计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.201100106

基于网络结构的正则化逻辑回归

胡艳梅[1]　杨　波[2]　多　滨[1]
1 成都理工大学计算机与网络安全学院　成都 610059
2 电子科技大学计算机科学与工程学院　成都 611731

摘　要　逻辑回归是一个应用广泛的分类模型,但由于高维数据分类任务在实际应用中变得越来越频繁,使得分类模型面临着巨大的挑战。应对该挑战的一种有效方法是对模型进行正则化。许多已有的正则化逻辑回归直接运用 $L1$ 范数罚作为正则化罚项,而不考虑特征之间的复杂关联关系。也有一些研究工作基于特征的组信息设计了正则化罚项,但它们假设组信息是预先给定的。文中从网络的视角对特征数据中存在的潜在模式进行挖掘,并基于此提出了一个基于网络结构的正则化逻辑回归。首先,以网络的形式描述特征数据并构建出特征网络;其次,从网络科学的角度对特征网络进行观察和分析,并基于此设计罚函数;然后,以该罚函数为正则化罚项,提出网络结构 Lasso 逻辑回归;最后,结合 Nesterov 加速近端梯度下降法和 Moreau-Yosida 正则化方法,推导了模型的求解过程。在真实数据集上的实验结果显示,所提网络结构 Lasso 逻辑回归表现优异,这表明从网络的视角观察和分析特征数据是研究正则化模型的一个具有潜力的方向。

关键词:正则化罚项;逻辑回归;网络结构;特征选择;近端梯度下降法
中图法分类号　TP181

Logistic Regression with Regularization Based on Network Structure

HU Yan-mei[1], YANG Bo[2] and DUO Bin[1]
1 College of Computer Science and Cyber Security, Chengdu University of Technology, Chengdu 610059, China
2 School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

Abstract　Logistic regression is widely used as classification model. However, as the task of high-dimensional data classification becomes more and more frequent in practical application, the classification model is facing great challenge. Regularization is an effective approach to this challenge. Many existing regularized logistic regression models directly use L1-norm penalty as regularized penalty term without considering the complex relationships among features. There are also some regularization penalty terms designed on the basis of group information of features, but assuming that the group information is prior knowledge. This paper explores the pattern hidden in feature data from the perspective of network and then proposes a regularized logistic regression model based on the network structure. Firstly, this paper constructs feature network by describing feature data in the form of network. Secondly, it observes and analyzes the feature network from the perspective of network science and designs a penalty function based on the observation. Thirdly, it proposes a logistic regression model with network structured Lasso by taking the penalty function as regularized penalty term. Lastly, it infers the solution of the model by combining the Nesterov's accelerated proximal gradient method and the Moreau-Yosida regularization method. Experiments on real datasets show that the proposed regularized logistic regression performs excellently, which demonstrates that observing and analyzing feature data from the perspective of network is a potential way to study regularized model.

Keywords　Regularized penalty term, Logistic regression, Network structure, Feature selection, Proximal gradient method

1　引言

随着信息技术的发展,实际应用中对分类的需求越来越多,如人脸识别、信用卡违约预测、疾病辅助诊断等。逻辑回归(logistic regression)是一个经典的分类模型,它不需要事先假设数据分布,可处理非线性的分类任务。另外,逻辑回归函数是任意阶可导的凸函数,具有较好的求解性质。这些优良特性使得逻辑回归被广泛应用于文本分类[1]、计算机视觉[2]、

到稿日期:2020-11-13　返修日期:2021-03-24　本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
基金项目:国家自然科学基金(61802034,61977013);国家重点研发项目基金(2019YFC1509602);四川省科技计划重点研发项目基金(2021YFG0333)
This work was supported by the Natural Science Foundation of China(61802034,61977013), National Key Research and Development Program of China(2019YFC1509602) and Sichuan Science and Technology Program(2021YFG0333).
通信作者:胡艳梅(huyanmei@cdut.edu.cn)

自然语言处理[3]和生物医学[4]等领域，但高维数据给逻辑回归的实际应用带来了巨大的挑战。一方面，高维度的特征数据导致模型的时间复杂度过高；另一方面，高维特征中常常含有噪声或无用特征，直接使用所有特征会导致分类结果不能满足应用需求。另外，当样本数量受限时，高维特征还会导致模型过拟合。通过对逻辑回归进行正则化来选择重要特征是应对上述挑战的一个有效措施[5-6]。

正则化模型将最小化损失函数和正则化罚项相结合来同时实现模型学习和特征选择，其基本形式如下：

$$\min_{\boldsymbol{w}\in R^d} f(\boldsymbol{w})=l(\boldsymbol{w})+\phi(\boldsymbol{w}) \tag{1}$$

其中，$\boldsymbol{w}$ 为特征的系数向量，$l(\boldsymbol{w})$ 为损失函数，$\phi(\boldsymbol{w})$ 为正则化罚项。L_1 范数罚和 L_2 范数罚是最常见的正则化罚项，L_1 范数罚通过对系数的绝对值之和进行惩罚来压缩系数[7]，而 L_2 范数罚通过对系数向量的模进行惩罚来压缩系数。但是，在某些情况下，L_2 范数罚会将系数压缩到一个很小的数而非 0，L_1 范数罚则针对强相关的特征只会保留其中的一个，因此容易导致次优选择结果。为了解决这个问题，有研究将 L_1 范数罚和 L_2 范数罚结合起来作为正则化罚项[8-13]。文献[14-21]则通过在 L_1 范数罚的基础上对系数之差或和的绝对值进行惩罚，使得强相关特征对应的系数(几乎)相等。文献[22-23]通过对两两系数的无穷范数罚来实现强相关特征对应的系数(几乎)相等。组 Lasso 罚项(group lasso penalty)[24]通过特征组信息对系数的 $L_{2,1}$ 范数进行惩罚以实现任意给定特征组的稀疏选择。随着研究的深入，组 Lasso 罚项被扩展以考虑重复组(overlapping groups)[25]和树组(tree-structured groups)[26]等特殊组结构。稀疏组 Lasso 罚项(sparse group lasso penalty)则是将特征系数上的 L_1 范数罚和组 Lasso 罚函数结合在一起，以同时实现特征组间和组内的稀疏选择[27-28]。关于这些罚函数的更详细的介绍可参考文献[29]。已有研究表明，特征的组结构信息对正则化模型很有用，但现有的模型都假设特征的组结构信息是已知的或简单地将同一功能的特征划分为一组[26,30]，而未深入地探究特征数据的结构信息。

目前已有许多关于逻辑回归正则化模型的研究工作，但大部分都以 L_1 范数罚和 L_2 范数罚为正则化罚项。文献[31]中，组 Lasso 罚函数被作为多分类逻辑回归的正则化罚项，但组信息是针对系数而言，即将同一特征对应的系数看作一个组。而在许多应用场景下，特征之间的关联关系是复杂的且隐藏着特殊的模式，单一的线性相关或单纯的功能组划分不能有效地对其进行描述。这使得已有的正则化逻辑回归不能直接且有效地利用特征数据中隐藏的有用信息。

因此，面对现有逻辑回归模型的正则化罚项未考虑特征数据的潜在结构模式这一问题，受“网络是描述真实世界中各种复杂系统的一种简单且直观的方式”这种认识的启发，本文从网络的视角对特征之间潜在的结构模式进行挖掘，并基于挖掘出的模式设计正则化罚项，提出了一种基于网络结构的正则化逻辑回归，以同时实现特征选择和模型学习。首先，本文以网络的形式对特征及特征之间的关系进行描述，并构建出特征网络。接着，对特征网络的结构进行观察和分析，并在此基础上设计正则化罚项，将其命名为网络结构 Lasso 罚项(network structured lasso penalty)。然后，将网络结构 Lasso 作为逻辑回归的正则化罚项，得到基于网络结构的正则化逻辑回归。最后，对模型的求解过程进行推导并得出相应的求解算法。基于网络结构的正则化逻辑回归通过挖掘特征数据中潜在的结构模式来确定罚项，可以有效地利用特征数据本身的特性。本文的主要贡献如下：

(1)从网络的视角提出了一种基于网络结构的正则化逻辑回归模型。采用网络的形式将特征及特征之间的关系描述为特征网络。基于对特征网络的观察和分析，设计了网络结构 Lasso 罚项作为逻辑回归的正则化罚项。该正则化罚项是通过挖掘特征网络中存在的潜在模式而设计的，因此可以考虑特征数据本身的特性。

(2)推导了模型的求解过程并提供了用于模型求解的 NsLasso 算法。以 Nesterov 加速近端梯度下降法[32-33]为模型的基本求解方法，进一步地，推导了网络结构 Lasso 罚项的求解可转换为树组 Lasso 罚项的求解。因此，在 Nesterov 加速近端梯度下降法的基础上结合求解树组 Lasso 罚项的 Moreau-Yosida[26]，推导出了用于模型求解的 NsLasso 算法。

(3)在真实数据集上对模型进行了实验验证，并对结果进行了详细分析。在 4 个真实数据集上与传统的正则化逻辑回归模型进行了实验对比，并基于实验结果分析和总结了基于网络结构的正则化逻辑回归的适用场景。实验结果表明，相比传统的正则化逻辑回归，基于网络结构的正则化逻辑回归可以实现同等(甚至更好)的分类效果，且需要的特征数量往往更少。这表明了挖掘特征数据中隐藏的潜在模式可以更好地帮助模型完成分类任务。

本文第 2 节首先简要描述了以 L_1 范数罚为正则化罚项的逻辑回归，详细描述了网络结构 Lasso 罚函数的设计过程，并将其用于逻辑回归的正则化罚项，阐述了网络结构 Lasso 正则化逻辑回归的求解过程；第 3 节对实验结果进行了描述和分析；最后总结全文并展望未来。

2 网络结构 Lasso 正则化逻辑回归

逻辑回归是广泛应用于分类或预测的经典机器学习模型。为了提高模型的泛化效果，L_1 范数罚作为正则化罚项被引入逻辑回归中[6,34-35]。但是，随着特征维度的增大，特征之间的关系可能越来越复杂，如果能将特征之间的关联模式应用到模型中，势必会提高模型的效果。用网络的形式对特征空间进行描述，赋予了我们一个观察特征及特征之间复杂关系的新视角。基于此，通过结合特征网络结构和 L_1 范数罚，本文设计了网络结构 Lasso 作为逻辑回归的正则化罚项，进而提出了一种基于网络结构的正则化逻辑回归。本节首先介绍了 L_1 正则化逻辑回归；其次描述了特征网络及其构建，并对特征网络进行了观察和分析；然后对网络结构 Lasso 正则化逻辑回归进行了描述；最后给出了模型的求解算法 NsLasso。

2.1 L_1 正则化逻辑回归

特别地，针对二分类任务，给定一个样本 $\boldsymbol{x}=\{x^1,x^2,\cdots,x^d\}$($d$ 为特征个数)，$p(y=1\mid\boldsymbol{x})=1/(1+e^{-\boldsymbol{xw}+b})$，其中 $\boldsymbol{w}=(w_1,w_2,\cdots,w_d)^{\mathrm{T}}$ 为系数向量(w_i 表示第 i 个特征对应的系

数),b 为偏置。对于数据集 $\boldsymbol{T}=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_n,y_n)\}$,其似然函数为 $\prod_{i=1}^{n}p(y_i\mid\boldsymbol{x}_i;\boldsymbol{w},b)$。对该似然函数取对数并求反即为对数似然损失函数:

$$l(\boldsymbol{w},b)=-\ln\left(\prod_{i=1}^{n}p(y_i\mid\boldsymbol{x}_i;\boldsymbol{w},b)\right) \tag{2}$$

对式(2)进行极小值求解即可求得系数向量 $\boldsymbol{w}$ 和偏置 b:

$$\min_{\boldsymbol{w},b} l(\boldsymbol{w},b) \tag{3}$$

为了剔除冗余特征和噪声特征,同时达到更好的泛化效果,L_1 范数罚作为正则化罚项被引入模型中[6,34-35]。基于式(2)和 L_1 范数罚的逻辑回归有如下形式:

$$\min_{\boldsymbol{w},b} f_1(\boldsymbol{w},b)=l(\boldsymbol{w},b)+\lambda_1\|\boldsymbol{W}\|_1 \tag{4}$$

2.2 特征网络及其构建

现实世界中的各种系统均可通过网络进行直观地描述,特征空间也是如此。将每个特征看作一个节点,特征之间的关联关系看作相应节点之间的边,特征空间就转化为了特征网络[1)]。图 1 给出了一个特征网络示例。

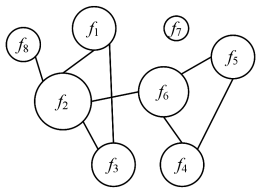


图 1 特征网络示例(节点大小与其度成正比)

Fig. 1 Example of feature network (the size of each node is propotional to its degree)

给定数据集 $\boldsymbol{T}=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_n,y_n)\}$,其中第 i 个样本 $\boldsymbol{x}_i=\{x_i^1,x_i^2,\cdots,x_i^d\}$ 是由 d 个特征上的取值组成的 d 维特征向量,$y_i\in\{0,1\}$ 为样本 $\boldsymbol{x}_i$ 的标签,所有的 $\boldsymbol{x}_i$ 组成一个 $n\times d$ 的矩阵 $\boldsymbol{X}$。构建数据集 $\boldsymbol{T}$ 的特征网络的过程如算法 1 所示。首先创建一个空网络 $G_{\boldsymbol{X}}$,即 $G_{\boldsymbol{X}}$ 中无节点和边,见算法 1 中的步骤 1。接着,在 $G_{\boldsymbol{X}}$ 中为每个特征(即 $\boldsymbol{X}$ 中的每一列)创建一个节点,见算法 1 中的步骤 2。最后,为 $G_{\boldsymbol{X}}$ 添加边,计算矩阵 $\boldsymbol{X}$ 中每两列 $\boldsymbol{X}_{*,j}$ 和 $\boldsymbol{X}_{*,k}$($1\leqslant j,k\leqslant d$ 且 $j\neq k$)之间的皮尔逊相关系数,即:

$$pc(j,k)=\frac{\sum_{i=1}^{n}(X_{i,j}-\overline{X_{*,j}})(X_{i,k}-\overline{X_{*,k}})}{\sqrt{\sum_{i=1}^{n}(X_{i,j}-\overline{X_{*,j}})^2}\sqrt{\sum_{i=1}^{n}(X_{i,k}-\overline{X_{*,k}})^2}} \tag{5}$$

其中,$1\leqslant i\leqslant n$,$X_{i,j}$ 表示矩阵 $\boldsymbol{X}$ 的第 i 行和第 j 列的元素,$\overline{X_{*,j}}$ 为 $\boldsymbol{X}_{*,j}$ 中所有元素的平均值(见算法 1 中的步骤 4);如果第 j 列和第 k 列之间的皮尔逊相关系数(即 $pc(j,k)$)大于给定的阈值 δ,则在 $G_{\boldsymbol{X}}$ 中为节点 j 和 k 添加一条边且其权值为 $pc(j,k)$(见算法 1 中的步骤 5—步骤 8)。如此,便可构建出数据集 $\boldsymbol{T}$ 所对应的特征网络。

算法 1 构建特征网络 ConstNet(Construct Network,Const-Net)

输入:$\mathbf{X}$,δ

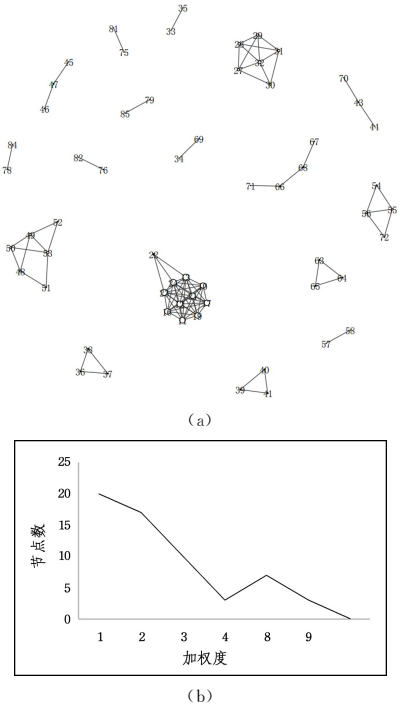
输出:$G_{\mathbf{X}}$

1. 令 $G_{\mathbf{X}}$ 为空网络
2. 为 $\mathbf{X}$ 的每一列创建一个节点 v 加入 $G_{\mathbf{X}}$ 中
3. FOR $\forall v_j,v_k(j\neq k)$:
4. (1)计算 $\mathbf{X}_{*,j}$ 和 $\mathbf{X}_{*,k}$ 的皮尔逊相关系数 pc(j,k)(见式(5))
5. (2)IF pc(j,k)$>\delta$:
6. 1)将边 $e(v_j,v_k)$ 添加到 $G_{\mathbf{X}}$ 中
7. 2)令 $w(v_j,v_k)=pc(j,k)$
8. END IF
9. END FOR

2.3 特征网络的结构分析

运用特征网络对特征空间进行描述,以便能够从网络的视角对特征及特征之间的复杂关联关系进行观察和分析。下文以 Credit 数据集(数据集的介绍见实验部分)为例进行说明。

图 2 给出了数据集 Credit 的特征网络(见图 2(a))和加权度分布(见图 2(b))。从图中可以看出:1)特征网络中存在社区结构(图 2(a)中连通的空心节点可看作是一个社区),即高度相关的特征呈聚簇现象,形成不同的特征组;2)节点的加权度呈现出一定的差异性(图 2(a)中节点的加权度与其大小成正比,图 2(b)中展示的节点加权度分布更直接地说明了它们的差异性),这直观地体现出了特征的冗余度存在差异,进而体现出针对分类模型的学习不同特征的重要度并不一样。以图 1 为例,$f2$ 的加权度大于 $f1$ 和 $f3$,这是因为 $f2$ 同时与 $f1$,$f3$,$f6$ 和 $f8$ 高度相关,而 $f1$ 和 $f3$ 仅与彼此和 $f2$ 高度相关,进而也说明了 $f2$ 的冗余度大于 $f1$ 和 $f3$。那么在学习分类或预测模型时,$f2$ 的重要性低于 $f1$ 和 $f3$,因为前者的信息基本已被 $f1$,$f3$,$f6$ 和 $f8$ 覆盖。而 $f7$ 不与任何特征相关,因此其冗余度最低,那么它对于分类或预测任务的贡献可能是最大的。



(a)

(b)

图 2 基于数据集 Credit 构建的特征网络和节点加权度分布

Fig. 2 Feature network constructued on dataset of Credit and corresponding distribution of weighted degree

1) 在此前的一项工作中我们也采用了网络的形式来描述特征空间。该工作的主要贡献是提出了一种过滤式特征选择方法,详细内容请参看文献[36]。

2.4 网络结构 Lasso 正则化逻辑回归

通过观察特征网络有如下发现：1)特征网络可能存在社区结构，即高度相关的特征聚集成不同的特征组；2)不同的特征具有不同的重要度。针对特征呈聚簇现象这一情形又可能存在如下情况：不是所有的特征组对分类都贡献有效信息，在同一特征组中也不是所有特征都对分类有用。另一方面，将所有特征都用于训练分类模型可能导致过拟合。尤其是在数据维度过高时，上述两个问题会更加明显。因此，本小节基于特征网络的结构设计一个罚函数来作为逻辑回归的正则化罚项。

具体地，首先运用社区发现算法[37]找出特征网络中的社区，并将每个社区看作一个特征组，得到特征组集合 $\boldsymbol{C}=\{C_1, C_2, \cdots, C_K\}$（$K$ 表示特征组的个数）。然后，结合特征组和特征节点加权度这两种信息得到网络结构 Lasso 罚函数，以同时在特征组之间和特征组内进行选择，并在选择时考虑每个特征的重要度。网络结构 Lasso 罚函数的形式化表示如下：

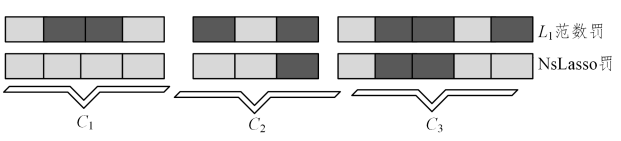
$$\phi_{\lambda_2}^{\lambda_1}(w)=\lambda_1\|\boldsymbol{D}\boldsymbol{w}\|_1+\lambda_2\sum_{g=1}^{K}\|\boldsymbol{w}_{C_g}\|_1 \tag{6}$$

其中，$\boldsymbol{D}$ 是对角矩阵，对角线上的元素值分别为相应特征节点的加权度；$\|\boldsymbol{D}\boldsymbol{w}\|_1$ 可以看作系数向量 $\boldsymbol{w}$ 受特征节点加权度限制的 L_1 范数罚，控制特征层面的选择；$\sum_{g=1}^{K}\|\boldsymbol{w}_{C_g}\|_1$ 是特征组上的 $L_{2,1}$ 范数罚，控制特征组层面的选择；$C_g\in\boldsymbol{C}$ 表示由社区结构得到的第 g 个特征组，且 $\|\boldsymbol{w}_{C_g}\|_1=\sqrt{\sum_{j\in C_g}w_j^2}$，而 $\lambda_1,\lambda_2\geqslant 0$ 控制着两个层面的选择程度。针对呈聚簇现象的特征，$\phi_{\lambda_2}^{\lambda_1}(\boldsymbol{w})$ 可以从重要的特征组中选取重要的特征参与分类任务；而针对无聚簇现象的特征，$\phi_{\lambda_2}^{\lambda_1}(\boldsymbol{w})$ 也可以对重要特征进行甄别，以剔除冗余和无用特征，进而提高模型的泛化能力。值得注意的是，当特征网络无明显的社区结构时，网络结构 Lasso 退化为一种特殊的 Lasso。确切地说，当 $\boldsymbol{C}$ 为空时网络结构 Lasso 的第二项为 0，此时就是系数惩罚程度与对应特征节点加权度成正比的 Lasso；而当 $\boldsymbol{C}$ 相对整个网络来说很小时，网络结构 Lasso 的第二项也会很小，也就意味着特征组层面的惩罚对系数的影响很小）。当特征网络呈现出明显的社区结构时（$\boldsymbol{C}$ 不为空且有很大一部分节点都在 $\boldsymbol{C}$ 中），网络结构 Lasso 就是稀疏组 Lasso 的一个实例化。

进一步地，以网络结构 Lasso 为正则化罚项的逻辑回归为：

$$\min_{w,b} f_2(\boldsymbol{w},b)=l(\boldsymbol{w},b)+\phi_{\lambda_2}^{\lambda_1}(\boldsymbol{w}) \tag{7}$$

式(7)所示的正则化逻辑回归在 L_1 范数罚的基础上考虑了特征网络的结构，实现了特征组之间及特征组内的特征选择。而只考虑 L_1 范数罚的特征选择（见式(4)）只考虑特征层面的选择，未考虑特征的重要度。二者的区别如图 3 所示。



注：特征划分为 3 组，深色表示被选中的特征，浅色表示未被选中的特征

图 3 L_1 范数罚和网络结构 Lasso 罚的主要区别

Fig. 3 Main difference between the L_1-norm penality and the network structured Lasso penality

虽然网络结构 Lasso 可看作稀疏组 Lasso 的一个实例化，但本文与已有稀疏组 Lasso 工作的最大区别在于：首先将特征空间转换为特征网络，然后从网络科学的角度挖掘特征的潜在模式，并以此设计罚函数。所幸的是，由此得到的罚函数正好可以归到稀疏组 Lasso 这一理论框架下。

2.5 NsLasso 算法

式(7)所示的正则化逻辑回归是一个以对数似然为损失函数、网络结构 Lasso 罚函数为正则化罚项的优化问题。其中，对数似然损失函数是任意阶可导的凸函数，而网络结构 Lasso 罚项是凸函数却不可微分。Nesterov 加速近端梯度下降法是解决这类优化问题的一种有效方法[32-33]。另外，通过推导可知，网络结构 Lasso 罚项的求解可转换为树组 Lasso 罚项的求解。而 Moreau-Yosida[26]是求解树组 Lasso 罚项的一种有效方法。因此，本文在 Nesterov 加速近端梯度下降法的基础上结合 Moreau-Yosida 正则化方法推导出求解式(7)的 NsLasso 算法。在介绍 NsLasso 算法之前，先结合式(7)对 Nesterov 加速近端梯度下降法和 Moreau-Yosida 正则化方法进行描述。

2.5.1 Nesterov 加速近端梯度下降法

令 $\boldsymbol{y}=[y_1, y_2, \cdots, y_n]^{\mathrm{T}}\in R^n$，$\boldsymbol{1}\in R^n$ 是元素均为 1 的向量，$\boldsymbol{b}\in R^n$ 是元素均为 b 的向量，∇ 表示导数，则有：

$$\nabla_w l(\boldsymbol{w},b)=\boldsymbol{X}^{\mathrm{T}}\left(\frac{1}{1+\exp(-\boldsymbol{X}\boldsymbol{w}^{\mathrm{T}}-\boldsymbol{b})}-\boldsymbol{y}\right) \tag{8}$$

$$\nabla_b l(\boldsymbol{w},b)=\frac{1}{1+\exp(-\boldsymbol{X}\boldsymbol{w}^{\mathrm{T}}-\boldsymbol{b})}-\boldsymbol{y} \tag{9}$$

令 $l'(\boldsymbol{w},b)=[\nabla_w l(\boldsymbol{w},b), \nabla_b l(\boldsymbol{w},b)]$，$\hat{\boldsymbol{w}}=[\boldsymbol{w},b]$，且在下文中用 $\hat{\boldsymbol{w}}^b$ 指代 b 部分。易知 $l'(\hat{\boldsymbol{w}})$ 满足 L-Lipschitz 条件，因此在给定的 $\hat{\boldsymbol{s}}$ 附近可将 $l(\hat{\boldsymbol{w}})$ 通过二阶泰勒展式近似为：

$$f_{L,\hat{\boldsymbol{s}}}(\hat{\boldsymbol{w}})=l(\hat{\boldsymbol{s}})+\langle l'(\hat{\boldsymbol{s}}), \hat{\boldsymbol{w}}-\hat{\boldsymbol{s}}\rangle+\phi_{\lambda_2}^{\lambda_1}(\boldsymbol{w})+\frac{L}{2}\|\hat{\boldsymbol{w}}-\hat{\boldsymbol{s}}\|^2 \tag{10}$$

其中，$L>0$，$\langle\cdot,\cdot\rangle$ 表示内积。因此，若用梯度下降法对式(7)进行求解，则每一步梯度下降迭代实际上等价于最小化 $f_{L,\hat{\boldsymbol{s}}}(\hat{\boldsymbol{w}})$。而式(10)可转换为：

$$f_{L,\hat{\boldsymbol{s}}}(\hat{\boldsymbol{w}})=\frac{L}{2}\left\|\hat{\boldsymbol{w}}-\left(\hat{\boldsymbol{s}}-\frac{1}{L}l'(\hat{\boldsymbol{s}})\right)\right\|_2^2+\phi_{\lambda_2}^{\lambda_1}(\boldsymbol{w})-\frac{1}{2L}\|l'(\hat{\boldsymbol{s}})\|^2+l(\hat{\boldsymbol{s}}) \tag{11}$$

式(11)中的最后两项与 $\hat{\boldsymbol{w}}$ 无关。因此，在求解式(7)的梯度下降迭代中，每一步迭代实际上应为：

$$\min_{\hat{\boldsymbol{w}}}\frac{L}{2}\left\|\hat{\boldsymbol{w}}-\left(\hat{\boldsymbol{s}}-\frac{1}{L}l'(\hat{\boldsymbol{s}})\right)\right\|_2^2+\phi_{\lambda_2}^{\lambda_1}(\boldsymbol{w}) \tag{12}$$

针对式(7)，Nesterov 加速近似梯度下降方法通过交替更新近似解序列 $\{\hat{\boldsymbol{w}}_k\}$ 和搜索点序列 $\{\hat{\boldsymbol{s}}_k\}$ 进行求解。搜索点 $\hat{\boldsymbol{s}}_k$ 由 $\hat{\boldsymbol{w}}_k$ 和 $\hat{\boldsymbol{w}}_{k-1}$ 的放射组合确定，即 $\hat{\boldsymbol{s}}_k=\hat{\boldsymbol{w}}_k+\beta_k(\hat{\boldsymbol{w}}_k-\hat{\boldsymbol{w}}_{k-1})$（$\beta_k$ 是确定组合比例的参数）。近似解 $\hat{\boldsymbol{w}}_{k+1}$ 则通过在 $\hat{\boldsymbol{s}}_k-\frac{1}{L_k}l'(\hat{\boldsymbol{s}}_k)$ 处最小化网络结构 Lasso 罚函数得到：$\hat{\boldsymbol{w}}_{k+1}=\min_{\hat{\boldsymbol{w}}}\frac{L}{2}\left\|\hat{\boldsymbol{w}}-\left(\hat{\boldsymbol{s}}_k-\frac{1}{L_k}l'(\hat{\boldsymbol{s}}_k)\right)\right\|_2^2+\phi_{\lambda_2}^{\lambda_1}(\boldsymbol{w})$（$\frac{1}{L_k}$ 可看作是梯度下降步长）。Nesterov 加速近端梯度下降法的过程如算法 2 所示。

其中，步骤 3、步骤 10 和步骤 13 用于确定参数β_k和L_k[6]。当迭代次数达到设定值时，则终止算法。实际应用中，也可根据相邻两次迭代的目标函数值变化来终止算法，如当相邻两次迭代的目标函数值变化小于10^{-5}时，终止算法。

算法 2 Nesterov 加速近端梯度下降法(Nesterov's Accelerated Proximal Gradient Method，NAPGD)

输入：$L_0>0$，$\hat{\boldsymbol{w}}_0$，N，$\boldsymbol{C}$，$\boldsymbol{D}$

输出：$\hat{\boldsymbol{w}}_{N+1}$

1. 令$\hat{\boldsymbol{w}}_1=\hat{\boldsymbol{w}}_0$，$t_{-1}=0$，$t_0=1$

2. FOR k=1 TO N：

3. 令$\beta_k=\frac{t_{k-2}-1}{t_{k-1}}$，$L_k=L_{k-1}$

4. 令$\hat{\boldsymbol{s}}_k=\hat{\boldsymbol{w}}_k+\beta_k(\hat{\boldsymbol{w}}_k-\hat{\boldsymbol{w}}_{k-1})$

5. WHILE(TRUE)：

6. $\hat{\boldsymbol{w}}_{k+1}=\min_{\hat{\boldsymbol{w}}}\frac{L_k}{2}\left\|\hat{\boldsymbol{w}}-\left(\hat{\boldsymbol{s}}_k-\frac{1}{L_k}l'(\hat{\boldsymbol{s}}_k)\right)\right\|_2^2+\phi_{\lambda_2}^{\lambda_1}(\boldsymbol{w})$

7. IF $l(\hat{\boldsymbol{w}}_{k+1})\leqslant l(\hat{\boldsymbol{s}}_k)+<l'(\hat{\boldsymbol{s}}_k),\hat{\boldsymbol{w}}_{k+1}-\hat{\boldsymbol{s}}_k>+\frac{L_k}{2}\|\hat{\boldsymbol{w}}_{k+1}-\hat{\boldsymbol{s}}_k\|^2$ THEN：

8. 跳转到步骤 13

9. ELSE：

10. $L_k=2L_k$

11. END IF

12. END WHILE

13. 令$t_k=(1+\sqrt{1+4t_{k-1}^2})/2$

14. END FOR

2.5.2 Moreau-Yosida 正则化

上述加速近端梯度下降法的关键是在每次迭代中求解$\hat{\boldsymbol{w}}_{k+1}$。接下来，本文将通过推导说明网络结构 Lasso 可以转化为树组 Lasso(更准确地说，是去掉根节点的树组 Lasso)，进而可以利用树组 Lasso 的求解方法，即 Moreau-Yosida 正则化方法，完成$\hat{\boldsymbol{w}}_{k+1}$的求解。

为了叙述方便，我们给出如下定义：

$$\pi_{\lambda_2}^{\lambda_1}(\boldsymbol{s})=\min_{\boldsymbol{w}}\frac{1}{2}\|\boldsymbol{w}-\boldsymbol{s}\|^2+\phi_{\lambda_2}^{\lambda_1}(\boldsymbol{w}) \tag{13}$$

其中，$\pi_{\lambda_2}^{\lambda_1}(\boldsymbol{s})$称作近邻算子(proximal operator)[38]，本质上就是$\boldsymbol{w}_{k+1}$(偏置 b 不会影响$\boldsymbol{w}_{k+1}$的解。因此，当$\boldsymbol{s}=\boldsymbol{s}_k-\frac{1}{L}l'(\boldsymbol{s}_k)$时，式(13)等价于式(12))。根据文献[38]可得推论 1。

推论 1 令$\boldsymbol{w}^*=\pi_{\lambda_2}^{\lambda_1}(\boldsymbol{s})$，$SGN(\boldsymbol{s})$和$sgn(\boldsymbol{s})$是作用于$\boldsymbol{s}$的分量函数：1)当$s^i=0$时，$SGN(s^i)=[-1,1]$且$sgn(s^i)=0$；2)当$s^i>0$时，$SGN(s^i)=\{1\}$且$sgn(s^i)=1$；3)当$s^i<0$时，$SGN(s^i)=\{-1\}$且$sgn(s^i)=-1$。有如下性质成立：1)如果$s^i>0$，则$0\leqslant w^{*i}\leqslant s^i$；2)如果$s^i<0$，则$s^i\leqslant w^{*i}\leqslant 0$；3)如果$s^i=0$，则$w^{*i}=0$；4)$SNG(\boldsymbol{s})\subseteq SNG(\boldsymbol{w}^*)$；5)$\pi_{\lambda_2}^{\lambda_1}(\boldsymbol{s})=sng(\boldsymbol{s})\odot\pi_{\lambda_2}^{\lambda_1}(|\boldsymbol{s}|)$。

证明：易知在$\lambda_1,\lambda_2>0$的情况下，$\frac{1}{2}\|\boldsymbol{w}-\boldsymbol{s}\|^2+\phi_{\lambda_2}^{\lambda_1}(\boldsymbol{w})$是严格凸函数，因此$\boldsymbol{w}^*$是唯一的最优解。对于第一个性质：如果$w^{*i}>s^i$，那么可以通过令$w'^j=w^{*j}(j\neq i)$且$w'^i=s^i$而构造出一个$\boldsymbol{w}'$；类似地，如果$w^{*i}<0$，那么可以通过令$w'^j=w^{*j}(j\neq i)$且$w'^i=0$而构造出一个$\boldsymbol{w}'$；易知，在上述两种构造方式中，$\boldsymbol{w}'$对应的目标值均比$\boldsymbol{w}^*$对应的目标值更小。因此，第一条性质成立。同理，第二条和第三条性质也成立。通过前 3 条性质和 SGN 的定义易知第四条性质成立。对于第五条性质：当$s^i>0$和$s^i=0$时，$|s^i|=s^i$，显然满足性质；当$s^i<0$时，$s^i\leqslant w^{*i}\leqslant 0$，则对于$|s^i|=-s^i>0$，其解为$-w^{*i}$，满足$0\leqslant -w^{*i}\leqslant -s^i$。因此，第五条性质成立。证毕。

基于推论 1 可以得到推论 2。

推论 2 令$\boldsymbol{v}=[\lambda_1,\lambda_1,\cdots,\lambda_1]\in R^d$，$\boldsymbol{\lambda}_1^d=\boldsymbol{D}\boldsymbol{v}^{\mathrm{T}}$，$\boldsymbol{u}=\mathrm{sgn}(\boldsymbol{s})\odot\max(|\boldsymbol{s}|-\boldsymbol{\lambda}_1^d,0)$，且

$$\pi_{\lambda_2}^0(\boldsymbol{u})=\min_{\boldsymbol{w}}\frac{1}{2}\|\boldsymbol{w}-\boldsymbol{u}\|^2+\lambda_2\sum_{g=1}^{K}\|\boldsymbol{w}_{Cg}\|_1 \tag{14}$$

则有$\pi_{\lambda_2}^{\lambda_1}(\boldsymbol{v})=\pi_{\lambda_2}^0(\boldsymbol{u})$成立。

证明：令$g(\boldsymbol{w})=\frac{1}{2}\|\boldsymbol{w}-\boldsymbol{s}\|^2+\phi_{\lambda_2}^{\lambda_1}(\boldsymbol{w})$，$h(\boldsymbol{w})=\frac{1}{2}\|\boldsymbol{w}-\boldsymbol{u}\|^2+\lambda_2\sum_{g=1}^{K}\|\boldsymbol{w}_{Cg}\|_1$，$\boldsymbol{w}^*$为$h(\boldsymbol{w})$的最优解，则$\boldsymbol{w}^*$作为最优解的充分必要条件为：

$$0\in\partial h(\boldsymbol{w}^*)=\boldsymbol{w}^*-\boldsymbol{u}+\partial\phi_{\lambda_2}^0(\boldsymbol{w}^*) \tag{15}$$

其中，$\partial h(\boldsymbol{w}^*)$和$\partial\phi_{\lambda_2}^0(\boldsymbol{w}^*)$分别表示$h(\cdot)$和$\phi_{\lambda_2}^0(\cdot)$在$\boldsymbol{w}^*$处的次微分。接下来，我们只需要证明$0\in\partial g(\boldsymbol{w}^*)$即可完成推论 2 的证明。$g(\cdot)$在$\boldsymbol{w}^*$处的次微分为：

$$\partial g(\boldsymbol{w}^*)=\boldsymbol{w}^*-\boldsymbol{u}+\lambda_1^d\odot SGN(\boldsymbol{w}^*)+\partial\phi_{\lambda_2}^0(\boldsymbol{w}^*) \tag{16}$$

由$\boldsymbol{u}$的定义可知，$\boldsymbol{u}\in\boldsymbol{s}-\boldsymbol{\lambda}_1^d\odot SGN(\boldsymbol{u})$。根据推论 1 中的第四条性质可知，$SGN(\boldsymbol{u})\subseteq SGN(\boldsymbol{w}^*)$。因此，有$0\in\partial g(\boldsymbol{w}^*)$。

$$\boldsymbol{u}\in\boldsymbol{s}-\boldsymbol{\lambda}_1^d\odot SGN(\boldsymbol{w}^*) \tag{17}$$

根据式(15)－式(17)可知，$0\in\partial g(w^*)$。证毕。

根据文献[26]，树组 Lasso 罚函数的定义如定义 1 所示。

定义 1 给定一个深度为dp的索引树Tr，令$Tr_i=\{G_1^i,G_2^i,\cdots,G_{n_i}^i\}$包含第$i$层的所有节点，其中$n_0=1$，$G_1^0=\{1,2,\cdots,d\}$，$n_i\geqslant 1(i=1,2,\cdots,dp)$。所有节点满足：1)同层节点之间无重叠，即对于任意的$i\in\{1,\cdots,dp\}$有$G_j^i\cap G_k^i=\emptyset(j\neq k,1\leqslant j,k\leqslant n_i)$；2)若$G_j^i$的父节点为$G_j^{i-1}$，则有$G_j^i\subseteq G_j^{i-1}$。树组 Lasso 罚函数定义为$\phi(\boldsymbol{w})=\sum_{i=0}^{dp}\sum_{j=1}^{n_i}\lambda_j^i\|\boldsymbol{w}_{G_j^i}\|(\lambda_j^i\geqslant 0$为节点$G_j^i$的权值)。

因此，可知$\pi_{\lambda_2}^{\lambda_1}(\boldsymbol{v})$的计算可由$\pi_{\lambda_2}^0(\boldsymbol{u})$通过软阈值得到，即只需要最优化式(14)即可求解出$\pi_{\lambda_2}^{\lambda_1}(\boldsymbol{v})$。又由定义 1 可知，若索引树$Tr'$满足$dp=1$，$G_1^0=\{1,2,\cdots,d\}$，$Tr_1=\{G_1^1,G_2^1,\cdots,G_K^1\}$且$G_g^1(g=1,2,\cdots,K)$为社区$Cg$的节点序集，则式(14)中的罚项$\lambda_2\sum_{g=1}^{K}\|\boldsymbol{w}_{Cg}\|$对应索引树$Tr'$的第一层。进而求解出索引树$Tr'$的第一层对应的参数即可完成$\pi_{\lambda_2}^0(\boldsymbol{u})$的求解。

Moreau-Yosida 正则化是求解树组 Lasso 的一种有效方法，其基本思想是自索引树的叶子节点开始沿着分支往上依次做正则化处理，直到根节点停止，即可完成树组 Lasso 的正则化计算。索引树的第 0 层(即根节点)对应的参数就是树组 Lasso 的解(更详细的细节请参考文献[26])。通过前述分析可知，$\pi_{\lambda_2}^{\lambda_1}(\boldsymbol{v})$可转化为$\pi_{\lambda_2}^0(\boldsymbol{u})$，而后者正好对应着索引树$Tr'$的第一层(同时也是叶子层)。因此，只需要对这一层进行 Moreau-Yosida 的正则化处理即可求得$\pi_{\lambda_2}^0(\boldsymbol{u})$。由此，我们便

得到了优化式(12)的算法(即算法2中第6步的求解),如算法3所示。由于偏置不参与正则化罚项,因此 b 可直接通过梯度下降得到,如算法3中的步骤1所示。对于 $\boldsymbol{w}$ 的求解,先进行梯度下降得到 $\boldsymbol{v}=\boldsymbol{s}-\frac{1}{L}\nabla_{\boldsymbol{w}}l(\boldsymbol{s},\hat{\boldsymbol{s}}^b)$,如算法3中的步骤2所示。然后,根据推论2得到 $\boldsymbol{u}$,将 $\pi^{\lambda_1}_{\lambda_2}(\boldsymbol{v})$ 转换为 $\pi^{0}_{\lambda_2}(\boldsymbol{u})$,如算法3中的步骤3和步骤4所示。最后,运用 Moreau-Yosida 正则化方法的求解方式得到 $\pi^{0}_{\lambda_2}(\boldsymbol{u})$,如算法3中的步骤6和步骤7所示。

算法3 求解式(12)的算法

输入:$\hat{\boldsymbol{s}}$, L, λ_1, λ_2, **C**, **D**

输出:$\hat{\boldsymbol{w}}$

1. 令 $\mathrm{b}=\hat{\mathbf{s}}^{\mathrm{b}}-\frac{1}{\mathrm{L}}\nabla_{\mathrm{b}}\mathrm{l}(\hat{\mathbf{s}},\hat{\mathbf{s}}^{\mathrm{b}})$

2. 令 $\mathbf{v}=\mathbf{s}-\frac{1}{\mathrm{L}}\nabla_{\mathbf{w}}\mathrm{l}(\mathrm{s},\hat{\mathbf{s}}^{\mathrm{b}})$

3. 令 **u** 为与 **v** 具有相同维度的向量,且 **u**=0

4. 对于 **u** 中的每一元素 u^{j},令

$$\mathrm{u}^{\mathrm{j}}=\begin{cases}\mathrm{v}^{\mathrm{j}}-\frac{\lambda_1 D_{\mathrm{jj}}}{\mathrm{L}}, & \text{当}\mathrm{v}^{\mathrm{j}}>\frac{\lambda_1 D_{\mathrm{jj}}}{\mathrm{L}}\\ \mathrm{v}^{\mathrm{j}}+\frac{\lambda_1 D_{\mathrm{jj}}}{\mathrm{L}}, & \text{当}\mathrm{v}^{\mathrm{j}}<-\frac{\lambda_1 D_{\mathrm{jj}}}{\mathrm{L}}\\ 0, & \text{当}-\frac{\lambda_1 D_{\mathrm{jj}}}{\mathrm{L}}\leqslant\mathrm{v}^{\mathrm{j}}\leqslant\frac{\lambda_1 D_{\mathrm{jj}}}{\mathrm{L}}\end{cases}$$

5. 令 **w**=**u**

6. 对于每个特征组 $\mathrm{C_g}$,令 $\mathrm{norm_g}=\|\mathbf{w}_{\mathrm{Cg}}\|_1$

7. 对于每一个特征组 $\mathrm{C_g}$ 中的每个特征 $\mathrm{j}\in\mathrm{C_g}$,令

$$\mathbf{w}^{\mathrm{j}}=\begin{cases}0 & \text{,当}\mathrm{norm_g}\leqslant\lambda_2/\mathrm{L}\\ \frac{\mathrm{norm_g}-\lambda_2/\mathrm{L}}{\mathrm{norm_g}}\cdot\mathrm{u_j} & \text{,当}\mathrm{Lnorm_g}>\lambda_2/\mathrm{L}\end{cases}$$

2.5.3 NsLasso 算法

算法4为用于求解基于网络结构的正则化逻辑回归的NsLasso算法。首先运用算法1在数据集上构建特征网络 $G_{\boldsymbol{X}}$,然后运用社区发现算法发现特征网络中的社区结构 $\boldsymbol{C}$,并将每一社区看作一个特征组,最后运用算法2所示的加速近端梯度下降法求得特征系数(即完成式(7)的求解)。而在近似梯度下降求解的每一迭代步中,需要运用算法3处理结合了特征节点加权度的 L_1 罚项和通过特征组得到的 $L_{2,1}$ 罚项(即求解式(12))。

假设数据集由 n 个样本和 d 个特征组成,构建特征网络的时间复杂度为 $O(nd^2)$。这是因为计算任意一对特征之间的皮尔逊相关系数的时间复杂度为 $O(n)$,而总共有 $d(d-1)/2$ 对特征。特征组划分的时间复杂度取决于采用的社区发现算法,不过目前已有与网络节点数或边数呈线性时间复杂度的算法。本文实验中采用的社区发现算法的时间复杂度就约为 $O(d)$(详细的分析请参考文献[39])。求解参数 $\hat{\boldsymbol{w}}$ 采用的是 Nesterov 加速近端梯度下降法(见算法2)。从初始值 $\hat{\boldsymbol{w}}_0$ 开始,根据梯度迭代更新。在第 k 次迭代中需要多次搜索 $\hat{\boldsymbol{s}}_k$ 周围的点以找到最佳的 $\hat{\boldsymbol{w}}_{k+1}$。每一个搜索点的确定需要通过调用算法3来完成,时间复杂度为 $O(d)$。而每次搜索的范围都以减半的速度靠近 $\hat{\boldsymbol{s}}_k$,故搜索次数通常不会太多(在本文的实验中不超过26次),可看作一个小于某个常量 c 的正整数。因此第 k 次迭代的时间复杂度为 $O(cd)$。由于总共需要迭代 N 次,因此算法2的时间复杂度为 $O(cNd)$。

算法4 NsLasso 算法

输入:**T**, δ, $\mathrm{L_0}>0$, $\hat{\mathbf{w}}_0$, N

输出:$\hat{\mathbf{w}}_{\mathrm{N}+1}$

1. 根据 **T** 得到矩阵 **X**,构建特征网络 $\mathrm{G_T}=\mathrm{ConstNet}(\mathbf{X},\delta)$,并得到对角矩阵 **D**

2. 在 $\mathrm{G_X}$ 上运用社区发现算法得到社区集合 **C**,并将每个社区看作一个特征组

3. $\hat{\mathbf{w}}_{\mathrm{N}+1}=\mathrm{NAPGD}(\hat{\mathbf{w}}_0,\mathrm{L_0},\mathrm{N},\mathbf{C},\mathbf{D})$

3 实验结果与分析

本节将在真实数据集上对网络结构 Lasso 逻辑回归的有效性进行实验验证,并选取与其同类的其他正则化逻辑回归作为对比模型。在展示和分析实验结果之前,先对采用的数据集、评价指标、对比模型以及实验环境进行简要描述。

3.1 实验数据和评价指标

本实验采用4个数据集进行测试,它们分别是 Credit, Arcene, Dexter 和 Gisette。Credit 数据集是由中国某银行提供的信息卡违约数据,每条记录中的特征对应一张信用卡的消费行为,如月消费次数、最高消费金额等,标签0表示未违约,1表示违约。Arcene, Dexter 和 Gisette 来自 NIPS 会议的特征选择竞赛。Arcene 数据集是通过 SELDI 技术获得的质谱数据,每条记录的特征对应人的各项指标值,标签0表示不是癌症,1表示是癌症。Dexter 数据集是文字集合,每条记录对应一篇文章且由词袋的形式表示,每个特征对应一个单词的频度,标签1表示文章是关于企业并购的,0表示不是关于企业并购的。Gisette 数据集则是手写体字4和9的图片集,每条记录对应一张规定大小的图片,每个特征是一个像素值,要求的任务是区分手写体4和9。由于竞赛需要,Arcene, Dexter 和 Gisette 都经过了特殊处理且只以数值的形式供大家研究使用。4个数据集的统计信息如表1所列。

表1 数据集统计信息

Table 1 Statistics of the datasets

数据集	样本数	特征数
Credit	5499	98
Arcene	200	10000
Dexter	600	20000
Gisette	7000	5000

AUC(Area Under Curve)指 ROC 曲线(Receiver Operating Characteristic Curve)下方的面积,是综合评价学习模型优劣的指标,尤其是针对逻辑回归这种需要确定决策阈值的模型。另外,本文的目的是通过挖掘并利用特征空间的潜在模式来改进正则化罚项,以排除无用特征和冗余特征,并优化模型。因此,本文采用 AUC 和排除的特征数量(系数为0的特征)作为模型的评价指标。AUC 的值越接近1表明结果越好,排除的特征数越多表明结果越好。

3.2 对比模型和实验环境

为了保持一致性,本实验采用同属于正则化逻辑回归的 Lasso 逻辑回归、加权 Lasso 逻辑回归和稀疏组 Lasso 逻辑回

归作为对比模型。后两者是通过分别考虑 3.2 节中所述的两点观察（即特征节点的加权度呈现出差异性和高度相关的特征呈聚簇现象）而确定的两个对比模型。

（1）Lasso 逻辑回归：以 L_1 范数罚为正则化罚项的逻辑回归，如式（4）所示。

（2）加权 Lasso 逻辑回归：在 L_1 范数罚的基础上考虑了特征节点的加权度，并以此为逻辑回归的正则化罚项，其形式化表示为：

$$\min_{\boldsymbol{w},b} f_1(\boldsymbol{w},b)=l(\boldsymbol{w},b)+\lambda_1\|\boldsymbol{D}\boldsymbol{w}\|_1$$

（3）稀疏组 Lasso 逻辑回归：以 L_1 范数罚和特征组的 $L_{2,1}$ 范数罚为正则化罚项的逻辑回顾，其形式化表示为：

$$\min_{\boldsymbol{w},b} f_1(\boldsymbol{w},b)=l(\boldsymbol{w},b)+\lambda_1\|\boldsymbol{w}\|_1+\lambda_2\sum_{g=1}^{K}\|\boldsymbol{w}_{Cg}\|_1$$

所有实验均采用五折交叉验证的方式进行，即将数据 5 等分，依次取其中的 1 份作为测试集，剩下的 4 份作为训练集，最后的指标值则是 5 次实验的平均值。构建特征网络的相关性系数阈值 δ 的取值范围为 $\{0.8,0.85,0.9,0.95\}$，并通过交叉验证的方式确定（δ 在 Credit 数据集上的值为 0.85，在其他 3 个数据集上的值则均为 0.95）。λ_1 和 λ_2 的值是按照文献[26,30,38]中的方法进行设置的，即 $\lambda_1=\lambda_2=\gamma\lambda_1^{\max}$，其中 $\gamma=\{5\times10^{-1},2\times10^{-1},1\times10^{-1},5\times10^{-2},2\times10^{-2},1\times10^{-2},5\times10^{-3},2\times10^{-3},1\times10^{-3}\}$，$\lambda_1^{\max}$ 则是使得 $f_2(\boldsymbol{w},b)$ 或 $f_1(\boldsymbol{w},b)$（对应对比模型）的解为 0 的最小值（当 $\lambda_1\geqslant\lambda_1^{\max}$ 时，$f_2(\boldsymbol{w},b)$ 或 $f_1(\boldsymbol{w},b)$ 的解为 0）。现实中采用热启动的方式进行模型求解，即初始的 $\boldsymbol{w}$ 设置为 0，然后从最大的 γ 值（即 0.5）开始求解直到最小的 γ 值（即 0.001），且当前 γ 值对应的 $\boldsymbol{w}$ 初始值为前一 γ 值求解出的 $\boldsymbol{w}$。迭代次数 N 设置为 1 000，但当相邻两次迭代的目标函数值变化小于 10^{-5} 时则停止迭代。由于要求组内特征高度聚集而组间特征尽量少关联，因此需要每个社区内的节点尽可能地连接紧密而社区之间的节点尽量少连接。因此，本文采用著名的 SCAN（Structural Clustering Algorithm for Networks）[39] 来完成社区发现任务，并将参数 μ（最少邻居数量）设置为 3，而相似性阈值则采用原文建议的方式确定（细节请参看文献[39]）。特征网络构建和社区发现的代码通过 C++ 实现，模型求解代码则通过 Python 实现。所有实验均在一台 CPU 为 Intel i7 4.0 GHz、内存为 16GB 的个人电脑上进行。

3.3 实验结果与分析

3.3.1 社区发现结果和加权度分析

在 4 个数据集上的社区发现结果如图 4 所示。对于数据集 Arcene 和 Dexter，由于社区个数较多，因此将每个社区当作一个节点（我们称之为社区节点）进行可视化展示。社区节点的大小与社区成员个数成正比，若原社区之间有边，则对应的两个社区节点之间也有边且边的大小与其权值成正比（社区节点之间的边权值为原社区之间的边权值之和）。从可视化结果可以看出，Arcene 的各社区之间的界限很模糊，而其他数据集的社区结构很分明，社区之间几乎没有交集。Credit 和 Gisette 数据集对应的特征网络各有 5 个社区，但对于 Gisette 数据集来说，发现的 5 个社区都很小，这也意味着在该数据集中只有很小一部分节点呈聚簇现象。

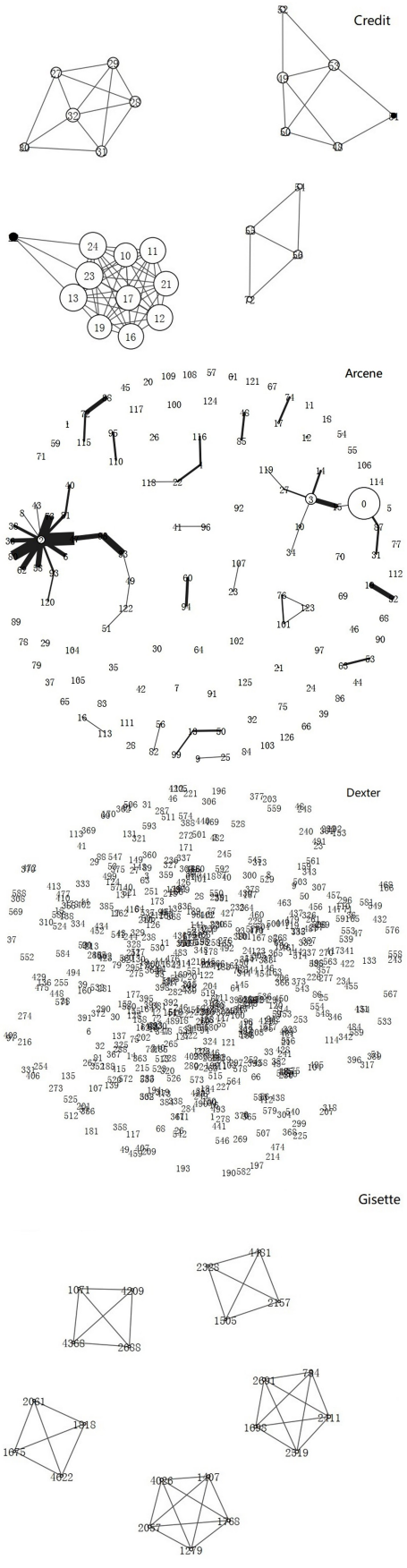
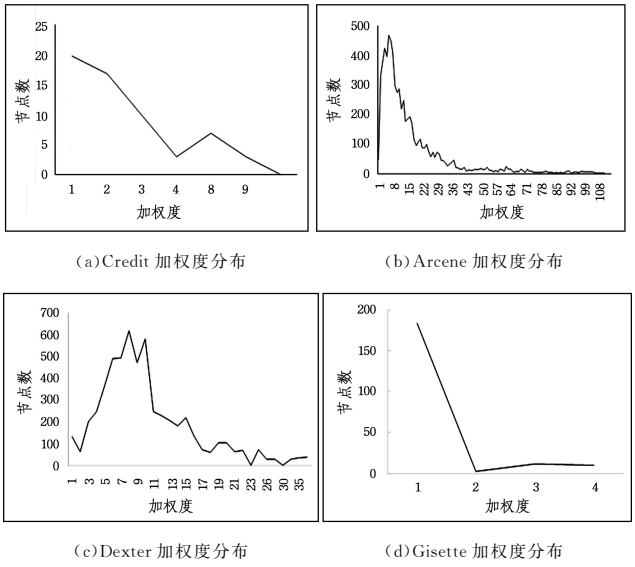


图 4 Credit，Arcene，Dexter 和 Gisette 对应的社区发现结果可视化

Fig. 4 Visualization of the community structure discovered on datasets of Credit，Arcene，Dexter and Gisette

进一步地，本文分析了 4 个数据集对应的加权度分布情况。首先，将节点的加权度进行四舍五入取整，然后统计具有相同度的节点数，最后以横轴为度、纵轴为节点数得到加权度分布图，如图 5 所示。从图中可以看出，在由 Arcene 和 Dexter 两个数据集构建的特征网络中，节点的加权度分布较广，范围分别为[1,113]和[1,37]。虽然加权度在 8 左右的节点占比最大，但拥有更大加权度的节点也是存在的，并且数量也不少(例如，对于 Arcene 数据集，加权度在[50,100]内的节点个数为 401)。由此可知，Arcene 和 Dexter 两个数据集中的节点加权度差异很明显。相比 Arcene 和 Dexter 数据集，Credit 数据集对应的节点加权度差异更小，但也存在一定的差异性。例如，有 3 个节点的加权度在 4 左右，7 个节点的加权度在 8 左右，3 个节点的加权度在 9 左右。而 Gisette 数据集对应的节点加权度差异较小，其加权度分布范围为[1,4]，且 88%的节点加权度为 1 左右。



(a)Credit 加权度分布 (b)Arcene 加权度分布

(c)Dexter 加权度分布 (d)Gisette 加权度分布

图 5 Credit，Arcene，Dexter 和 Gisette 对应的加权度分布

Fig. 5 Distribution of the weighted degree corresponding to datasets of Credit, Arcene, Dexter and Gisette

3.3.2 不同模型的实验结果和分析

接下来，本文对各正则化逻辑回归的实验结果进行分析。表 2 列出了不同模型的最佳结果。其中，Lasso 表示 Lasso 逻辑回归，WLasso 表示加权 Lasso 逻辑回归，SgLasso 表示稀疏组 Lasso 逻辑回归，NsLasso 表示网络结构 Lasso 逻辑回归。从表中可以看出，在数据集 Credit 上，NsLasso 和 WLasso 对应的 *AUC* 值相等，且相比 Lasso 其 *AUC* 值提高了约 1.5%；进一步地，NsLasso 排除了 55 个特征，WLasso 排除了 50 个特征，分别约为 Lasso 排除的特征数(34 个)的 1.6 倍和 1.5 倍。而 SgLasso 的 *AUC* 值与 Lasso 相等，但比 Lasso 多排除了 1 个特征。在 Arcene 数据集上，WLasso 的 *AUC* 值最高(即 0.763)，随后分别是 NsLasso(即 0.754)，Lasso(即 0.739)和 SgLasso(即 0.735)；并且 WLasso 排除的特征数也是最多的(即 3407 个)，高于 NsLasso 的 3230 个、SgLasso 的 2022 个和 Lasso 的 786 个。这是因为 Arcene 对应的社区结构边界模糊且节点加权度差异很大(见图 4 和 5)，所以在考虑了节点加权度的情况下，NsLasso 相比 WLasso 没有任何优势。但社区结构依然可以在训练模型的过程中做贡献，因为 SgLasso 比 Lasso 更优(虽然 SgLasso 的 *AUC* 值略低于 Lasso 的 *AUC* 值，但前者排除的特征数约是后者的 2.6 倍)。在数据集 Dexter 上，NsLasso 和 SgLasso 的 *AUC* 值相等且略高于 Lasso 的，后者又略高于 WLasso。但 NsLasso 和 WLasso 排除的特征(分别为 3850 个和 3780 个)约是 Lasso 排除的特征(为 1317 个)的 2.9 倍。SgLasso 相比 Lasso 也有一定的优势，前者的 *AUC* 值略高于后者且前者排除的特征数(1334)也多于后者。在数据集 Gisette 上，NsLasso 和 WLasso 的 *AUC* 值相等(为 0.929)，且略低于 SgLasso 和 Lasso 的 0.931。不过，前两者排除的特征为 3518 个，比后两者(3326 个)多排除 192 个。这种微小的优势与 Gisette 对应的节点加权度差异不明显相吻合。而 NsLasso(SgLasso)较之 WLasso(Lasso)无优势，这是因为其特征网络中只有 5 个小型的社区。对于 5000 个特征来说，只有极小部分节点组成了特征组。此时，特征组层面的罚项贡献几乎被特征层面的罚项掩盖了。

表 2 不同模型的实验结果

Table 2 Results of different models

指标	数据集	NsLasso	SgLasso	WLasso	Lasso
AUC	Credit	**0.592**	0.583	**0.592**	0.583
	Arcene	0.754	0.735	**0.763**	0.739
	Dexter	**0.966**	0.966	0.962	0.965
	Gisette	0.929	**0.931**	0.929	**0.931**
排除的特征数量	Credit	**55**	35	50	34
	Arcene	3230	2022	**3407**	786
	Dexter	**3850**	1334	3780	1317
	Gisette	3518	3226	3518	3326

总的来说，NsLasso 在 Credit 和 Dexter 上的表现最佳，其次是 Arcene 和 Gisette。对于数据集 Credit 和 Dexter，特征网络呈现出明显的社区结构且节点加权度也呈现出一定的差异性，因此 NsLasso 和 WLasso 均表现出一定的优势。对于数据集 Arcene，由于其社区结构边界模糊(相当于没有明显的特征组划分)，NsLasso 表现不突出；但其节点加权度差异明显，因此 WLasso 较 Lasso 具有明显的优势。对于数据集 Gisette，其特征网络只有 5 个小型社区(相比其特征网络规模来说，只有很少一部分节点组成特征组)，因此 NsLasso 相比 WLasso 基本无优势。另外，由于其节点加权度差异微小，NsLasso 相比 SgLasso 以及 WLasso 相比 Lasso 也基本无优势。

由此可知，当特征网络呈现出明显的社区结构和节点加权度差异时，NsLasso 具有显著的优势；当特征网络无明显的社区结构但节点加权度差异明显时，NsLasso 退化为 WLasso，此时相比 Lasso 其依然具有优势，且节点的加权度差异越大，优势越明显；当特征网络呈现出明显的社区结构但节点加权度差异微小时，NsLasso 退化为 SgLasso，此时相比 Lasso 其依然存在一定的优势，且整个特征网络的社区结构越明显，优势就越大；而当特征网络既无明显的社区结构也无明显的节点加权度差异时，NsLasso 等同于 Lasso。表 3 列出了 NsLasso 的适用场景。

表 3 NsLasso 的适用场景

Table 3 Application scenes of NsLasso

场景	例子	结论(相比 Lasso 的优势)
特征网络的社区结构明显且节点加权度差异明显	Credit Dexter	优势显著
特征网络的社区结构明显但节点加权度差异微小	—	退化为 SgLasso,存在优势
特征网络的社区结构不明显但节点加权度差异明显	Arcene	退化为 WLasso,优势显著
特征网络社区结构不明显且节点加权度差异微小	Gisette	退化为 Lasso,无优势

3.3.3 运行时间

本文对各个模型的运行时间进行了比较,如表 4 所列。其中,FN 列是构建特征网络的时间开销,CD 列是社区发现的时间开销,最后 4 列分别是各模型在每个数据集上完成一个求解轨迹的时间开销。从表中可以看出,随着特征数量和样本数的增多,特征网络构建的时间开销增大,最多为 628 s(对应 Gisette 数据集),而社区发现的时间开销很小,最多为 2s(对应数据集 Arcene),与其他项相比几乎可以忽略不记。而对于模型求解,各模型的时间开销相差很小。具体地,在 Credit 数据集上,时间开销最小的是 SgLasso,随后是 WLasso,Lasso 和 NsLasso,且它们之间的差距最多为 2 s;在 Arcene 数据集上,最快的是 Lasso,随后是 WLasso,NsLasso 和 SgLasso,且它们之间的差距最多为 34.2 s;在数据集 Dexter 上,最快的是 SgLasso,随后是 Lasso,NsLasso 和 WLasso,且它们之间的差距最多为 18 s;在数据集 Gisette 上,最快的是 NsLasso,随后是 WLasso,SgLasso 和 Lasso,且它们之间的最大差距为 213.6 s。总的来说,在模型求解任务上,NsLasso 与其他 3 个模型相比并不慢,有时甚至更快。虽然 NsLasso 需要特征网络构建和社区发现,而特征网络构建的时间开销会随着特征数和样本数的增大而增大,但在训练模型之前花费一定的时间对特征空间进行挖掘和分析是值得的,这有助于训练出更好的模型。很多应用场景中模型训练都是线下完成的,此时为了获得更优秀的分类模型,特征网络和社区发现这两部分的时间开销是可以接受的。

表 4 不同模型的运行时间

Table 4 Running times of different models

(单位:s)

数据集	FN	CD	NsLasso	SgLasso	WLasso	Lasso
Credit	0	0	71	69	70.6	70.8
Arcene	71	2	176.8	185.6	172	151.4
Dexter	268	0	363.6	349.2	367.2	360.6
Gisette	628	0	3060.2	3265.2	3132.8	3273.8

3.3.4 实例分析

为了更直观地比较各种模型,本文以 Credit 数据集的两个特征组为例,随机选取一次实验结果进行说明。特征组 1 包含的节点为{48,49,50,52,53},特征组 2 包含的节点为{54,55,56,72}。这些节点之间的拓扑关系如图 4(第一幅图)所示,它们对应的特征如表 5 中的"名称"列所示。通过社区发现,将"近 1(3,6)个月转出次数"与"近 3(6)个月转出金额"划分为一组,而"近 1(3,6)个月最大转出金额"与"平均转出金额"划分为另一组。这个划分结果与我们的认知是吻合的。但是,"近 1 个月转出次数"(id 为 51)未被划分到任何特征组中。这是因为该特征只与"近 1 个月转出次数"(id 为 48)和"近 3 个月转出金额"(id 为 53)之间有连边,且前者与后两者之间的拓扑相似性相比其他节点之间的拓扑相似性更低,因此未被划分为一组。如果我们降低相似性阈值,那么该特征就会被归入第 1 组中,但这会导致其他连接不是很紧密的特征被归入某些特征组,从而与我们的期望"组内特征高度聚集"相左。总的来说,类似 id 为 51 这样的特征即使未被划分到任何组对结果的影响也不大,因为我们还有特征层面的惩罚对其进行甄选(4 个模型都排除了 51 号特征)。

表 5 特征组、特征及被选择情况的实例

Table 5 Examples of feature groups, features and selections

	id	名称	NsLasso	SgLasso	WLasso	Lasso
特征组 1	48	近 1 个月转出次数	0	0	0	0
	49	近 6 个月转出次数	0	0	0	0
	50	近 3 个月转出次数	0	0	0	0
	52	近 6 个月转出金额	0	0	0	0
	53	近 3 个月转出金额	0	0	0	0
特征组 2	54	近 1 个月最大转出金额	0	1	0	1
	55	近 6 个月最大转出金额	0	1	1	1
	56	近 3 个月最大转出金额	0	1	0	1
	72	平均转出金额	1	1	1	1

表 5 的最后 4 列列出了各模型对两组特征的选择结果,"1"表示选择了该特征,"0"表示排除了该特征。从结果可以看出,各模型都将第一组特征全部排除了。而对于第 2 组特征,NsLasso 排除了 54,55 和 56 号特征,WLasso 排除了 54 和 56 号特征,SgLasso 和 Lasso 则一个也未排除。总的来说,NsLasso 排除的特征最多,WLasso 次之,最后是 SgLasso 和 Lasso。

结束语 本文将特征数据转换为特征网络,然后从网络科学的角度对数据进行观察和分析发现:1)某些特征网络存在社区结构,即高度相关的特征呈聚簇现象;2)节点的加权度呈现出差异性。受此启发,本文设计出了网络结构 Lasso 罚函数,并将其作为正则化罚项提出了网络结构 Lasso 逻辑回归,以同时达到特征选择和分类模型学习的目的。进一步地,推导了模型的求解过程。最后,在真实数据集上对提出的模型进行了实验和分析。实验结果表明,相比传统的正则化逻辑回归,网络结构 Lasso 逻辑回归在需要更少特征的情况下表现持平甚至更优。这说明从网络的视角研究模型的正则化罚项是一个具有潜能的方向。

虽然本文提出的网络结构 Lasso 逻辑回归具有一定的优势,但前提是特征网络中节点的度差异明显且还需要进一步判断特征网络是否呈现出明显的社区结构。因此,相比传统的正则化逻辑回归,网络结构 Lasso 逻辑回归需要额外的工作,总的时间开销更大。另外,不是所有特征数据对应的特征网络都满足模型的两个前提。因此,下一步的工作是收集和分析更多的真实数据集,以分析网络结构 Lasso 的适应领域并得出更具普适性的结论。

参 考 文 献

[1] GEORGIANA I, GÖKHAN B, GERHARD W. Fast Logistic Regression for Text Categorization with Variable-length N-

grams[C]// International Conference on Knowledge Discovery and Data Mining. ACM Digital Library, 2008: 354-362.

[2] FRIEDMAN J, HASTIE T, TIBSHIRANI R. Additive Logistic Regression: a Statistical View of Boosting[J]. The Annals of Statistics, 2000, 28: 337-407.

[3] JURAFSKY D, MARTIN J H. Speech and Language Processing: An Introduction to Natural Language Processing[M]// Computational Linguistics and Speech Recognition. Upper Saddle River: Prentice Hall, 2000.

[4] ZHANG X C, ZHANG Q Z, WANG X F, et al. Structured Sparse Logistic Regression with Application to Lung Cancer Prediction Using Breath Volatile Biomarkers[J]. Statistic in Medicine, 2020, 39(7): 955-967.

[5] LEE S, LEE H, ABBEEL P, et al. Efficient L1 Regularized Logistic Regression[C]// The 21st AAAI Conference on Artificial Intelligence. AAAI, 2006: 401-408.

[6] LIU J, CHEN J, YE J P. Large-scale Sparse Logistic Regression [C]// International on Knowledge Discovery and Data Mining. ACM Digital Library, 2009: 547-556.

[7] NG A Y. Feature Selection, L1 vs L2 Regularization, and Rotational Invariance[C]// International Conference on Machine Learning. ACM Digital Library, 2004: 78-85.

[8] ZOU H, HASTIE T. Regularization and Variable Selection via the Elastic Net[J]. Journal of the Royal Statistical Society: Series B(Statistical Methodology), 2005, 67(2): 301-320.

[9] ZENG L M, XIE J. Group Variable Selection via Scad-l2[J]. A Journal of Theoretical and Applied Statistics, 2014, 48(1): 49-66.

[10] BECKER N, TOEDT G, LICHTER P, et al. Elastic SCAD as a Novel Penalization Method for SVM Classification Task in High-dimensional Data[J]. BMC Bioinformatics, 2011, 12(1): 1-13.

[11] LORBERT A, RAMADGE P J. The Pairwise Elastic Net Support Vector Machine for Automatic fMRI Feature Selection [C]// International Conference on Speech and Signal Processing. IEEE, 2013: 1036-1040.

[12] LORBERT A, EIS D, KOSTINA V, et al. Exploiting Covariate Similarity in Sparse Regression via the Pairwise Elastic Net [C]// International Conference on Artificial Intelligence and Statistics. PMLR, 2010: 477-484.

[13] GRAVE E, OBOZINSKI G R, BACH F R. Trace Lasso: a Trace Norm Regularization for Correlated Designs[C]// Annual Conference on Neural Information Processing Systems. ACM Digital Library, 2011: 2187-2195.

[14] TIBSHIRANI R, SAUNDERS M, ROSSET S, et al. Sparsity and Smoothness via the Fused Lasso[J]. Journal of the Royal Statistical Society: Series B (Statistics Methodology), 2005, 67(1): 91-108.

[15] RINALDO A. Properties and Refinements of the Fused Lasso [J]. The Annals of Statistics, 2009, 37(5B): 2922-2952.

[16] TIBSHIRANI R, WANG P. Spatial Smoothing and Hot Spot Detection for CGH Data Using the Fused Lasso[J]. Biostatistics, 2008, 9(1): 18-29.

[17] RAPAPORT F, BARILLOT E, VERT J P. Classification of ArrayCGH Data Using Fused SVM[J]. Bioinformatics, 2008, 24(13): 375-382.

[18] ZHOU J, LIU J, NARAYAN V A, et al. Modeling Disease Progression via Fused Sparse Group Lasso[C]// International Conference on Knowledge Discovery and Data Mining. ACM Digital Library, 2012: 1095-1103.

[19] YE G B, XIE X. Split Bregman Method for Large Scale Fused Lasso[J]. Computational Statistics & Data Analysis, 2011, 55(4): 1552-1569.

[20] HOEFLING H. A Path Algorithm for The Fused Lasso Signal Approximator[J]. Journal of Computational and Graphical Statistics, 2010, 19(4): 984-1066.

[21] DAYE Z J, JENG X J. Shrinkage and Model Selection with Correlated Variables via Weighted Fusion[J]. Computational Statistics & Data Analysis, 2009, 53(4): 1284-1298.

[22] BONDELL H D, REICH B J. Simultaneous Regression Shrinkage, Variable Selection, and Supervised Clustering of Predictors with OSCAR[J]. Biometrics, 2008, 64(1): 115-123.

[23] ZENG X R, FIGUEIREDO M A. Solving OSCAR Regularization Problems by Fast Approximate Proximal Splitting Algorithms [J]. Digital Signal Processing, 2014, 31(1): 124-135.

[24] YUAN M, LIN Y. Model Selection and Estimation in Regression with Grouped Variables[J]. Journal of the Royal Statistics Society: Series B (Statistical Methodology), 2006, 68(1): 49-67.

[25] JACOB L, OBOZINSKI G, VERT J P. Group Lasso with Overlap and Graph Lasso[C]// Annual International Conference on Machine Learning. ACM Digital Library, 2009: 433-440.

[26] LIU J, YE J. Moreau-Yosida Regularization for Grouped Tree Structure Learning[C]// Annual Conference on Neural Information Processing Systems. ACM Digital Library, 2010: 1459-1467.

[27] SIMON N, FRIEDMAN J, HASTIE T, et al. A Sparse-Group Lasso[J]. Journal of Computational and Graphical Statistics, 2013, 22(2): 231-245.

[28] FRIEDMAN J, HASTIE T, TIBSHIRANI R. A Note on the Group Lasso and a Sparse Group Lasso[J]. arXiv: 1001.0736, 2010.

[29] LIU J W, CUI L P, LIU Z Y, et al. Sparse on the Regularized Sparse Models[J]. Chinese Journal of Computers, 2015, 38(7): 1307-1325.

[30] YE J, LIU J. Sparse Methods for Biomedical Data [J]. ACM SIGKDD Explorations Newsletter, 2012, 14(1): 4-14.

[31] VINCENT M, HANSEN N R. Sparse Group Lasso and High Dimensional Multinomial Classification[J]. Computational Statistics & Data Analysis, 2014, 71: 771-786.

[32] NESTEROV Y. Smooth Minimization of Non-Smooth Functions [J]. Mathematical Programming, 2005, 103(1): 127-152.

[33] NESTEROV Y. Gradient Methods for Minimizing Composite Objective Function [J]. Mathematical Programming, 2013, 140(1): 125-161.

[34] TIBSHIRANI R. Regression Shrinkage and Selection via the Lasso[J]. Journal of the Royal Statistical Society: Series B (Methodological),1996,58(1):267-288.

[35] SHEVADE S K,KEERTHI S S. A Simple and Efficient Algorithm for Gene Selection Using Sparse Logistic Regression[J]. Bioinformatics,2003,19(17):2246-2253.

[36] HU Y,REN Y,WANG Q. A Feature Selection Based on Network Structure for Credit Card Default Prediction[C]// Chinese Conference on Computer Supported Cooperative Work and Social Computing. Springer,2019:275-286.

[37] WANG M,WANG C,YU X J,et al. Community Detection in Social Networks:an In-depth Benchmarking Study with a Procedure-oriented Framework[C]// International Conference on Very Large Data Bases. ACM Digital Library,2015,8(10):998-1009.

[38] YUAN L,LIU J,YE J. Efficient Methods for Overlapping Group Lasso[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,2013,35(9):2104-2116.

[39] XU X,YURUK N,FENG Z,et al. SCAN:a Structural Clustering Algorithm for Networks[C]// International Conference on Knowledge Discovery and Data Mining. ACM Digital Library,2007:824-833.

HU Yan-mei, born in 1984, Ph.D, associate professor, master supervisor, is a member of China Computer Federation. Her main research interests include data mining, social and information networks analysis, machine learning and evolutionary computation.