d))。

但用户在输入草图时, 由于习惯或不小心等原因可能不满足这些假设(如图2中(a), (b), (c), (d)), 因此需要对图形进行预处理使之满足。



图2 预处理中需要纠正的几种图形

另外,在在线图形识别系统中,处理的图元与其大小和位置无关,所以就必须要对记录下的用户的图形进行规一化,即将图形的周长调整为1.然后使用转角函数^[10]对图形进行特征抽取,得到一个d维的特征向量.转角函数 $\Theta_A(s)$ 标识的是图形边界上由初始点到当前点(s)累积的转角.其中 $\Theta_A(O)$ 为选取的初始点O的切线方向与水平方向的夹角.具体如图3所示.

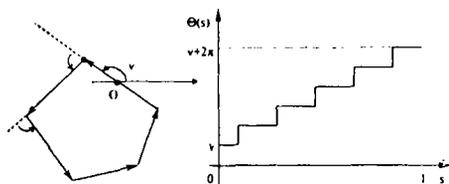


图3 多边形和它的转角函数

在我们的研究过程中,我们对转角函数加了一些限制,规定转角函数的值在 $[0, 2\pi]$ 的区间内,并且初始点的值为0,为了使它简单一些.为此,我们定义如下的转角函数:

$$v_i = \begin{cases} \varphi & \text{if } \varphi \geq 0 \\ \varphi + 2\pi & \text{if } \varphi < 0 \end{cases} \quad 0 \leq i \leq d \quad (12)$$

其中 $\varphi = \Theta_A(i \bmod d) - \Theta_A(i-1)$,d为需要抽取特征的维度,即将图形的边界d等分.

3.2 用户适应

用户在使用在线图形识别系统时,草图总是逐个输入、逐个识别,从机器学习效率的角度来考虑,用户每输入一个图形就对其进行学习,显然是低效的.可是等系统收集到足够的使系统稳定的数据时才开始学习,那必然要让系统长时间在较低精度下运行,显然也是不可行的.所以,在我们学习时用户的数据集的大小有一个下限和上限.当系统开始使用时,数据量积累到下限时(即初始训练集-IS),系统就开始学习,然后等再次收集到一定量的数据(即增量训练集-INS)再开始学习,如此反复系统的适应性就越强.

现在有许多种基于SVM的解决学习的策略,如:

方法1:重复学习,即每次进行学习时,将所有的历史数据都加入到新收集到的数据中进行学习;

方法2:使用算法1,即文[8]的算法进行增量学习;

方法3:使用算法2,即文[7]的算法进行增量学习.

当系统刚开始训练时,数据比较少,那么每次增加的数据与历史数据的比值就比较大,所以这些算法的复杂度就比较困难,而且这时进行复杂度比较的讨论也没有意义.但当系统积累到足够数据时,即每次增加的数据与历史数据的比值很小,也就是说在进行增量学习中的分类边界无重大变化,满足 $\|INS\| \ll \|IS\|$ 的条件,那么:

$$\frac{(10)}{(9)} = \frac{\|INS \cup IS_m\|}{\|IS\|} \approx \frac{\|IS_m\|}{\|IS\|} \quad (13)$$

$$\frac{(11)}{(9)} = \frac{(k-1)\|IS_m\| + \|INS_m\|}{\|IS\|} + \frac{kd(1-\tau)}{\|IS_m\|}$$

$$\approx \frac{(k-1)\|IS_m\|}{\|IS\|} \quad (14)$$

由于 $\|IS_m\| \ll \|IS\|$,从上面两个公式可以看出,Syed算法的效率最高.一般k是在2~5范围内的变量,所以萧昧的算法的效率则要低一些,但仍然要比重复学习算法高很多.需要说明的是,算法2由于被错误分类的点在重新训练的时候仍然可能被错误分类,因此有时会产生不稳定或k值比较大的现象,从而每次增量学习的时间波动比较大,在我们的实验中也体现了这一点.

在实际应用中通常都为多值分类问题,比如在线图形识别系统中的图形分类问题,我们总共处理了椭圆、三角形、四边形、五边形和六边形等5个类别.所以,我们需要构造多值分类器,一般多值分类器的构造有两种模式:一对多模式和一对一模式.在一对一模式下,需要为每两个类别分别构造分类器,在测试时,每个分类器都计算一次,然后投票,最后得票最多的类别就被选为正确的分类^[9].而对于一对多模式,每一个分类器分别将某一分类的数据从其它分类的数据集数据中鉴别出来.表2是这两种模式构造的多值分类器的性能比较.

表2 两种模式构造的多值分类器的性能对比(其中m为样本类别数,n为样本的测度)

模式	一对一模式	一对多模式
训练	低,平均为 $O(n^2/m)$	高,平均为 $O(mn^2)$
分类	需要 $m(m-1)/2$ 次分类	需要m次分类
VC 维度	相对比较低	相对很高,易于过量匹配
增加一个新样本时的复杂度	低,重新训练m-1个子分类器 计算复杂度为 $O(n)$	高,重新训练所有分类器 计算复杂度为 $O(mn)$

因此,使用基于SVM增量学习的方法来实现用户适应,整个学习的过程可以表示为“识别(收集数据)—学习”循环反复的过程.系统收集用户的输入信息,并对其进行规整,然后使用转角函数将其抽取为特征向量.当系统收集的数据量达到一定量时,就开始进行学习.对每两个类别训练分类器,共需要训练10个分类器,并记录下它们的特征向量,即训练多值分类器.然后系统利用学习的结果来对用户输入的草图进行识别,用所有的分类器对用户的数据进行分类,通过投票决定具体的类别,即利用训练好的多值分类器来进行分类.用户通过界面判断系统分类的正确与否.系统边识别边收集用户数据和分类结果,当收集的数据满足一定量时,将特征向量加入到新收集到的数据集,然后再训练多值分类器.如此反复,系统逐渐趋于稳定,系统也就适应了多用户.

4. 实验结果与分析

4.1 实验环境

在实验中,我们总共收集了1367个用户1的图形和325个用户2的图形.每个用户的图形都分为5个类别,并且所有的图形都是使用转角函数来抽取20维的特征.除了这些数据之外,

我们还通过旋转和翻转来获得虚样本:

(1)旋转:在图形分成20个等分之后,每次取样时选取一个不同初始点,这样就获得19个虚样本。

(2)翻转:在旋转获得的数据,通过反转20维特征的顺序,这样在每一个实、虚样本上又获得一个虚样本。

关于特征抽取和样本变换的详细资料,可以参考文[2]。

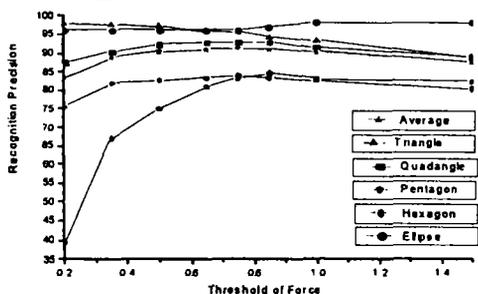
通过这些变换,每一个样本最多可以获得39个虚样本,这样总共就要54680个用户1的数据和13000个用户2的数据(所有的这些数据包括整个实验的程序在 http://jxy.hanweb.com/research/shape_classes.zip 可以下载得到)。随机抽取19210个用户1的数据作为测试数据 T_1 ,将所有的13000个用户2的数据都作为测试数据 T_2 。将剩下的用户1的数据分成40个增量训练集 S_1, S_2, \dots, S_{40} 。由于支持向量机在比较少的样本的情况下可以获得比较好的效果,而且在机器开始学习的时候一般数据量比较少,因此将开始的几个数据集的数

据取得比较小是很合理的。然而由于增量学习过程中,系统的学习时间将随着增量训练集的尺度增长而急剧增长,所以在学习过程中我们控制了每次增量训练集的最大上限(1000个样本)。在实验中,我们选取的前六个增量数据集的尺度分别为100、100、120、150、300、700,剩下的34000个均分到34个训练集里。当然每个训练集的选取都是随机的。需要说明的是,在实验中由于算法2在数据量很少的情况下可能出现抖动,所以将 $S_1 + S_2$ 作为算法2初始训练集。

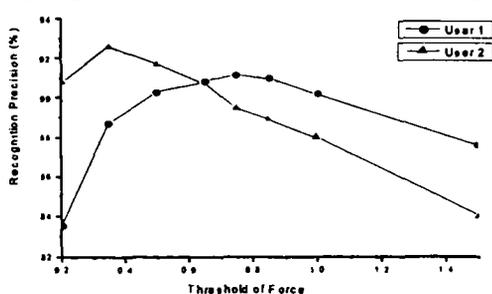
在整个实验过程中,使用的是径向基核函数(Radial Basic Function, RBF)^[4],在训练过程中使用的训练方法与 SVM Torch^[11]相同。所有的实验都是在操作系统为 Microsoft Windows 2000的 Intel P IV PC (1.4G Hz CPU、256MB memory)上进行的。

4.2 基于规则的相关反馈

我们首先使用了基于规则的相关反馈的方法进行自适

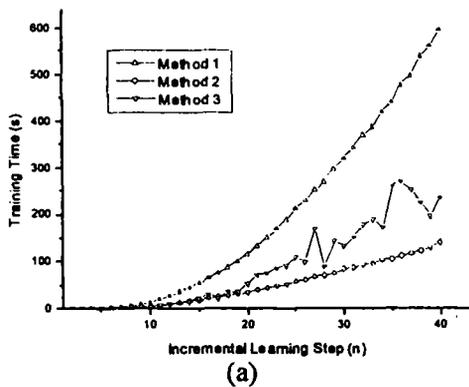


(a) 用户1的5个类别识别精度和平均精度随阈值变化的结果图

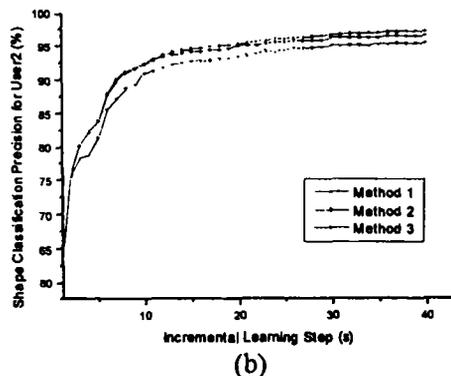


(b) 不同用户的识别精度随阈值变化图

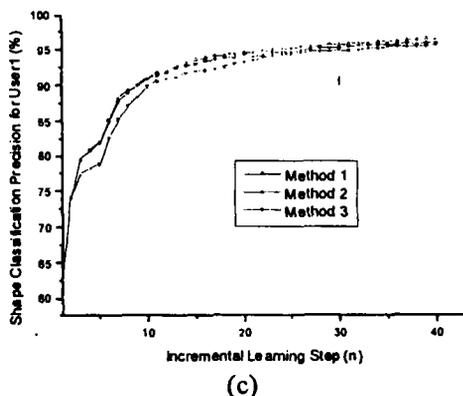
图4 基于规则的相关反馈的多用户适应结果图



(a)



(b)



(c)

图5 性能评价。(a)训练时间对比;(b)用户1的图形分类精度对比;(c)用户2的图形分类精度对比。

在使用文[7]进行增量学习,其中 $\beta=0.99$, $\tau=0.50$ (这样设置参数可以获得最好的性能)

应,在研究中,发现随着阈值由小到大的调整每个用户的平均精度都是先提高然后降低,并且对于不同的用户它们达到精

度最大值时的阈值一般都不相同,对于用户1在阈值为0.75的时候效果最好(91.2%),对于用户2则在阈值为0.35时达到最好(92.6%),对于用户1和用户2而言则在阈值为0.65时达到最好(90.8%),也就是说通过调整阈值使系统适应于多用户不是很合适,见图4的实验结果图。

4.3 基于SVM的增量学习

在实验中,我们在相同的数据集上对比了上述的三种方法。总共有40个增量训练数据集,我们对比了训练时间、对用户1的测试集 T_1 的开放测试和对用户2的测试集 T_2 的开放测试,分别如图5(a),(b),(c)所示。

4.4 性能分析

从实验的比较来看,在在线图形识别系统中我们可以得出以下结论:

1. 增量学习可以适应多个用户。在基于SVM的学习过程中,不存在对于不同的用户的冲突现象。在基于规则的相关反馈的实验中,对于用户1和用户2而言在阈值为0.65时达到最好,平均精度为90.8%。而在以方法2为例,用户1和用户2的分类精度随着增量学习次数的增加都得到提高,当用户1的精度达到95.92%时,用户2的精度达到了96.56%,平均精度为96.18%。图6是使用方法2的用户1和用户2精度对比。

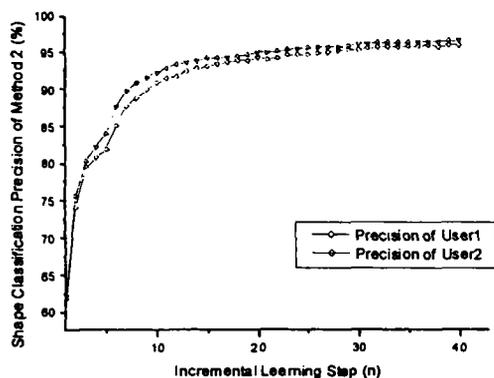


图6 基于方法2的多用户适应

2. 增量学习优于重复学习,即方法2、3优于方法1。在开始学习时,由于数据量比较小,而且增加的增量训练集与已训练的数据集的比值比较大,增量学习的优势不是很明显。而当数据量积累到很大时,增量训练集与历史数据相比的权值比较小时,增量学习的效率要比重复学习的高得多,如图5中(a)所示。但是增量学习的精度并没有因为效率的提高而损失太多,不管是对用户1还是对用户2,它们的精度都与重复学习的精度相差不多,如图5中的(b)和(c),尽管增量学习的精度都比重复学习要低一些。

3. 文[8]中的算法优于文[7]中的算法,即方法2优于方法3。正如我们上面分析的一样,我们的实验(如图5中(a))也得

出了方法2的效率高于方法3。另外,图5(a)中方法2训练时间的增长总是稳步的,而方法3的变换有时则显得比较突然,这主要是由于在方法3中存在不稳定的系数 k 所致。虽然它们的精度相差不多,但是方法3的精度比方法2的高还是可以看出来,如图5(b)和(c)。

结束语 本文主要针对多用户适应性问题,提出了基于SVM增量学习的解决方案,并且分析和对比了增量学习与重复学习、两种增量学习算法各自存在的优缺点。研究表明基于SVM增量学习的方法优于规则的方法、增量学习优于重复学习、文[8]的算法要好于文[7]的算法。在 Smart Sketchpad 实验系统中,我们既实现了使用基于规则的相关反馈方法(其精度一般可以达到90.8%),也实现了基于SVM的增量学习方法(其精度一般可以达到96.2%)。基于规则的相关反馈方法实现简单,计算时空复杂度低,适于单用户的便携环境,因为这些设备的计算能力有限而且内存也不大;而基于SVM的增量学习方法则适合于多用户的、有较强计算能力的桌面环境。

参考文献

- Dix A, et al. Human-Computer Interaction. Englewood Cliff, NJ: Prentice Hall, 1993
- Liu W, Jin X, Qian W, Sun Z. Inputting Composite Graphic Objects by Sketching A Few Constituent Simple Shapes. In: Proc. of GREC2001, Kingston, Canada, 2001. 73~84
- Jameson A. Adaptive interfaces and agents. In Julie A. Jacko and Andrew Sears, editors, Handbook of Human-Computer Interaction in Interactive Systems. Erlbaum, Mahwah, NJ, 2002. In press
- Vapnik V N. The Nature of Statistical Learning Theory. NY: Springer-Verlag, 1995
- 张学工译. 统计学习理论的本质. 北京:清华大学出版社, 1999
- Christopher J, Burges C. A tutorial on support vector machines for pattern recognition. Knowledge Discovery and Data Mining, 1998, 2(2): 235~244
- Xiao R, Wang J, Zhang F. An Approach to Incremental SVM learning algorithm. In: Proc. the 12th Intl. Conf. on Tools with Artificial Intelligence, 2000. 268~273
- Syed N, Liu H, Sung K K. Incremental Learning with Support Vector Machines. In: Proc. Workshop on Support Vector Machines at the Intl. Joint Conf. on Artificial Intelligence (IJCAI-99), Stockholm, Sweden, 1999
- Weston J, Watkins C. Multi-class Support Vector Machines. [Technical Report CSD-TR-98-04]. Department of Computer Science, University of London, 1998
- Esther M A. An Efficient Computable Metric for Comparing Polygonal Shapes. IEEE Trans. on PAMI, 1999. 13(3): 209~216
- Collobert R, Bengio S. SVMtorch: Support vector machines for large-scale regression problems. Journal of Machine Learning Research, 2001. 143~160