一种大规模支持向量机的高效求解算法

冯 昌 李子达 廖士中
(天津大学计算机科学与技术学院 天津 300072)

摘 要 现有大规模支持向量机求解算法需要大量的内存资源和训练时间,通常在大集群并行环境下才能实现。提出了一种大规模支持向量机(SVM)的高效求解算法,以在个人PC机求解大规模SVM。它包括3个步骤:首先对大规模样本进行子采样来降低数据规模;然后应用随机傅里叶映射显式地构造随机特征空间,使得可在该随机特征空间中应用线性SVM来一致逼近高斯核SVM;最后给出线性SVM在多核环境下的并行实现方法以进一步提高求解效率。标准数据集的对比实验验证了该求解算法的可行性与高效性。

关键词 大规模支持向量机,子采样,随机傅里叶特征,并行线性支持向量机
中图法分类号 TP181 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.9.037

Efficient Algorithm for Large-scale Support Vector Machine

FENG Chang LI Zi-da LIAO Shi-zhong
(School of Computer Science and Technology, Tianjin University, Tianjin 300072, China)

Abstract The algorithm for solving large-scale support vector machine (SVM) needs large memory requirement and computation time. Therefore, large-scale SVMs are performed on computer clusters or supercomputers. An efficient algorithm for large-scale SVM was presented, which can be operated on a daily-life PC. First, the large-scale training examples were subsampled to reduce the data size. Then, the random Fourier mapping was explicitly applied to the subsample to generate the random feature space, making it possible to apply a linear SVM to uniformly approximate to the Gaussian kernel SVM. Finally, a parallelized linear SVM algorithm was implemented to speed up the training further. Experimental results on benchmark datasets demonstrate the feasibility and efficiency of the proposed algorithm.

Keywords Large-scale support vector machine, Subsampling, Random Fourier features, Parallelized linear SVM

1 引言

支持向量机[1]是在统计学习理论的基础上发展起来的一类重要的学习方法,是目前最为流行的数据挖掘方法。SVM中核的应用,将原始数据映射到再生核希尔伯特空间(RKHS)中,使得在原始空间中线性不可分的问题转化成RKHS中线性可分的问题。求解SVM的本质是求解一个二次凸优化问题。其求解时间复杂度为$O(l^2)$。由于要存储核矩阵,因此其空间复杂度为$O(l^3)$。当样本数据规模l较大时,SVM的训练时间太长,同时核矩阵的规模太大将导致内存空间不足。因此,原始SVM很难应用到大规模数据问题上。

将SVM扩展到大规模问题是研究SVM学习方法一直致力的目标[2],如序贯最小优化(SMO)分解方法[3]、核心向量机[4]的核心向量采样方法和SVM并行方法[5]等。以上方法虽然能在一定程度上缓解SVM对内存的极大需求,但是随着噪声样本的增加,支持向量的个数也会线性增加。不仅在训练阶段计算开销会急剧增加,而且在预测阶段也是如此。

线性SVM有较高的求解效率,已提出了随机梯度下降法[6]、对偶坐标下降法[7]等方法。线性SVM在超高维文本、基因数据上能得到较好的结果,然而对一般问题很难保证学习精度。线性SVM不需要维护大量的支持向量就可以显式地计算出模型权重,计算开销较少,能很好地扩展到大规模问题。这促进了应用线性SVM来求解核SVM的研究工作。随机傅里叶映射[8]提出一种逼近平移不变核的方法,应用相对低维显式特征映射来一致逼近高斯核对应的无限维隐式特征映射,从而可应用线性SVM来一致逼近高斯核SVM。

本文综合海量样本子采样、随机傅里叶映射、并行线性SVM方法,提出一种可以在个人PC上高效求解大规模SVM的算法,其包括3个步骤。首先,确定训练样本的采样规模,从海量样本中采样得到训练样本。然后,应用随机傅里叶特征来一致逼近核函数,将训练样本映射到随机特征空间,得到训练集。最后,应用基于交替乘子法(ADMM)的并行线性SVM算法[9],在随机特征空间中将训练集分块得到多个子训练集,并行训练线性SVM。标准数据集的对比实验验证了该算法的可行性与高效性。

到稿日期:2014-07-20 返修日期:2014-09-20 本文受国家自然科学基金(61170019)资助。
冯 昌(1989—),男,博士生,主要研究方向为机器学习理论与方法,E-mail:changfeng@tju.edu.cn;李子达(1990—),男,硕士生,主要研究方向为机器学习并行算法;廖士中(1964—),男,博士,教授,博士生导师,主要研究方向为人工智能应用基础、理论计算机科学,E-mail:szliao@tju.edu.cn(通信作者)。

本文第 2 节介绍大规模 SVM 高效算法所用到的随机傅里叶特征；第 3 节在优化理论的基础上，基于交替乘子法(ADMM)给出大规模 SVM 高效算法的具体描述；第 4 节给出该算法的对比实验，验证其可行性与高效性；最后是结束语。

2 随机傅里叶特征

随机傅里叶映射[8]可根据核函数显式地确定特征映射，将只能核化处理的非线性问题转化为映射后特征空间上的线性问题。下面介绍主要概念和结果。

定义 1(平移不变核) 如果一个核函数满足 $k(x,y)=\kappa(x-y)$，则称其为平移不变核。

定义 2(傅里叶变换) $f(t)$是 t 的函数，如果 $f(t)$满足 Dirichlet 条件：具有有限个间断点；有限个极值点；$f(t)$绝对可积，则

$$F(w)=\int_{-\infty}^{+\infty} f(t)\mathrm{e}^{-\mathrm{i}wt}\,\mathrm{d}t$$

称为 $f(t)$的傅里叶变换。且有

$$f(t)=\frac{1}{2\pi}\int_{-\infty}^{+\infty} F(w)\mathrm{e}^{\mathrm{i}wt}\,\mathrm{d}w$$

称为 $F(w)$的傅里叶逆变换。

定理 1(Bochner 定理[8]) 连续函数 $f:\mathbb{R}^d\to\mathbb{C}$ 正定，当且仅当 f 为某一有限非负 Borel 测度 μ 的傅里叶变换，

$$f(x)=\int_{\mathbb{R}^d}\mathrm{e}^{-\mathrm{i}x't}\,\mathrm{d}\mu(t)$$

只要核函数 $\kappa(\delta)$选择合适，Bochner 定理保证存在一个概率分布 $p(w)$的傅里叶变换与 $\kappa(\delta)$对应，则有如下推论。

推论 1 任一正定平移不变核 $k(x,y)=\kappa(x-y)$均满足下式

$$k(x,y)=\int_{\mathbb{R}^d} p(w)\mathrm{e}^{-\mathrm{i}w'(x-y)}\,\mathrm{d}w$$

其中，$p(w)$为 w 的概率密度函数。

令 $Z_w(x)=\mathrm{e}^{-\mathrm{i}w'x}$，则 $k(x,y)=\mathbb{E}[Z_w(x)Z_w(y)^*]$，故 $Z_w(x)Z_w(y)^*$是 $k(x,y)$的无偏估计。为了得到 k 的实值随机特征，注意到概率分布 $p(w)$和核函数 $\kappa(\delta)$均为实值，因此，$\mathrm{e}^{-\mathrm{i}w'(x-y)}$可由 $\cos(w'(x-y))$代替，得到实随机特征 $Z_w(x)=[\cos(w'x),\sin(w'x)]'$。应用蒙特卡罗采样，可构造如下 $2D$ 维随机特征映射：

$$\Phi:\vdash\sqrt{\frac{1}{D}}[\cos(w_1{}'x),\cdots,\cos(w_D{}'x),\cdots,\sin(w_1{}'x),\cdots,\sin(w_D{}'x)]' \tag{1}$$

定理 2 给出随机特征映射 Φ 逼近核函数的一致收敛界。

定理 2(随机傅里叶特征一致收敛界[8]) 设 $\mathcal{M}$ 是 $\mathbb{R}^d$ 上的紧子集，其直径为 $\mathrm{diam}(\mathcal{M})$。对于式(1)定义的随机特征映射，任意 $\varepsilon\in(0,1)$，有

$$\Pr[\sup_{x,y\in\mathcal{M}}|\langle\Phi(x),\Phi(y)\rangle-k(x,y)|\geqslant\varepsilon]\leqslant 2^8\left(\frac{\sigma_p\mathrm{diam}(\mathcal{M})}{\varepsilon}\right)^2\exp\left(-\frac{D\varepsilon^2}{4(d+2)}\right)$$

其中，$\sigma_p^2=\mathbb{E}_p[w'w]$为 $p(w)$的二阶矩。

定理 2 表明，$\langle\Phi(x),\Phi(y)\rangle$依随机特征采样维数 D 渐近地依概率收敛到 $k(x,y)$。

3 算法

交替乘子法(Alternating Direction Method of Multipliers，ADMM)[9]是一种结合了对偶上升法和乘子法的算法，是求解分布式凸优化问题的有效算法。首先简要介绍 ADMM 算法。

对于下列有约束优化问题，

$$\min_{w,z} f(w)+g(z)$$

$$\text{s. t. } Aw+Bz=c$$

使用乘子法得到增广的拉格朗日函数为

$$L(w,z,y)=f(w)+g(z)+u'(Aw+Bz-c)+(\rho/2)\|Aw+Bz-c\|_2^2$$

其中，u 是对偶变量，$\rho>0$ 为惩罚系数。使用上升法来求解上述对偶问题，

$$w^{k+1}=\arg\min_w L(w,z^k,u^k)$$

$$z^{k+1}=\arg\min_z L(w^{k+1},z,u^k)$$

$$u^{k+1}=u^k+\rho(Aw^{k+1}+Bz^{k+1}-c)$$

其中，k 表示迭代步数。ADMM 交替地更新 w 和 z，更新 w 时不涉及函数 g 的计算，更新 z 时不涉及函数 f 的计算。

接下来分析利用 ADMM 算法来设计并行线性 SVM 算法。

对于无约束最优化问题，

$$\min_w f(w)+g(w) \tag{2}$$

设函数 $f(w)$关于 w 可分，$g(w)$关于 w 不可分，则 $f(w)=\sum_{j=1}^m f_j(w)$。使用 ADMM 求解优化问题，问题可以等价地转化为

$$\min_{w,z}\sum_{j=1}^m f_j(w_j)+g(z)$$
$$\text{s. t. } w_j-z=0,j=1,\cdots,m \tag{3}$$

问题(3)对应的增广拉格朗日函数为

$$L_\rho(w_1,\cdots,w_m,u_1,\cdots,u_m,z)=\sum_{j=1}^m f_j(w_j)+\sum_{j=1}^m u_j^k(w_j-z)+\sum_{j=1}^m\frac{\rho}{2}\|w_j-z^k\|_2^2+g(z)$$

迭代求解：

$$w_j^{k+1}=\arg\min_{w_j}(f_j(w_j)+u_j^k(w_j^k-z^k)+\frac{\rho}{2}\|w_j-z^k\|_2^2) \tag{4}$$

$$z^{k+1}=\arg\min_z(g(z)+\sum_{i=1}^m u_j^k(-z^k+\frac{\rho}{2}\|w_j^{k+1}-z^k\|_2^2)) \tag{5}$$

$$u_j^{k+1}=u_j^k+\rho(w_i^{k+1}-z^{k+1}) \tag{6}$$

$$j=1,\cdots,m$$

w 和 z 交替更新，更新 w 时，不涉及原优化问题中不可分的函数 g。又由于函数 f 可分，因此 w 的更新可以并行进行。

线性 L2-SVM 优化问题为

$$\min_w C\sum_{i=1}^l\max(1-y_iw'x_i,0)^2+\frac{1}{2}\|w\|_2^2$$

将其按照问题(3)进行分解，可得

$$\min C\sum_{j=1}^{m}\sum_{i\in B_j}\max(1-y_i w_j' x_i,0)^2+\frac{1}{2}\|z\|_2^2 \quad (7)$$

$$\text{s.t. } w_j-z=0, j=1,\cdots,m$$

其中，将训练集分成 m 块，B_j 表示第 j 块数据下标的集合，问题(7)可以根据式(4)－式(6)分布式求解。

ADMM 在满足如下两条假设条件下收敛[9]。

假设 1　函数 $f,g:\mathbb{R}^n\to\mathbb{R}\cup\{+\infty\}$ 都是正常的闭的凸函数。

假设 2　增广的拉格朗日函数 L_ρ 存在鞍点。

当满足上述假设时，ADMM 迭代有如下收敛性保证：

1)当 $k\to\infty$，残差变量 $r^k=Aw^k+Bz^k-c\to 0$。

2)目标值收敛。设 p^* 表示最优目标函数值，当 $k\to\infty$，$f(w^k)+g(z^k)\to p^*$。

L2-SVM 中 $\sum\max(1-y_i w' x_i,0)^2$ 和 $(1/2)\|w\|_2^2$ 均为正常的闭的凸函数，故满足假设 1。L2-SVM 为二次规划问题且其目标函数为凸函数，约束条件为仿射函数，根据 Slater 条件，L2-SVM 满足强对偶性，即存在点 (w^*,z^*,λ^*)，使得 $L_\rho(w^*,z^*,\lambda)\leqslant L_\rho(w^*,z^*,\lambda^*)\leqslant L_\rho(w,z,\lambda^*)$，即 (w^*,z^*,λ^*) 满足鞍点条件，故 L2-SVM 存在鞍点，满足假设 2。综上，使用 ADMM 求解 L2-SVM 是收敛的。

并行线性 SVM 算法流程如图 1 所示。具体地，第 k 轮迭代时，在子训练集 j 上处理器 j 利用 z^k,u_j^k 训练，得到 w_j^{k+1}，然后将 w_j^{k+1},u_j^k 发送给中心处理器。中心处理器搜集完所有 w_j^{k+1},u_j^k 后计算 z^{k+1}，并广播给处理器 j，处理器 j 利用 u_j^k，w_j^{k+1},z^{k+1} 计算 u_j^{k+1}，完成一次迭代。然后进行下一轮处理，直至收敛。

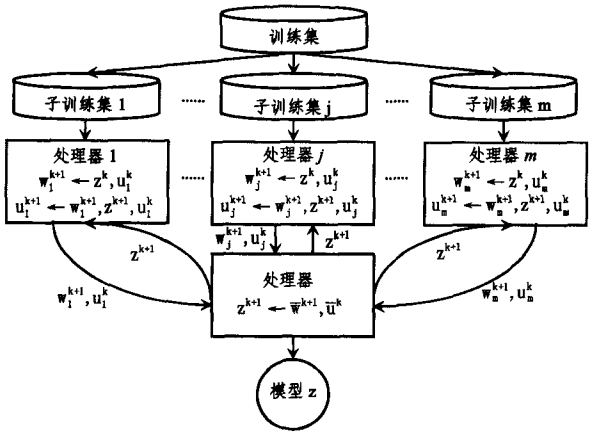

图 1　并行线性 SVM 流程

算法 1　大规模 SVM 的高效求解算法

输入：海量数据 $S=\{(x_i,y_i)\}_{i=1}^{l}$，子采样规模 M，随机特征采样规模 D，高斯核参数 σ，惩罚系数 C。

输出：随机特征 w，模型权重 z。

1. 子采样：$S'\subseteq S$，$|S'|=M$；

2. 随机特征采样：$w_i\sim\mathcal{N}(0,\mathbf{I}/\sigma^2)$，$i=1,\cdots,D$；

3. 随机特征映射：对于所有 $x\in S'$，根据式(1)计算 $\Phi(x)$；

4. 训练数据分块：$T=\{(\Phi(x_i),y_i)\}_1^M=\bigcup_{j=1}^{m}T_j$；

5. 根据式(4)－式(6)并行训练线性 SVM。

4　实验

实验所用的 11 个标准数据集的规格说明如表 1 所列。实验程序用 C＋＋实现，专门在个人 PC 上(4-core 3.1GHz CPU，4GB RAM)完成对比实验。采用高斯核函数 $k(x,y)=\exp(-\gamma\|x-y\|_2^2)$。提出的算法简记为 ESVM，表示以下几个技术的集合：子采样、随机特征映射、并行式线性 SVM。设计随机特征实验和子采样实验来对比 ESVM、已有核 SVM 与大规模 SVM。

4.1　随机特征

比较 ESVM(随机特征映射＋并行线性 SVM)与 LibSVM 的测试准确率和训练时间。LibSVM 参数选择范围 $C\in\{2^{-8},2^{-7},\cdots,2^8\}$，$\gamma\in\{2^{-10},2^{-9},\cdots,2^5\}$，通过 5 折交叉验证进行选择。ESVM 的参数 C,γ 与 LibSVM 的设置一致。实验结果如表 2 所列。

表 1　标准数据集说明

数据集	原始样本数	测试样本数	数据维数
Splice	1000	2175	60
SVMGuide1	3089	4000	4
Mushrooms	7322	802	112
Adult	32561	16281	123
W8A	49749	14951	300
IJCNN1	49990	91701	22
Webspam	280000	70000	254
CovType	464810	116202	54
RCV1	677399	20242	47236
KDD99	4898431	311029	127
Mnist8m-b	8000000	100000	784

表 2　随机特征实验结果比较

数据集	参数(log)		LibSVM		ESVM		
	γ	C	T(s)	Acc(%)	D	T(s)	Acc(%)
Splice	−8	0	0.078	86.89	400	0.094	82.04
SVMGuide1	−10	4	0.141	96.92	200	0.062	95.45
Mushrooms	−6	2	0.562	100.0	150	0.047	100.0
Adult	−6	4	97.62	85.12	50	0.874	85.12
W8A	−6	4	38.52	98.98	300	1.810	98.34
IJCNN1	−1	6	46.55	98.59	300	2.979	97.73

注：γ 为核函数参数，C 为惩罚系数，D 为随机特征采样维数。

在测试准确率上，对于 Splice 数据集，ESVM 比 LibSVM 低了近 5%；对于其他数据集，两算法相当(相差 1%左右)。在训练时间上，对于 Splice 数据集，ESVM 比 LibSVM 多花 0.016s；对于其他数据集，ESVM 所用的时间均比 LibSVM 少，甚至少很多。Splice 数据集相对较小，然而对于 ESVM，随机特征的采样规模 $D=400$ 远大于原数据维数 $d=60$，因而，其训练时间有能比 LibSVM 略多。

该实验表明应用相对低维的显式随机傅里叶特征映射来一致逼近高斯核对应的无限维隐式特征映射是可行的，从而能高效地在随机特征空间中利用线性 SVM 来一致逼近原始高斯核 SVM。

4.2　子采样

选用的 5 个大规模数据集分别是 Webspam、CovType、RCV1、Mnist8m-b、KDD99。ESVM(子采样＋随机特征映射＋并行线性 SVM)的参数包括被逼近的高斯核参数 γ、惩罚系数 C、随机特征采样规模 D 和子采样规模 M。参与对比的算法有 LibSVM、PSVM、Linear SVM、LLSVM，这些算法的实验结果来自对应的参考文献，均为最优参数设置下的测试准确率。

• LibSVM：基于分解方法的 SVM 求解器。其实验环境为 2.5GHz Xeon L5420 处理器，16GB 内存[10]。

• PSVM：基于不完全 Cholesky 分解的并行内点法 SVM 求解器。实验环境为 500 台独立机器，每台机器 CPU 主频高于 2GHz，4GB 内存[5]。

• Linear：线性 SVM，代码采用 LibLinear。实验环境为 2.66GHz Q9400 处理器，4GB 内存[7]。

• LLSVM：基于 Nyström 低秩近似方法的线性 SVM 求解器。实验环境与 Linear 的一致[11]。

部分实验结果如表 3 所列。在 KDD99 数据集上，ESVM 子采样规模 $M=40000$，随机特征采样规模 $D=1000$，得到的测试准确率为 92%（CVM[3] 测试准确率为 93.7%）。

表 3　大规模数据训练时间和测试准确率比较

算法 \ 数据集	Webspam		CovType		RCV1		Mnist8m-b	
	训练时间	测试准确率	训练时间	测试准确率	训练时间	测试准确率	训练时间	测试准确率
LibSVM	4.4h	99.26	13.0h	96.11	—	—	—	—
PSVM	—	—	1655s	97.66	2671s	92.64	—	—
LLSVM	2880s	97.90	4680s	85.10	—	—	22h	97.41
Linear	47s	93.04	16s	76.25	—	—	12h	75.82
ESVM	10.7s	95.90	**53.34s**	**69.89**	20.87s	89.12	1.81s	89.62
	(M=20000,D=1000)		(M=40000,D=1000)		(M=30000,D=1000)		(M=40000,D=1000)	

注："—"表示参考文献没有给出结果。

除 CovType 数据集外，在其他 4 个数据集上，ESVM 的测试准确率与已有工作的测试准确率基本相当。虽然实验环境有很大的差别，但是子采样与随机特征的应用显著地提高了 ESVM 的效率。

在 Webspam 和 Mnist8m-b 数据集上，ESVM 的测试准确率介于 LLSVM 与 Linear 之间。ESVM 逼近的是高斯核 SVM，因此比 Linear SVM 效果好。但是 ESVM 随机特征采样规模 $D=1000$ 小于 LLSVM 在 Webspam 选用的特征映射维数 $k=4000$，在 Mnist8m-b 上 $k=3000$。同时，ESVM 子采样规模仅为原始数据的 1%，但仍得出了很好的结果。

在 CovType 数据集上 ESVM 没有给出相当的结果。考虑到 CovType 训练集中几乎一半数据为支持向量（大约 230k 个），这表明该问题本质上是一个超高维线性可分问题。LibSVM 和 PSVM 最终得到的模型所在的空间维数为 230k 维以上，远远高于 ESVM 中随机特征采样规模（$D=1000$），因此在随机特征空间中线性不可分，采用线性算法效果必然不佳。此外，ESVM 子采样规模 $M=40000$，远小于支持向量个数，没有给出与其它算法相当的结果也是合理的；但在实验设定的环境下，相比其它方法，给出的结果也是适用的。

结束语　提出了一种大规模 SVM 的高效求解算法。应用大规模数据子采样降低训练样本的规模，使用随机傅里叶特征来一致近似核函数，进而在随机傅里叶特征空间上训练线性 SVM，因而并行线性 SVM 的实现进一步提高了训练效率。实验结果表明该算法是可行的、高效的。

进一步的工作将研究采样规模对算法的影响及随机特征采样维度 D 的选取方法，给出理论界、计算开销和可保证的预测精度。

参 考 文 献

[1] Vapnik V. The Nature of Statistical Learning Theory[M]. New York: Springer Verlag, 2000

[2] 文益民，王耀南，吕宝粮，等. 支持向量机处理大规模问题算法综述[J]. 计算机科学，2009，36(7)：20-25
Wen Yi-min, Wang Yao-nan, Lv Bao-liang, et al. Survey of Applying Support Vector Machines to Handle Large-scale Problems[J]. Computer Science, 2009, 36(7): 20-25

[3] Platt J C. Fast training of support vector machines using sequential minimal optimization[M]// Schölkopf B, Burges C, Smola A. Advances in Kernel Methods: Support Vector Learning. Cambridge: MIT Press, 1999: 185-208

[4] Tsang L W, Kwok J, Cheung P M. Core vector machines: Fast SVM training on very large data sets[J]. Journal of Machine Learning Research, 2005, 6(2): 363-392

[5] Chang E Y, Zhu Kai-hua, Wang Hao, et al. Parallelizing support vector machines on distributed computers[M]// Platt J C, Koller D, Singer Y, et al., eds. Advances in Neural Information Processing Systems 20. Cambridge: MIT Press, 2008: 257-264

[6] Zhang Tong. Solving large scale linear prediction problems using stochastic gradient descent algorithms[C]// Proceedings of the 21st International Conference on Machine Learning, 2004. New York, USA, 2004: 919-926

[7] Fan R E, Chang K W, Hsieh C J. LIBLINEAR: A library for large linear classification[J]. Journal of Machine Learning Research, 2008, 9(12): 1871-1874

[8] Rahimi A, Recht B. Random features for large-scale kernel machines[M]// Platt J C, Koller D, Singer Y, et al., eds. Advances in Neural Information Processing Systems 20. Cambridge: MIT Press, 2008: 1177-1184

[9] Boyd S, Parikh N, Chu E, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers[J]. Foundations and Trends® in Machine Learning, 2011, 3(1): 1-122

[10] Chang Y W, Hsieh C J, Chang K W, et al. Training and testing low-degree polynomial data mappings via linear SVM[J]. Journal of Machine Learning Research, 2010, 11(4): 1471-1490

[11] Zhang K, Lan L, Wang Z, et al. Scaling up kernel SVM on limited resources: A low-rank linearization approach[C]// Proceedings of the 15th International Conference on Artificial Intelligence and Statistics, 2012. Canary, Spain, 2012: 1425-1434