

文章编号:1001-9081(2009)06-1612-03

光滑支持向量机两种求解算法的比较

李广明, 刘群峰

(东莞理工学院 计算机学院, 广东 东莞 523808)
(lgnmg@163.com)

摘要: BFGS-Armijo 法和 Newton-Armijo 法是求解光滑模型的常用算法。分别列出用此两种算法求解光滑支持向量机模型 (SSVM) 的具体步骤, 并用这两种算法对求解 SSVM 模型进行比较研究。数值实验结果表明: Newton-Armijo 法的分类性能和 BFGS-Armijo 法基本相同, 而分类效率比 BFGS-Armijo 法高出约 26.2%。

关键词: 分类; 支持向量机; Newton-Armijo 法; BFGS-Armijo 法

中图分类号: TP18 **文献标志码:** A

Comparison between two solving methods for smooth support vector machine

LI Guang-ming, LIU Qun-feng

(Computer College, Dongguan University of Technology, Dongguan Guangdong 523808, China)

Abstract: The BFGS-Armijo method and Newton-Armijo method are two popular methods to solve the smooth models. The authors respectively listed the procedures of these two methods to solve Smooth Support Vector Machine (SSVM), and made a comparative study between the two methods to solve SSVM. The numerical results show that the classification performance of BFGS-Armijo method is almost the same as the Newton-Armijo method, but the classification efficiency of Newton-Armijo method is 26.2% higher than that of BFGS-Armijo method.

Key words: classification; Support Vector Machine (SVM); Newton-Armijo method; BFGS-Armijo method

0 引言

支持向量机 (Support Vector Machine, SVM) 是数据挖掘的一种新方法^[1]。它广泛用于分类问题和回归问题, 特别应用在文本分类、图像检索、人脸识别、语音识别和手写数字识别等领域, 因此有很多学者对 SVM 算法开展了深入研究。对分类问题的 SVM 算法, 最具代表性的算法是: SMO 算法^[2]、SVM^{light} 算法^[3]、SOR 算法^[4]、SSVM 模型^[5] 和 PSSVM^[6]。特别值得一提的是 SSVM 模型: 采用 Newton-Armijo 法进行求解的结果表明, SSVM 的求解效果优于 SOR 算法、SMO 算法, 以及 SVM^{light} 等算法。文献[7-8]提出一类逼近精度更好的多项式光滑函数和多项式光滑的支持向量机一般模型。通过研究人员的不断努力, 光滑支持向量机的理论得到了一定发展, 然而, 作为一个新的研究方向, 还存在许多尚未解决的理论问题, 其中包括两个问题: 1) BFGS-Armijo 法和 Newton-Armijo 法都是求解光滑模型的常用算法, 哪个算法更有效? 2) 如果其中一个算法更好, 那么该算法的分类性能比另一个算法好多少? 计算效率高多少? 本文针对这两个问题进行研究, 分别列出这两种算法求解 SSVM 模型的具体步骤, 对数值实验结果进行对比分析。

1 光滑支持向量机的原理和模型

给定 R^n 中 m 个点 A_1, A_2, \dots, A_m , 用矩阵 $A_{m \times n}$ 表示, 目的是将 A_1, A_2, \dots, A_m 分为 A^+ 和 A^- 两类。若 A_i 属于类 A^+ , 记为 1; 若属于 A^- , 记为 -1, 则可以用一个 $m \times m$ 的对角阵 D 来表示分类情形, 其中 D 的对角元素为 1 或 -1。该分类问题的标准 SVM 模型为^[5-6]: 对某个 $v > 0$, 有:

$$\min_{(\omega, \gamma, y) \in R^{n+1+m}} v e^T y + \frac{1}{2} \omega^T \omega$$

$$\text{s. t. } D(A\omega - e\gamma) + y \geq e, y \geq 0, \quad (1)$$

其中 e 表示分量全部为 1 的列向量。 ω 是如下的边界面的 n 维法线向量:

$$\begin{cases} x^T \omega - \gamma = 1 \\ x^T \omega - \gamma = -1 \end{cases} \quad (2)$$

而 γ 决定了边界面到原点的距离, y 为松弛变量, 当类 A^+ 和 A^- 是严格线性可分时, $y = 0$ 。 $x^T \omega - \gamma = 1$ 和 $x^T \omega - \gamma = -1$ 分别是类 A^+ 和 A^- 的边界, 此时分类的超平面为:

$$x^T \omega = \gamma \quad (3)$$

该平面处于式(2)的两平面之间, 如图 1 所示。如果类 A^+ 和 A^- 是线性不可分的, 则可以引入一个松弛变量 y ($y \geq 0$), 使得: 若 $x^T = A_i, D_{ii} = 1$, 则 $x^T \omega - \gamma + y_i \geq 1$; 若 $x^T = A_i, D_{ii} = -1$, 则 $x^T \omega - \gamma + y_i \leq -1$ 。

在标准的 SVM 模型(1)中, 目标函数对松弛变量 y 取 1-范数, 并用 v 进行加权。如对目标函数进行修正, 则可以得到如下的修正模型^[4-6]:

$$\min_{(\omega, \gamma, y) \in R^{n+1+m}} \frac{v}{2} y^T y + \frac{1}{2} (\omega^T \omega + \gamma^2)$$

$$\text{s. t. } D(A\omega - e\gamma) + y \geq e, y \geq 0 \quad (4)$$

令

$$y = (e - D(A\omega - e\gamma))_+ \quad (5)$$

其中: $(\cdot)_+$ 为正号函数, 其第 i 个分量定义为 $(x_+)_i = \max\{0, x_i\}$ 。将式(4)中的 y 用式(5)代替, 就把有约束的支持向量机模型式(4)变成了无约束的支持向量机模型:

收稿日期: 2008-12-22; 修回日期: 2009-03-23。

基金项目: 国家自然科学基金资助项目 (60773050); 广东省科技计划项目 (2008B060600076)。

作者简介: 李广明 (1968-), 男, 河南信阳人, 讲师, 硕士, 主要研究方向: 人工智能、数据挖掘; 刘群峰 (1978-), 男, 江西赣州人, 讲师, 博士研究生, 主要研究方向: 多重网格计算、最优化方法。

$$\min_{(\omega, \gamma)} \frac{1}{2} v \| (e - D(A\omega - e\gamma))_+ \|_2^2 + \frac{1}{2} (\omega^T \omega + \gamma^2) \quad (6)$$

该优化问题的目标函数不具有光滑性,不能用快速的算法求解。

文献[5]引入光滑技术,用一个 Sigmoid 函数的积分函数 $p(x, \alpha) = x + (1/\alpha) \log(1 + e^{-\alpha x})$ ($\alpha > 0$, e 是自然对数的底) 对上述模型(6) 作光滑处理, 得到光滑支持向量机模型 (Smooth Support Vector Machine, SSVM):

$$\min_{(\omega, \gamma) \in \mathbb{R}^{n+1}} \frac{v}{2} \| p(e - D(A\omega - e\gamma), k) \|_2^2 + \frac{1}{2} (\omega^T \omega + \gamma^2) \quad (7)$$

引理^[5] 设光滑支持向量机模型 SSVM 的解和原支持向量机模型(6) 的解分别为 \bar{x}_k 和 \bar{x} , 则 $\lim_{k \rightarrow +\infty} \bar{x}_k = \bar{x}$ 。

该引理表明, 当 k 趋于正无穷时, 光滑支持向量机模型的解收敛于原支持向量机模型(6) 的解。显然, 该模型的目标函数具有任意阶光滑, 用 BFGS-Armijo 法和 Newton-Armijo 法都可进行求解^[1]。

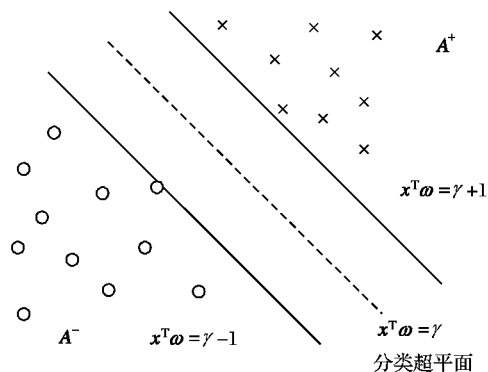


图1 线性可分的 SVM 分类示意图

2 求解 SSVM 的步骤

2.1 BFGS-Armijo 法

BFGS-Armijo 算法的适用条件是目标函数具有一阶光滑性^[1]。显然式(7)的目标函数可采用 BFGS-Armijo 算法进行求解。记式(7)的目标函数为 $f(x): \mathbb{R}^n \rightarrow \mathbb{R}$, 其一阶导数记为列向量 $g(x)$, 用 i 表示迭代步数。 ε 为算法本身的求解精度, 迭代点列记为 $x^{(1)}, x^{(2)}, \dots, x^{(i)}, \dots$ 。为便于表述, 记 $g^{(i)} = g(x^{(i)})$, $f^{(i)} = f(x^{(i)})$, 记 x^* 为最终解。于是, 用 BFGS-Armijo 算法解光滑支持向量机模型 SSVM 的具体步骤为^[1,6]:

第1步 给定初始阵 $H^0 = I$, 初始点 $x^{(0)}$, 置精度要求 $\varepsilon > 0$, 令步数 $i = 1$ 。

第2步 计算 $g^{(i)}, f^{(i)}$ 。

第3步 若 $\|g^{(i)}\| \leq \varepsilon$, 则停机, 取 $x^* = x^{(i)}$, 否则由方程(8) 计算下降方向 $d^{(i)}$ 。

$$d^{(i)} = H^i g^{(i)} \quad (8)$$

第4步 Armijo 步长 $\alpha_i; s, \beta, \sigma$ 取固定值, 其中 $s > 0, \beta \in (0, 1), \sigma \in (0, \frac{1}{2})$, m_i 取使下式成立的最小正整数 m : $f(x^{(i)}) + f(x^{(i)} + \beta^m s d^{(i)}) \geq -\sigma \beta^m s d^{(i)T} g^{(i)}$, 得步长 $\alpha_i = \beta^{m_i} s$ 。

第5步 若 $\alpha_i < 10^{-12}$, 则停机, 取 $x^* = x^{(i)}$; 否则, 计算 $x^{(i+1)} = x^{(i)} + \alpha_i d^{(i)}, \delta^{(i)} = \alpha_i d^{(i)}$, 计算 $f^{(i+1)}, g^{(i+1)}$ 。令 $y^{(i)} = g^{(i+1)} - g^{(i)}$ 。

第6步 计算 H^{i+1} :

$$H^{i+1} = H^i + \frac{\delta^{(i)} \delta^{(i)T}}{y^{(i)T} \delta^{(i)}} - \frac{H^i y^{(i)} (H^i y^{(i)})^T}{y^{(i)T} H^i y^{(i)}} + \frac{(y^{(i)T} H^i y^{(i)}) t^{(i)} t^{(i)T}}{t^{(i)T} t^{(i)}} \quad (9)$$

其中:

$$t^{(i)} = \frac{\delta^{(i)}}{y^{(i)T} \delta^{(i)}} - \frac{H^i y^{(i)}}{y^{(i)T} H^i y^{(i)}} \quad (10)$$

第7步 置 $i = i + 1$, 转第3步继续迭代。

从式(8) 可以看出, 原支持向量机模型(6) 作光滑处理后, 运用 BFGS-Armijo 算法, 只需要解一个线性方程组。不需要像传统的 SVM 那样, 解一个非线性规划问题。于是就把非线性问题成功地转化成了线性问题。

2.2 Newton-Armijo 法

Newton-Armijo 算法的适用条件是目标函数具有二阶光滑性^[1]。显然式(7)的目标函数可采用 Newton-Armijo 算法进行求解。

记式(7)的目标函数为 $f(x): \mathbb{R}^n \rightarrow \mathbb{R}$, 其一阶导数记为列向量 $g(x)$, 其二阶导数矩阵记为 $G(x)$, 即海色矩阵, 用 i 表示迭代步数。 ε 为算法本身的求解精度, 迭代点列记为 $x^{(1)}, x^{(2)}, \dots, x^{(i)}, \dots$ 。记 $g^{(i)} = g(x^{(i)})$, $G_i = G(x^{(i)})$, $f^{(i)} = f(x^{(i)})$, 记 x^* 为最终解。于是, 用 Newton-Armijo 算法解光滑支持向量机模型 SSVM 的具体步骤为^[1-6]:

第1步 取初始点 $x^{(1)}$, 置该算法本身的精度要求 ε , 令 $i = 0$ 。

第2步 置 $i = i + 1$, 计算 $g^{(i)}$, 若 $\|g^{(i)}\| \leq \varepsilon$, 则停机, 得最终解 $x^* = x^{(i)}$ 。

第3步 计算 $G_i = G(x^{(i)})$, 由方程(11) 求解下降方向 $d^{(i)}$ 。

$$G_i d^{(i)} = -g^{(i)} \quad (11)$$

第4步 Armijo 步长 $\alpha_i; s, \beta, \sigma$ 取固定值, 其中 $s > 0, \beta \in (0, 1), \sigma \in (0, \frac{1}{2})$, m_i 取使下式成立的最小正整数 m : $f(x^{(i)}) + f(x^{(i)} + \beta^m s d^{(i)}) \geq -\sigma \beta^m s d^{(i)T} g^{(i)}$, 得步长 $\alpha_i = \beta^{m_i} s$ 。

第5步 若步长 $\alpha_i < 10^{-12}$, 则停机, 取 $x^* = x^{(i)}$; 否则令 $x^{(i+1)} = x^{(i)} + \alpha_i d^{(i)}$, 转第2步继续迭代。

从式(9) 可以看出, 原支持向量机模型(6) 作光滑处理后, 运用 Newton-Armijo 算法, 也只需要解一个线性方程组。不需要像传统的 SVM 那样, 解一个非线性规划问题。于是也把非线性问题成功地转化成了线性问题。

3 实验

为方便 BFGS-Armijo 法和 Newton-Armijo 法进行比较, 本文采用文献[6] 的部分实验数据和实验结果。数据集规模为 $(m + m_1) \times n$, 其中 m, m_1 和 n 为正整数, m 为训练样本数, m_1 为测试样本数, n 为每个样本的属性个数。设 w, x 是 n 维列向量, w 的全部分量皆为 1, $x \in [-100, 100]^n$ 。样本点分 2 种情形: 1) 当 $w^T x \geq 1$ 时, 样本点 x 归为第 1 类; 2) 当 $w^T x \leq -1$ 时, 样本点 x 归为第 2 类。若 $-1 < w^T x < 1$, 则不要该样本点, 重新产生新的样本点。数据集的前 m 个样本用来训练支持向量机的分类器, 后 m_1 个样本用来测试分类器的效果 (性能和效率)。

按训练样本数的规模分别用 BFGS-Armijo 法和 Newton-Armijo 法做 4 个实验, 共 8 个实验。前 6 个实验分为两组: 一组是样本属性个数不变而训练样本数变化时的实验, 数据规模为 $(100, 10)$ 、 $(400, 10)$ 、 $(1000, 10)$; 另一组是训练样本数不变而样本属性个数变化时的实验, 数据规模为 $(200, 10)$ 、

(200,40)、(200,100)。后2个实验为用两种算法求解大规模问题时的测试样本分类正确率比较和CPU时间比较。

实验中的参数 $\varepsilon = 1.0E-3$, 测试样本数 $m_1 = 1000$ 。实验结果见表1~6, 其中第一列是训练样本数的规模, 样本数为 m , 样本属性为 n , 其余各列是取不同算法的测试样本分类正确率或CPU时间。

表1 训练样本数变化时两种算法的测试样本分类正确率 %

数据规模 m, n	BFGS-Armijo 法	Newton-Armijo 法
100, 10	93.20	93.20
400, 10	98.60	97.80
1000, 10	94.20	98.80

表2 训练样本数变化时两种算法的CPU时间 s

数据规模 m, n	BFGS-Armijo 法	Newton-Armijo 法
100, 10	4.07	2.36
400, 10	18.62	15.30
1000, 10	131.43	111.83

表3 样本属性个数变化时两种算法的测试样本分类正确率 %

数据规模 m, n	BFGS-Armijo 法	Newton-Armijo 法
200, 10	96.10	96.90
200, 40	90.60	90.30
200, 100	79.90	78.10

表4 样本属性个数变化时两种算法的CPU时间 s

数据规模 m, n	BFGS-Armijo 法	Newton-Armijo 法
200, 10	7.97	5.44
200, 40	13.62	7.25
200, 100	34.61	9.01

表5 大规模问题两种算法的测试样本分类正确率 %

数据规模 m, n	BFGS-Armijo 法	Newton-Armijo 法
30000, 1000	92.30	97.70
80000, 4000	82.17	77.93

表6 大规模问题两种算法的CPU时间 s

数据规模 m, n	BFGS-Armijo 法	Newton-Armijo 法
30000, 1000	3002.89	1883.42
80000, 4000	90247.50	11708.43

为了判断两种算法的求解效果, 结合表1~6, 分别从以下两方面进行比较:

1) 分类正确率。从表1至表6可以看出, 当数据规模相

同时, Newton-Armijo 法和 BFGS-Armijo 法的测试样本分类正确率基本相同。

2) 分类效率。从表1~6还可以看出, Newton-Armijo 法的CPU时间明显比 BFGS-Armijo 法少, 也就是说 Newton-Armijo 法的效率明显比 BFGS-Armijo 法高; BFGS-Armijo 法的平均CPU时间是35.05 s, Newton-Armijo 法的CPU时间是25.87 s, 即 Newton-Armijo 法的分类效率比 BFGS-Armijo 法平均高出约26.2%。

4 结语

针对引言中提到的光滑支持向量机两个尚未解决的问题, 本文分别列出了用 BFGS-Armijo 法和 Newton-Armijo 法求解 SSVM 模型的具体步骤, 并分别用这两种算法进行了求解, 对实验结果进行了比较研究, 结果表明: BFGS-Armijo 法和 Newton-Armijo 法都是求解光滑模型的有效算法, 但 Newton-Armijo 法更有效。归纳起来可得出以下结论:

1) 这两个算法的分类性能基本相同, 都是求解光滑支持向量机模型 SSVM 的有效算法;

2) Newton-Armijo 法的分类效率明显优于 BFGS-Armijo 法, 其分类效率比 BFGS-Armijo 法平均高出约26.2%。

参考文献:

- [1] 邓乃扬, 田英杰. 数据挖掘的新方法——支持向量机[M]. 北京: 科学出版社, 2004.
- [2] PLATT J C. Fast training of support vector machines using sequential minimal optimization [M]. Cambridge, MA: MIT Press, 1999: 185-208.
- [3] JONCHIMS T. Making large-scale support vector machine learning practical [M]. Cambridge, MA: MIT Press, 1999: 169-184.
- [4] MANGASARIAN O L, MUSICANT D R. Successive overrelaxation for support vector machines [J]. IEEE Transactions on Neural Networks, 1999, 10(8): 1032-1037.
- [5] LEE Y J, MANGASARIAN O L. SSVM: A smooth support vector machine for classification [J]. Computational Optimization and Applications, 2001, 22(1): 5-21.
- [6] 袁玉波, 严杰, 徐成贤. 多项式光滑的支撑向量机[J]. 计算机学报, 2005, 28(1): 9-17.
- [7] 熊金志, 胡金莲, 袁华强, 等. 一类光滑支持向量机新函数的研究[J]. 电子学报, 2007, 35(2): 366-370.
- [8] 熊金志, 袁华强, 彭宏. 多项式光滑的支持向量机一般模型研究[J]. 计算机研究与发展, 2008, 45(8): 1346-1373.

(上接第1597页)

展方法应用于图概念学习、图聚类以提高其效率是本文今后的研究目标。

参考文献:

- [1] INOKUCHI A, WASHIO T, MOTODA H. An Apriori-based algorithm for mining frequent substructures from graph data [C]// PKDD'2000, LNCS 1910. London: Springer-Verlag, 2000: 13-23.
- [2] KURAMOCHI M, KARYPIS G. An efficient algorithm for discovering frequent subgraphs [J]. IEEE Transactions on Knowledge and Data Engineering, 2004, 16(9): 1038-1051.
- [3] YAN X, HAN J W. gSpan: Graph-based substructure pattern mining [C]// ICDM 2002: Proceedings of IEEE International Conference on Data Mining. Washington, DC: IEEE Press, 2002: 721-

724.

- [4] COOK D J, HOLDER L B. Graph-based data mining [J]. IEEE Intelligent Systems, 2000, 15(2): 32-41.
- [5] BANDYOPADHYAY S, MAULIK U, COOK D J, et al. Enhancing structure discovery for data mining in graphical databases using evolutionary programming [C]// Proceedings of the 15th International Florida Artificial Intelligence Research Society Conference. [S.l.]: AAAI Press, 2002: 232-236.
- [6] 常新功, 寇纪淞, 李敏强. 基于带状态回溯个体进化的子结构发现[J]. 计算机应用, 2007, 27(8): 1944-1947.
- [7] 常新功, 寇纪淞, 李敏强. 进化子结构发现在区域经济研究中的应用[J]. 计算机应用, 2008, 28(5): 1173-1176.
- [8] GRUNWALD P. A tutorial introduction to the minimum description length principle [EB/OL]. [2008-12-16]. <http://homepages.cwi.nl/pdg/ftp/mdlintro.pdf>.