文章编号:1671-4598(2018)03-0258-05 DOI:10.16526/j.cnki.11—4762/tp.2018.03.064 中图分类号:TP39 文献标识码:A

基于不相似性的软件缺陷预测算法

张雪莹[1],李瑞贤[2]
(1. 中国电子科学研究院,北京 100041;2. 北京自动化控制设备研究所,北京 100074)

摘要:软件缺陷预测是典型的类不均衡学习问题,其中有缺陷的样本数量远少于无缺陷的样本数量,但有缺陷的样本通常是预测的重点;现有的软件预测模型大多建立在基于静态度量元的软件缺陷数据集上,重点关注如何平衡类分布,而忽略了数据集中属性特征对软件缺陷的判别能力;当软件缺陷数据集中的属性特征对类目标概念缺乏判别能力时,传统机器学习算法难以构建有效的软件缺陷预测模型,从而无法获得有效的预测性能;为此,提出了一种基于不相似性的软件缺陷预测算法,通过改善软件缺陷数据集中属性的判别能力,进而提升软件缺陷预测性能;实验证明:基于不相似性的软件缺陷预测算法能够有效地改善传统机器学习算法在软件缺陷数据集上的预测性能。

关键词:类不均衡学习;软件缺陷预测;原型选择;不相似性转换

Dissimilarity—Based Software Defect Prediction Algorithm

Zhang Xueying[1], Li Ruixian[2]
(1. China Academy of Electronics and Information Technology, Beijing 100041, China;
2. Beijing Automatic Control and Equipment Institute, Beijing 100074, China)

Abstract: Software defect prediction is a typical imbalance learning problem, in which the number of 'True' (defective) examples is far less than that of 'False' (Non—defective) ones. The minority class is often the learning objective. However, the existing researchers devote on building the prediction model on the static metric—based software defect data set, focusing on how to balance the imbalance class distribution, but ignoring the effect from the discrimination ability of features in the context of the software defect data set. Therefore, a dissimilarity—based software defect prediction algorithm (DSDPA) is proposed, which can be used for effectively increasing the performance of software defect prediction.

Keywords: class imbalance learning; software defect prediction; prototype selection; dissimilarity transformation

0 引言

软件缺陷数据集中有缺陷的样本数量往往比无缺陷的样本数量少得多,因此,软件缺陷预测可被视作一个类不均衡学习问题。在类不均衡学习学习过程中,不同类别的误分代价各不相等,少数类(有缺陷)的误分代价远高于多数类(无缺陷)的误分代价,为尽可能地降低误分代价,预测算法更重视那些有缺陷的少数类样本的预测结果。然而,传统的分类算法通常建立在类分布均衡且误分代价相等的前提下,以最小化分类误差为最终目标,因此直接采用决策树分类[1-3]、神经网络[3]、贝叶斯分类[4]、支持向量机[3-6]及k—最近邻分类[1,7]等传统的机器学习算法并不能获得较好的软件缺陷预测性能。

近年来,类不均衡学习问题受到了学术界的广泛关注,机器学习和数据挖掘领域专家们在AAAI'00[8]、ICML'03[9]及ACM SIGKDD'04[10]等权威研讨会上,对类不均衡问题的本质、解决方法及其性能评估指标进行了深入地探索与研究,并从数据层和算法层两方面提出了许多行之有效的解决方法。

数据层方法,主要通过(1)抽样或生成新样本的方式,使类分布恢复均衡,如随机欠抽样[11](RUS)和随机过抽样[12](ROS)。重复抽样可以平衡类分布,但欠抽样往往会忽略某些重要样本,导致信息缺失;反之,过抽样会引入大量副本,产生冗余信息,导致过拟合。

算法层方法,侧重于改进已有分类算法或研究新的分类算法,以更好地解决类不均衡学习问题。(1)"One—Class Learning"方法[13],该方法仅在多数类上构建分类模型,难以准确预测少数类;(2)组合学习方法,通过重复抽样构建多个分类模型、迭代更新训练样本的权重或组合多个决策树的方式,获得稳定的分类精度,如Bagging[14]、Boosting[15]及Random Forest[16]等算法。特别是,当分类模型间存在显著差异时,组合分类模型比基本分类模型更准确,但其计算量大且复杂度较高;(3)代价敏感分析,以最小化误分类代价为学习目标,如MetaCost[17]不依赖于分类算法,且可应用于任意形式的代价矩阵上,但如何确定代价矩阵目前仍然是一个难题。

学者们[18]发现:除不均衡的类分布以外,小样本、高维度及问题复杂度等因素也会影响算法性能。预测算法本质上是在挖掘数据集中属性特征与类目标概念间内在的关联关系,并建立相应的形式化预测模型。当不均衡数据集自身属性对类目标概念缺乏判别能力时,预测算法的性能将会有所下降,特别是少数类样本的预测。

现有的类不均衡学习方法侧重于如何调整类分布或改进算法,而忽略了类不均衡数据集中属性特征的判别能力。为了提升数据集属性特征的判别能力,Pekalska和Duin等人[19]提出了一种基于不相似性的表示法,用样本间不相似性替代原始属性特征,不仅保留了数据集原有统计信息,也能够获

收稿日期:2018-01-12; 修回日期:2018-01-30。

作者简介:张雪莹(1987-),女,黑龙江哈尔滨人,博士,工程师,主要从事数据挖掘方向的研究。

取到数据集内在的结构信息，该方法已被证实有利于预测模型的构建[20-21]。基于已有的研究成果，提出了一种基于不相似性的软件缺陷预测算法（Dissimilarity－based Software Defect Prediction Algorithm，DSDPA），用以提升软件缺陷的预测性能。

DSDPA 主要由原型选择、不相似性转换和缺陷预测三部分组成。实验过程中，采用随机选择法进行原型选择，欧几里德距离衡量样本间的不相似性，将 18 个软件缺陷数据集转换到不相似性空间；然后，采用最近邻分类算法（1－NN，IB1）、决策树（Decision Tree，DT）、神经网络（Multi－layer Perceptron，MLP）、朴素贝叶斯（Naive Bayes，NB）、随机森林（Random Forest，RF）和支持向量机（Support Vector Machine，SVM）6 种传统机器学习算法，在基于不相似空间中的软件缺陷数据集上构建预测模型；最后，对比分析了基于不相似性的软件缺陷预测方法 DSDPA 的预测性能。实验结果表明，DSDPA 能够有效地改善软件缺陷预测的准确性。

1　算法原理及框架

当利用机器学习算法预测软件缺陷时，预测模型的建立通常基于静态度量元的软件缺陷数据集上，而基于不相似性的软件缺陷预测算法（DSDPA）则是将原始数据集预先映射到不相似性空间，而后在不相似性空间中构建软件缺陷预测模型。DSDPA 主要由原型选择、不相似性转换和分类三部分组成。图 1 给出了 DSDPA 的基本框架，主要由基于不相似性预测模型的构建和软件缺陷预测两大环节组成。

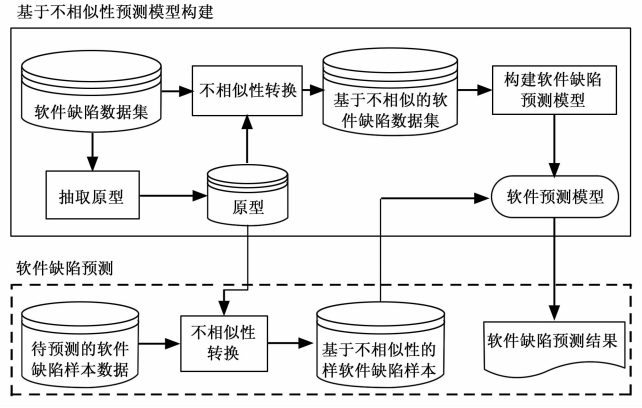

图 1　基于不相似性的软件缺陷预测算法框架

1）基于不相似性预测模型的构建。

基于不相似性的软件缺陷预测算法的构建过程主要由原型选择、不相似性转换及构建预测模型三部分组成。首先，采用原型选择方法从原始数据集中筛选出具有代表性的样本作为原型，创建原型集；然后，计算原始数据集与原型集样本间的不相似性，从而将其映射到相应的不相似性空间中；最后，利用传统分类算法在不相似性空间中构建软件缺陷预测模型。

2）软件缺陷预测。

当未知样本到来时，首先计算未知样本与原型集中各样本间的不相似性，将其映射到到不相似性空间；然后，利用已构建的软件缺陷预测模型对不相似性空间中的未知样本进行预测，即可获悉未知样本是否有缺陷。

1.1　原型选择

原型选择旨在从原始数据集中选取具有代表性的样本作为原型，作为不相似性转换时的参照。为了更好的选取原型，学者们提出了基于共享最近邻（Shared Nearest Neighbors，SNN）的 Jarvis－Patrick clustering（JPC）算法[21]、随机选择[22]（RandomC，RC）、线性规划[23]（LinPro）、属性选择[24]（FeaSel）、模式搜索[25]（ModeSeek）、基于聚类的线性规划[26]（KCenters－LP）及编辑压缩（EdiCon）等方法。其中，随机选择法（RC），即随机地从原始数据集中抽取指定数量的样本作为原型，是最简单且有效的一种原型选择方法。Pekalska 等人[26,21]对比分析了上述原型方法对基于不相似性分类方法性能的影响，实验结果表明：RC 和 KCenters 总体表现较好，但 RC 更便捷。

为了保证 DSDPA 算法的性能，选用 RC 作为原型选择方法，从原始软件缺陷数据集中抽取具有代表性的样本，创建原型集。假设 D 代表一个软件缺陷数据集，属于二类分类问题，即 $C=\{c_1,c_2\}$，D_t 为训练集，D_1 和 D_2 分别代表有缺陷和无缺陷类的训练集。从 D 抽取 r 个样本作为原型集合 P，利用随机选择的方法分别从 D_1 和 D_2 中随机抽取 r_1 和 r_2 个样本，使原型集 $P=\{r_1,r_2\}$。

1.2　不相似性转换

不相似性转换旨在将原始数据集映射到不相似性空间，图 2 给出了不相似性转换的详细过程。

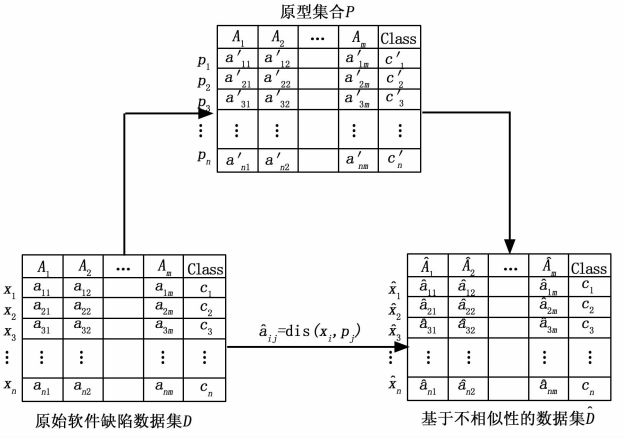

图 2　不相似性转换

假设 $D=\{x_1,x_2,\ldots,x_n\}$ 代表样本数量为 n 的软件缺陷数据集。其中，$x_i=\{a_{i1},a_{i2},\ldots,a_{im},c_i\}$ 代表数据集 D 中第 i 个样本；x_i 由 m 个独立属性和一个类属性组成。$P=\{p_1,p_2,\ldots,p_r\}$ 表示由 r 个具有代表性的样本构成的原型集，其中 $p_i=\{a_{i1}',a_{i2}',\ldots,a_{im}',c_i'\}$ 代表第 i 个原型。

参照原型集 P，通过计算原始的软件缺陷数据集 D 与原型集 P 中样本间的不相似性，将数据集 D 映射到相应的不相似性空间，得到基于不相似性的数据集 $\hat{D}=\{\hat{x}_{i1},\hat{x}_{i2},\ldots,\hat{x}_{in}\}$，$\hat{D}$ 由 n 个样本构成。每个样本 $\hat{x}_i=\{\hat{a}_{i1},\hat{a}_{i2},\ldots,\hat{a}_{ir},c_i\}$ 由 $r+1$ 个属性组成，其中 $\hat{a}_{ij}$ 代表样本 x_i 与原型 p_j 间的不相似性。

当评估密集、连续型样本间的不相似性时，基于度量的距离样本间的不相似性通常用距离度量来描述，距离越大，越不相似；反之，则越相似。目前最常用的距离度量有欧几里德距

离、曼哈顿距离及闵可夫斯基距离。其中，闵可夫斯基距离是欧几里德距离和曼哈顿距离的推广，其计算方法见公式（1）：

$$\hat{a}_{ij} = dis(x_i, p_j) = \left(\sum_{k=1}^{m} | a_{ik} - a_{jk}' |^{l}\right)^{\frac{1}{l}} \quad (1)$$

式中，l 是实数，$l \geqslant 1$。

当 $l = 1$ 时，曼哈顿距离，即 L_1 范数；

当 $l = 2$ 时，欧几里德距离，即 L_2 范数，常用于度量密集、连续的数据集中样本间的不相似性；

当 $l = \infty$ 时，上确界距离，又称 L_∞ 范数和切比雪夫距离，度量样本间的最大值差。

不相似性转换过程

输入：$D = \{x_1, x_2, \ldots, x_n\}$ 为原始的软件缺陷数据集；

$P = \{p_1, p_2, \ldots, p_r\}$ 为原型集合；

输出：$\hat{D} = \{\hat{x}_{i1}, \hat{x}_{i2}, \ldots, \hat{x}_{im}\}$ 为不相似性空间中的软件缺陷数据集；

1：for each $x_i \in D$ do

2：for each $p_j \in P$ do

3：$\hat{a}_{ij} = dis(x_i, p_j)$ ；//计算样本间的不相似性

4：end

5：$\hat{x}_i = \{\hat{a}_{i1}, \hat{a}_{i2}, \ldots, \hat{a}_{ir}, c_i\}$ ；

6：return $\hat{D}$ ；

由于软件缺陷数据集中大多数属性特征是密集连续的，所以选用基于度量的欧几里德距离来度量样本间的不相似性，以实现软件缺陷数据集从原始特征空间到不相似性空间的转换。

1.3 时间复杂度分析

DSDPA 算法由原型选择、不相似性转换及分类三个环节组成，其算法时间复杂度即各环节时间复杂度的总和。给定一个软件缺陷不均衡数据集 D，样本数量为 n，属性数量为 m，利用 DSDPA 算法在 D 上进行缺陷预测时，各环节时间复杂度的计算方法如下所述：

1）原型选择的时间复杂度。

从样本数量为 n 的不均衡数据集中选取 r 个原型时，不同的原型选择方法的时间复杂也不相同。随机选择方法无放回抽样，重复 r 次的时间复杂度为 $T_{RC} = O(r)$。

2）不相似性转换的时间复杂度。

不相似性转换旨在计算原始的软件缺陷数据集与原型集中样本间的不相似性，从而将其转换到不相似性空间，不相似性转换的时间复杂度为 $T_{DT} = O(n \cdot r)$。

3）缺陷预测的时间复杂度。

在样本数量为 n，属性数量为 $r+1$ 的基于不相似性的软件缺陷数据集上进行预测时，时间复杂度依赖于所选用的机器学习算法，$T_C = O(C(n, r+1))$。

DSDPA 算法的时间复杂度 $T_{DSDPA} = T_{RC} + T_{DT} + T_C$，即 $T_{DSDPA} = O(r) + O(n \cdot r) + O(C(n, r+1))$。

2 实验结果与分析

为了保证实验的客观性和可再现性，在公开的软件缺陷数据集上对 DSDPA 的预测性能进行了实验评估与验证。

2.1 实验数据

本实验在 18 来自 Promise[27] 数据库的软件缺陷数据集，其中 5 个源自 PROMISE 软件工程数据库，13 个源自美国宇航局（NASA）MDP 项目的数据集。MDP 是由美国宇航局提供一个软件度量库，并通过网站提供给普通用户。MDP 数据存储了系统在模块（函数/方法）级的软件产品的度量数据和相关的缺陷数据；实验数据集的样本数量分布在 36～171 68 之间，属性数量分布于 22～41 之间，不均衡率分布在 1.049 2～138.21 之间。表 1 给出了 18 个软件缺陷数据集的统计信息，其中 I、F、Cmin、Cmaj 和 IR 分别代表样本数量、属性数量、少数类样本数量（有缺陷的样本数量）、多数类样本数量（无缺陷的样本数量）以及不均衡率（Cmaj/ Cmin）。

表 1 18 软件缺陷数据集的统计信息

ID	数据集	I	F	Cmin	Cmaj	IR
1	ar1	121	30	9	112	12.444
2	ar3	63	30	8	55	6.875
3	ar4	107	30	20	87	4.35
4	ar5	36	30	8	28	3.5
5	ar6	101	30	15	86	5.7333
6	CM1	505	41	48	457	9.5208
7	JM1	10878	22	2102	8776	4.1751
8	KC1	2107	22	325	1782	5.4831
9	KC3	458	41	43	415	9.6512
10	KC4	125	41	61	64	1.0492
11	MC1	9466	40	68	9398	138.21
12	MC2	161	41	52	109	2.0962
13	MW1	403	41	31	372	12
14	PC1	1107	41	76	1031	13.566
15	PC2	5589	41	23	5566	242
16	PC3	1563	41	160	1403	8.7688
17	PC4	1458	41	178	1280	7.191
18	PC5	17186	40	516	16670	32.306

2.2 实验设置

为了全面地验证基于不相似性软件缺陷预测算法（DSDPA）的有效性，并保证实验的可再现性，本节对实验中的各环节进行了如下设置：

1）软件缺陷预测算法。

不同的机器学习算法在软件缺陷数据集上的预测性能也不相同。为了考察 DSDPA 能否有效地改善软件缺陷数据集上的预测性能，采用了 6 种最常用的机器学习算法作为候选预测算法[28-29]，包括：基于实例学习的 k－最近邻算法（1－NN，IB1）、决策树（J48）、神经网络（MLP）、朴素贝叶斯（NB）、随机森林（Random Forest）和支持向量机（SVM），用以对不相似性空间中的软件缺陷数据集进行预测。

2）性能评估方法。

在评估不均衡学习方法的性能时，采用 10×10 折交叉验证，充分利用数据信息的同时，尽可能地减少随机序列产生的偶然误差。

3）性能评价指标。

为了评价软件缺陷预测性能，特别是有缺陷样本的预测准确率，采用 AUC 评估各算法在软件缺陷数据集上的预测准确性。

2.3 结果与分析

表 2 给出了采用 DSDPA 与原始数据上（Org）时，最近邻（IB1）、决策树（J48）、神经网络（MLP）、朴素贝叶斯（NB）、随

表 2 算法性能比较(AUC)

Data	IB1		J48		MLP		NB		RF		SVM	
	Org	DSDPA	Org	DSDPA	Org	DSDPA	Org	DSDPA	Org	DSDPA	Org	DSDPA
ar1	0.57	0.77	0.50	0.55	0.67	0.74	0.63	0.70	0.62	0.69	0.45	0.50
ar3	0.55	0.69	0.60	0.75	0.67	0.83	0.67	0.82	0.62	0.78	0.4	0.62
ar4	0.60	0.60	0.62	0.62	0.71	0.71	0.80	0.80	0.76	0.75	0.50	0.63
ar5	0.57	0.71	0.61	0.76	0.70	0.88	0.72	0.90	0.69	0.82	0.41	0.75
ar6	0.63	0.67	0.56	0.56	0.76	0.76	0.72	0.73	0.65	0.65	0.5	0.56
CM1	0.55	0.54	0.56	0.58	0.71	0.67	0.75	0.70	0.73	0.70	0.5	0.50
JM1	0.61	0.63	0.67	0.67	0.71	0.71	0.69	0.69	0.72	0.72	0.50	0.50
KC1	0.66	0.73	0.69	0.69	0.79	0.79	0.79	0.79	0.80	0.80	0.50	0.52
KC3	0.59	0.59	0.63	0.58	0.74	0.66	0.82	0.68	0.77	0.73	0.5	0.52
KC4	0.69	0.69	0.78	0.78	0.75	0.75	0.77	0.77	0.79	0.79	0.69	0.72
MC1	0.76	0.94	0.77	0.77	0.90	0.90	0.91	0.92	0.88	0.88	0.5	0.50
MC2	0.69	0.67	0.62	0.60	0.72	0.71	0.73	0.72	0.71	0.69	0.54	0.60
MW1	0.61	0.59	0.54	0.54	0.69	0.65	0.76	0.75	0.68	0.68	0.5	0.50
PC1	0.67	0.66	0.67	0.68	0.74	0.81	0.76	0.79	0.82	0.81	0.5	0.50
PC2	0.49	0.76	0.51	0.51	0.85	0.86	0.84	0.86	0.67	0.68	0.5	0.50
PC3	0.62	0.66	0.61	0.65	0.79	0.80	0.77	0.76	0.80	0.81	0.5	0.50
PC4	0.69	0.71	0.74	0.74	0.89	0.89	0.84	0.84	0.92	0.92	0.5	0.56
PC5	0.75	0.93	0.78	0.78	0.94	0.94	0.83	0.94	0.94	0.94	0.51	0.54
AVG	0.63	0.70	0.64	0.66	0.76	0.78	0.77	0.79	0.75	0.77	0.50	0.56

机森林（RF）及支持向量机（SVM）6 种机器学习方法在 18 个软件缺陷数据集上的预测性能 AUC。由表可见，提出的 DSDPA 方法能够有效地改善传统机器学习方法在软件缺陷数据集上的预测性能，特别是在使用 IB1 和支持向量机 SVM 算法进行软件缺陷缺陷预测时，IB1 算法在软件缺陷数据集上的分类性能平均提升了 11.11%；SVM 算法在软件缺陷数据集上的分类性能平均提升了 12%。J48、MLP、NB 算法在软件缺陷数据集上的平均分类性能也得到了提升，提升率分别为 3.12%、2.5%、2.6%及 2.7%。

3　结论

从改善软件缺陷数据集中属性特征判别能力的角度出发，提出了一种基于不相似性的软件缺陷预测算法（DSDPA），主要由原型选择、不相似性转换及缺陷预测三部分组成。针对最近邻、决策树、神经网络、朴素贝叶斯、随机森林及支持向量机 6 种传统机器学习算法，对比分析了 DSDPA 在软件缺陷数据集上的预测性能 AUC。实验结果表明：DSDPA 算法能够有效地改善传统机器学习算法在软件缺陷数据集上的预测性能。

参考文献：

[1] Chawla NV, Japkowicz N, Kotcz A. Editorial: special issue on learning from imbalanced data sets [J]. ACM SIGKDD Explorations Newsletter, 2004, 6 (1): 1 - 6.

[2] Batista GE, Prati RC, Monard MC. A study of the behavior of several methods for balancing machine learning training data [J]. ACM SIGKDD Explorations Newsletter, 2004, 6 (1): 20 - 29.

[3] Japkowicz N, Stephen S. The class imbalance problem: A systematic study [J]. Intelligent Data Analysis, 2002, 6 (5): 429 - 449.

[4] Ezawa KJ, Singh M, Norton SW. Learning goal oriented Bayesian networks for telecommunications risk management [A]. International Conference on Machine Learning [C]. 1996: 139 - 147.

[5] Raskutti B, Kowalczyk A. Extreme re-balancing for SVMs: a case study [J]. ACM SIGKDD Explorations Newsletter, 2004, 6 (1): 60 - 69.

[6] Wu G, Chang EY. Class-boundary alignment for imbalanced dataset learning [C]. ICML 2003 workshop on learning from imbalanced data sets II, 2003: 49 - 56.

[7] Mani I, Zhang I. kNN approach to unbalanced data distributions: a case study involving information extraction [C]. Proceedings of Workshop on Learning from Imbalanced Datasets, 2003.

[8] Japkowicz N, others. Learning from imbalanced data sets: a comparison of various strategies [C]. AAAI workshop on learning from imbalanced data sets, 2000, 68.

[9] Japkowicz N. Class imbalances: are we focusing on the right issue [C]. Workshop on Learning from Imbalanced Data Sets II, 2003, 1723: 63.

[10] Chawla NV, Japkowicz N, Kotcz A. Editorial: special issue on learning from imbalanced data sets [J]. ACM SIGKDD Explorations Newsletter, 2004, 6 (1): 1 - 6.

[11] Kotsiantis SB, Pintelas PE. Mixture of expert agents for handling imbalanced data sets [J]. Annals of Mathematics, Computing & Teleinformatics, 2003, 1 (1): 46 - 55.

[12] Kubat M, Matwin S, others. Addressing the curse of imbalanced training sets: one-sided selection [A]. International Conference on Machine Learning [C]. 1997: 179 - 186.

[13] Manevitz LM, Yousef M. One-class SVMs for document classification [J]. The Journal of Machine Learning Research, 2002, 2: 139 - 154.

[14] Breiman L. Bagging predictors [J]. Machine Learning, 1996, 24 (2): 123 - 140.

[15] Joshi MV, Kumar V, Agarwal RC. Evaluating boosting algorithms to classify rare classes: Comparison and improvements [A]. IEEE International Conference on Data Mining [C]. 2001: 257 - 264.

[16] Breiman L. Random forests [J]. Machine Learning, 2001, 45 (1): 5 - 32.
[17] Domingos P. Metacost: A general method for making classifiers cost—sensitive [A]. Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining [C]. 1999: 155 - 164.
[18] Batista GE, Prati RC, Monard MC. A study of the behavior of several methods for balancing machine learning training data [J]. ACM SIGKDD Explorations Newsletter, 2004, 6 (1): 20 - 29.
[19] Pkekalska E, Duin RP. The dissimilarity representation for pattern recognition: foundations and applications [M]. World Scientific, 2005.
[20] Pkekalska E, Duin RPW. Dissimilarity representations allow for building good classifiers [J]. Pattern Recognition Letters, 2002, 23 (8): 943 - 956.
[21] Zhang XY, Song QB, Zhang KY, He L, Jia XL. A dissimilarity—based imbalance data classification algorithm [J]. Applied Intelligence, 2015, 42 (3): 544 - 565.
[22] Pekalska E, Paclik P, Duin RP. A generalized kernel approach to dissimilarity—based classification [J]. Journal of Machine Learning Research, 2002, 2: 175 - 211.
[23] Bradley PS, Mangasarian OL, Street W. Feature selection via mathematical programming [J]. Informs Journal on Computing, 1998, 10: 209 - 217.
[24] Jain A, Zongker D. Feature selection: Evaluation, application, and small sample performance [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997, 19 (2): 153 - 158.
[25] Cheng Y. Mean shift, mode seeking, and clustering [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1995, 17 (8): 790 - 799.
[26] Pekalska E, Duin RPW, Paclik P. Prototype selection for dissimilarity—based classifiers [J]. Pattern Recognition, 2006, 39 (2): 189 - 208.
[27] Boetticher G, Menzies T, Ostrand T. Promise repository of empirical software engineering data. Department of Computer Science [J]. West Virginia University, 2007.
[28] 马 樱. 基于机器学习的软件缺陷预测技术研究 [D]. 成都. 电子科技大学, 2012.
[29] 程 俊, 张雪莹, 李瑞贤. 基于元学习的软件缺陷预测推荐方法 [J]. 中国电子科学研究院学报, 2015, 10 (6): 620 - 627.

(上接第 257 页)

以最大灵敏度的激励方式激励各个单元产生的四条波束的方向图如图 13 所示。

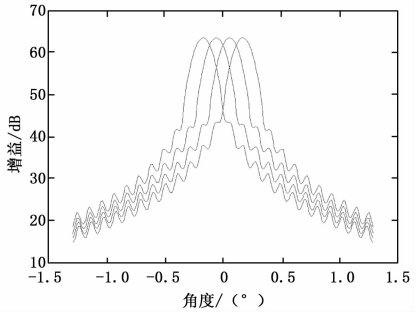

图 13 天线方向图

通过图 12 和图 13 可以看出，馈源阵列在整个频段内的驻波比在 1.051.35 之间，驻波性能良好。通过仿真还可以得到单个波束的效率最高可达 74.7%，增益达到 63.36 dB，达到项目要求的 60 dB，天线的噪声温度为 91.9 K，满足项目的要求，单个波束的灵敏度为 1.077 m2/K，相邻波束性能基本一致。

4 结束语

本文以 SKA 项目为依托，针对具体的反射面天线，进行了 Ku 频段的相控阵馈源设计。首先对焦面场进行分析，确定阵列的尺寸，然后根据项目需求确定阵元的形式，设计并以驻波比为目标对阵元进行优化。通过仿真对比了六边形阵列和矩形阵列对系统性能的影响，最终选定了六边形的布阵方式。接下来对阵元间距对灵敏度的影响进行仿真，得到了系统灵敏度随阵元间距的变化规律，得到了是系统灵敏度最优的阵元间距。最后对相控阵馈电反射面天线的模型进行仿真，可以看出系统的效率高，波束性能优良且近似，满足了项目的需求。

参考文献：

[1] 杜 彪, 伍 洋, 周建寨, 等. 平方公里阵中国验证天线光学设计 [J]. 中国科学：信息科学, 2015, 45: 1001 - 1014.
[2] Kildal P S, Johansson M, Hagfors T, et al. Analysis of a cluster feed for the Arecibo trireflector system using forward ray tracing and aperture integration [J]. IEEE Transactions on Antennas and Propagation, 1993, 41 (8): 1019 - 1025.
[3] Popenko N, Khaikin V, Lebedev M, et al. Highly effective array feed for RATAN - 600 radio telescope in a multibeam mode [A]. International Workshop on Teraherz and Applications [C]. Turunc - Marmaris, Turkey, 2009: 61 - 62.
[4] 王永根. 堆积均匀多波束成像反射面天线研究 [D]. 成都：电子科技大学, 2008.
[5] 伍 洋, 杜 彪, 金乘进, 等. 射电望远镜相控阵馈源技术 [J]. 电波科学学报, 2013, 28 (2): 348 - 353.
[6] 李建斌, 彭 勃, 孙建民, 等. 射电天文站电磁环境测量方法及分析 [J]. 电波科学学报, 2009, 24 (3): 523 - 528.
[7] 孙建民, 罗 韬. 射电天文业务及干扰保护标准研究 [J]. 中国无线电, 2008 (4): 51 - 54.
[8] Hansen C, Warnick K F, Jeffs B D. Interference cancellation using an array feed design for radio telescopes [A]. International Symposium on Antennas and Propagation Society [C], 2004 (1): 539 - 542.
[9] 闫 丰, 杜 彪. 赋形卡式天线最佳吻合反射面的计算方法 [J]. 无线电通信技术, 2011, 41 (3): 38 - 40.
[10] 赵 卫, 叶 骞, 冯正进. 射电望远镜主动反射面控制技术简析 [J]. 现代雷达, 2011, 33 (5): 85 - 90.
[11] Rahmat Y—Samii. Array Feeds for Reflector Surface Distortion Compensation: Concepts and Implementation [J]. IEEE Antennas and Propagation Magazine, 1990: 20 - 26.
[12] Wu Y, Jin C, Warnick K. Design study of an L - band phased array feed for wide - field surveys and vibration compensation on FAST [J]. IEEE transactions on antennas and propagation, 2013, 61 (6): 3026 - 3033.
[13] Jeff B D, Warnick K F, Landon J, et al. Signal processing for phased array feed in radio anstronmy [J]. IEEE Journal of Selected Topics in Signal Processing, 2008, 2 (5): 635 - 646.