

用支持向量机为学习过程建模

肖 瑜, 刘宏义

(西安陆军学院, 陕西 西安 710108)

摘 要: 在游戏开发的人工智能领域中, 最关键的部分是学习过程. 经典的学习系统需要花费很多时间忘记旧规则, 同时学习新规则. 为了解决这个问题, 文中提出使用支持向量机这种技术, 并为这种学习分类器引进了一个时间概念, 这样, 系统就可以很容易地忘记过去的事情. 同时也为支持向量的数量限制和 CPU 消耗的限制提出了解决方法.

关键词: 游戏开发; 人工智能; 支持向量机

中图分类号: TN911.22

文献标识码: A

文章编号: 1000-7180(2011)11-0197-04

Using Support Vector Machines to Model the Learning Process

XIAO Yu, LIU Hong-yi

(Xi'an Military Academy, Xi'an 710108, China)

Abstract: In the field of the artificial intelligence for game development, the learning process is very important. Classical learning process will spend lots of time forgetting the old rules, and at the same time learning the new rules. In order to solve the problem, this paper put forward the method of Support Vector Machines (SVM). By introducing the time concept for this method, system can easily forget the old rules, and also bring forward an effective solution to the amount limit of SVM and CPU's consuming.

Key words: game development; artificial intelligence; SVM

1 引言

在当前的游戏开发中, 最具有挑战性的工作是人工智能的编程. 其中最关键的部分就是学习过程. 当游戏玩家不断地变化玩游戏的策略, 学习系统就会遇到很多的困难. 经典的学习系统需要花费很多时间忘记旧规则, 同时学习新规则. 文中为学习分类器引进了一个时间概念, 这样, 系统就可以很容易地忘记过去的事情. 在这里提出支持向量机(Support Vector Machines, SVM), 这种技术可以应用于其他类型的分类器.

2 支持向量机

支持向量机(SVM)是学习分类器. 例如, 神经网络学习分类器. 但是它们使用的概念略有不同.

SVM 是一个线性分类器的概括, 与神经网络相比, SVM 的优越性是它的创建和训练算法.

2.1 线性分类器

在超平面的帮助下, 线性分类器可以把输入空间分成两个部分: 类 A 为类 B. 位于超平面上部的每一点属于类 A, 面位于超平面下部的每一点属于类 B. 图 1(a) 显示了一个已分类的数据集. 在方程(1)定义的平面 D 是分类器的唯一参数. 如果 x_i 被分为类 A 和类 B, 如果分类器是经过训练的, 这些数据的分类可以由方程(2)和方程(3)来实现.

(x, y) 代表了标准点积 $(x, y) = x_1 \cdot y_1 + x_2 \cdot y_2 + \dots$.

$$D \equiv (w, x) + d = 0 \quad (1)$$

$$x_i \in A \Leftrightarrow (w, x_i) + d > 0 \quad (2)$$

$$x_i \in B \Leftrightarrow (w, x_i) + d < 0 \quad (3)$$

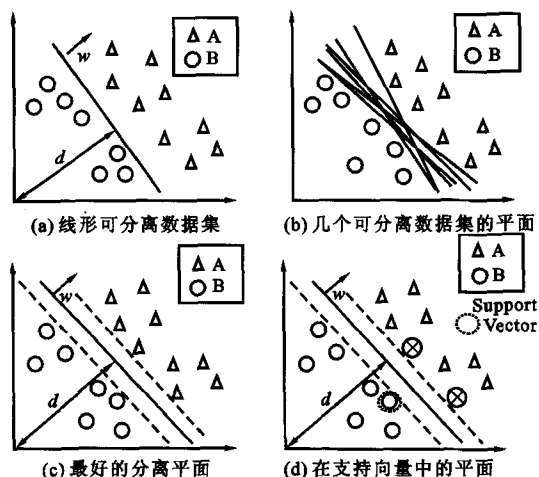


图1 SVM解决平面问题

然而,由两个约束条件组成系统的训练并非无足轻重.如果具备方程(4)和方程(5),那么分类的条件就变成了方程(6).

$$y_1 = 1 \Leftrightarrow x_i \in A \quad (4)$$

$$y_1 = -1 \Leftrightarrow x_i \in B \quad (5)$$

$$y_i \cdot ((w, x_i) + d) > 0 \quad (6)$$

因此,就像方程(6)验证的每个点,训练就是要寻找 w 和 d .但是,这里存在着无数个分离平面,图1(b),需要从这些平面中选择一个.尽量选择一个最能概括分类器的平面,也就是说,使用一些没有包含在训练集合中的数据,也能得到比较好的结果.就像在图1(c)显示的那样,有两个分离平面,每一个平面对应一个 y_i 值(1或-1),平面间的距离称为边界.可以表明,边界是与 $\frac{1}{|w|}$ 成比例的^[1].

为了获得更概括的分类器,必须把边缘最大化,或者最小化 $|w|$,这将被用于训练算法.

$$y_i \cdot ((w, x_i) + d) \geq 0 \quad (7)$$

验证方程(8)的那些点 x_i 称为支持向量.这些点支持我们这个解决方法,它们分布在限定平面上,并且位于它们类的边缘.

$$y_i \cdot ((w, x_i) + d) = 0 \quad (8)$$

可以证明,向量 w 是支持向量的一个线性组合^[2].方程(8)和方程(9)的组合可被改写为方程(10),现在分类的条件与训练数据无关.

$$w = \sum \alpha_j y_j x_j \text{ 且 } \alpha_j \geq 0 \quad (9)$$

$$\sum \alpha_j y_j ((w, x_i) + d) \geq 1 \quad (10)$$

现在的训练在于寻找 α_i 和 d .每一个不能验证方程(8)的点,系数 α_i 将是空的.因此可以把它从解决方法中移走,这是训练所要达到的一个目标.支持向量及其相应的系数作为分类器的参数.

图1(d)表明的解决平面都依赖于支持向量,具有如下特点:因为它们都共享法线 w ,这些平面必定互相平行.可以证明,去掉一个非支持向量的向量,解决方法并不改变.这是合乎逻辑的,因为它们的 α_i 是空的,所以不会改变分类器的参数.

如果想用一个新数据集来训练系统,没有必要添加整个旧数据集,需要的仅仅是旧数据集的支持向量.

2.2 非线性可分离数据集的扩展

如果输入空间被转换成一个高维空间,非线性可分离数据集也可以变成线性可分离的.图2就显示了这样一个空间转化的概念.

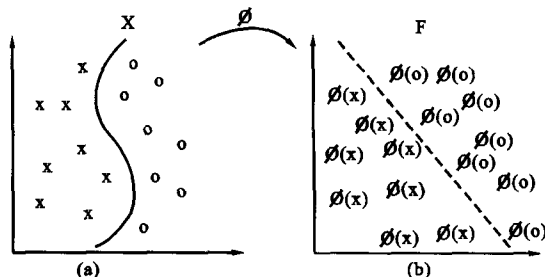


图2 输入空间转换

图3(a)显示了一个具体的例子.如果试图用线性分类器为这些数据集分类的话,无法获得分离平面.如果在方程(11)的帮助下,把一维的空间转换成一个二维的空间,那么这时候数据集就是线性可分离的.图3(b)表示转换过程和一个成功的分离平面.这个例子虽然简单,但也表明这种转化为更高维空间的神奇之处.

$$\phi(x) = (x, x^2) \quad (11)$$

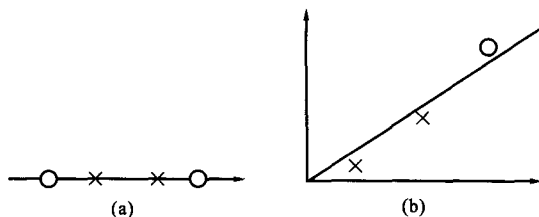


图3 转换高维空间的例子

2.3 核函数

如果转换应用于每一个向量 x_i 和 w ,方程(10)就变成了方程(12).

$(\phi(x_i), \phi(x))$ 是两个转换向量的点积.尽管转换空间已成为高维的,但返回值将总是一个标量值.可以把 $(\phi(x_i), \phi(x))$ 当成一个简单的函数,其功能就是要返回一个标量.这个函数称为 $\text{kernel}(K)$ (核函数).方程(13)给出了这个函数 $K(x, y)$ 定义.现在不需要知道 $\phi(x)$ 的值.此时的

$\phi(x)$ 也可以转换数据到一个无限维度空间(核心指数是这种转换的一个例子). 作为一个例子, 方程(11) 表示的等价核心的转换将在方程(14) 中计算. 重写分类条件, 最后形式如方程(15) 所示.

$$\sum x_i y_i (\phi(x_i), \phi(x)) + d \geq 0 \quad (12)$$

$$K(x, y) = (\phi(x), \phi(y)) \quad (13)$$

$$K(x, y) = \left\{ \begin{pmatrix} x \\ x^2 \end{pmatrix}, \begin{pmatrix} y \\ y^2 \end{pmatrix} \right\} = x \cdot y + x^2 \cdot y^2 \quad (14)$$

$$\sum a_i y_i K(x_i, y) + d \geq 0 \quad (15)$$

核心函数必须验证在文献[2] 中可以看到的一些条件. 这种情况下, 一些 $\phi(x)$ 与核心函数匹配. 方程(16) 中的核心函数具有如下的一些重要属性.

$$\begin{cases} K(x, y) = k(y, x) \\ K(x, y) = k_1(x, y) + k_2(x, y) \\ K(x, y) = a k_1(x, y) \text{ with } a > 0 \\ K(x, y) = k_1(x, y) \cdot k_2(x, y) \\ K(x, y) = f(x) \cdot f(y) \\ K(x, y) = k_3(\phi(x), \phi(y)) \end{cases} \quad (16)$$

方程(17) 展示了一些经典的核心函数. 它们可以同方程(16) 中显示的属性相结合.

$$\begin{cases} k(x, y) = \exp(-\|x - y\|^2 / 2\sigma^2) \\ k(x, y) = (x, x)^n \\ k(x, y) = ((x, y) + 1)^n \end{cases} \quad (17)$$

2.4 训练

如前所述, 在方程(7) 的约束下, 训练在于寻找 w 和 b . 使用方程式(9), 在方程(15) 的约束下, 训练就是要寻找 a_i 和 b . 目标就是要找到一个最概括的分离平面. 必须最大化边界, 或者最小化 $|w|$. 于是算法训练就变成了方程(18). 这是一种有条件的优化.

$$\min_w \frac{1}{2} \|w\|^2 \text{ with } \sum a_i y_i K(x_i, x) + d \geq 0 \quad (18)$$

应用拉格朗日优化理论, 方程(18) 就变成了方程(19)^[3]. 方程(19) 是一种二次优化问题.

$$\max_a \sum_i a_i - \frac{1}{2} \sum_{i,j} a_i a_j y_i y_j K(x_i, x_j) \quad (19)$$

二次优化理论表明问题是凸起的, 不存在局部最小化问题. 在这个优化问题中, 通常可以找到最好的结果. 这是支持向量机(SVM) 的一个重要优势: 当给定数据集并选中核心函数后, 通常可以找到最好的结果. 多次训练这个系统都是得到一个完全相同的结果. 虽然二次优化算法有很多种, 但已超出论

述范围. 此处, 文中仅介绍了支持向量机, 更多的信息可以在文献[4] 中找到.

3 短期记忆模型化

分类器系统是用收集的数据集进行训练的. 在一个神经网络中, 不可能决定或者知道何时网络忘记过去学到的一些东西. 在支持向量机(SVM) 中, 分类是支持向量和它们的系数 a_i , 见方程(9) 描述的. 这些向量是收集到的数据集的一部分. 这样就有可能给每个向量标定日期. 然后, 经过一定的时间段, 为了取消向量的影响, 可以把它从解法集中移走. 整个数据集就可以不断地减少最老的向量, 增加最新的向量. 图4 显示了移走一个过时的支持向量时, 相应的解法集的变化情况. 图5 显示了添加一个新的支持向量时, 相应的解法集的变化情况.

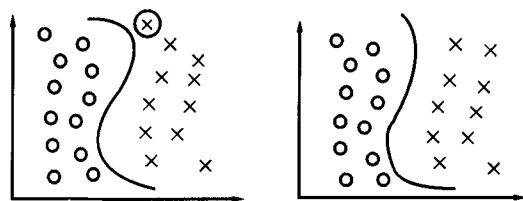


图4 移除一个过时的向量

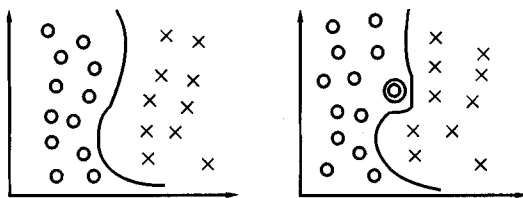


图5 添加一个新的支持向量

一方面, 只要支持向量没有改变(或者变化的新老数据都不是支持向量), 训练就会给出完全一样的结果, 解法集也不会改变, 并且训练速度也非常快. 另一方面, 如果移走一个支持向量, 与第一情况相比, 训练需要的时间会更长一些, 但是训练速度也很快, 因为非支持向量的数据权重(a_i) 仍然为零.

系统派生可能会是一个问题, 而且除了支持向量和它们的系数 a_i , 解法也受到所选择使用的核心函数的约束. 虽然这或许是一个优点, 或许是一个缺点, 因为要选择这样的核心就可能非常棘手. 这种技术在一个充满变化的环境中相当有用, 因为过去学习到的知识对现在不会有无止境的影响. 对于神经网络, 如果环境发生了变化, 一些决策可能就不再有效. 而对于带有标定日期的支持向量机(SVM), 只会有一次错误或一次长时间的等待(例如, 过时的向量).

虽然只在 SVM 中介绍了这种技术, 显然, 这种

技术也可适用到其他的分类器,只要分类器是从训练数据集中得到的数据来表示它们的参数,例如最近邻算法(K-Nearest Neighbors, KNN). 在加强学习方面,这种 SVM 可以代替神经网络.

4 CPU 的消耗限制

评估点 x 的值,需要计算 $K(x, y)$ 的值和每个支持向量的两个乘积.

$$\sum \alpha_i y_i K(x_i, x) \quad (20)$$

重要的是限制分类器对 CPU 的消耗. 降低 CPU 消耗的唯一方法是限制支持向量的数目,并限制到一个固定的值. 使用这些带有日期的系统,支持向量数目太高时,最老的支持向量在评估时可以抛开不计. 然而,抛开不计的支持向量不能从训练数据集中移走. 如果是这样的话,支持向量必将被另外一个取代. 这种行为将会降低分类器的有效性. 在游戏中,限制支持向量的数目是非常重要的,因为它在决策中引入了一些噪音,所在效率下降并非总是坏事.

5 结束语

文中介绍了支持向量机(SVM),并引出了老化数据的概念. 尽管这一概念可以用于大多数的分类

器系统,但 SVM 是个很好的候选者,因为分类器的参数是用训练数据集来表示的. 它还支持向量的数量限制和 CPU 消耗的限制提出了解决方法.

参考文献:

- [1] Moore, Andrew W. support vector machines[EB/OL]. [2011-02-15]. Available online at <http://www.autonlab.org/trtorial/svm.html>.
- [2] Burges, Christopher. A tutorial on support vector machines for pattern recognition[J]. knowledge Discovery and Data Mining, 1998(2):955-974.
- [3] Cristianini, Nello. Tutorial on support vector machines and kernel methods[EB/OL]. [2011-02-15]. ICML-2001 Tutorials. Available online at <http://www.support-vector.net/icml-tutorial.pdf>.
- [4] Gould Nick, Philippe Toint A quadratic programming page[EB/OL]. [2011-02-15]. Available online at <http://www.numerical.rl.ac.uk/qp/qp.html>.

作者简介:

肖 瑜 男,(1964-),教授. 研究方向为作战仿真与虚拟现实.

刘宏义 男,(1972-),讲师. 研究方向为作战仿真与虚拟现实.

(上接第 196 页)

toward feature space analysis[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence 2002,24(5): 603 - 619.

- [4] Kanazawa Y, Kawakami H. Detection of planar regions with uncalibrated stereo using distributions of feature points[C]//British Machine Vision Conference. [s. l.]: IEEE, 2004: 247 - 256.
- [5] Zuliani M, Kenney C S, Manjunath B S. The multi

RANSAC algorithm and its application to detect planar homographies[C]// Proceedings of the IEEE International Conference on Image Processing, Genova: IEEE, 2005: 11-14.

作者简介:

于永彦 男,(1969-),副教授,博士研究生. 研究方向为模式识别、多媒体通信.