Subclass Maximum Margin Tree Error Correcting Output Codes

Fa Zheng[1,2] and Hui Xue[1,2(✉)]

[1] School of Computer Science and Engineering, Southeast University, Nanjing 210096, People's Republic of China
{faaronzheng,hxue}@seu.edu.cn
[2] Key Laboratory of Computer Network and Information Integration, Southeast University, Ministry of Education, Nanjing, People's Republic of China

Abstract. Error Correcting Output Codes (ECOC) is an effective method to handle multi-class classification problems, whose performance is heavily affected by encoding strategies. However, traditional encoding strategies are usually data-independent which more likely leads to a less representation coding matrix. Therefore, recent researches emphasize more on the data-dependent construction of coding matrix. However, these methods usually can not guarantee that the binary problems in coding matrix are linearly separable. When the problems are linearly non-separable, the difficulty of designing base classifiers will increase. Even the non-linear base classifiers can not ensure that they can handle each binary problem well, which will decrease the performance of ECOC. In this paper, we propose subclass maximum margin tree error correcting output codes (SM^{2}ECOC) which aims to make each binary problem simple and linearly separable. Concretely, SM^{2}ECOC firstly uses hierarchical clustering techniques to split original classes into a series of linearly separable subclasses. Then it takes the margin as a criterion to evaluate the separability among subclasses and guide the construction of coding matrix. As a result, the binary problems in coding matrix more likely tend to be linearly separable which can reduce the difficulty in base classifiers effectively. Experimental results show the superiority of SM^{2}ECOC.

Keywords: Error correcting output codes · Maximum margin tree Subclass

1 Introduction

Error Correcting Output Codes (ECOC) is applied widely in multi-class classification problems [5], which usually involves encoding and decoding steps. The encoding step addresses the multi-class problem into a series of binary problems by generating a coding matrix. Each row and column of the coding matrix represent a class and binary problem respectively. The coding matrix is first coded by $+1$ and -1 as Table 1(a) shown, where $+1$ and -1 show that the corresponding

X. Geng and B.-H. Kang (Eds.): PRICAI 2018, LNAI 11012, pp. 454–462, 2018.
https://doi.org/10.1007/978-3-319-97304-3_35

base classifier regards this class as positive and negative respectively. Allwein et al. [1] further presented a ternary coding matrix which introduces symbol zero into the coding matrix as Table 1(b) shown. The symbol zero means that the corresponding classification does not include the relevant class. The decoding step makes a prediction for unseen data. Firstly, the predictions of all base classifiers will be combined as an output code word. The class which has the most similar code word with output code word will be selected as the class of the unseen data.

The researches of ECOC can generally be divided into three fields: encoding strategy, decoding strategy and other strategy.

Encoding strategy focuses on building a representative coding matrix. One-Versus-All(OVA) [13] obtains a binary coding matrix by considering one class as positive while the rest as negative. One-Versus-One (OVO) [8] generates a ternary coding matrix, each column of coding matrix only regards two classes as positive and negative respectively. Dense random and sparse random [1] generate the coding matrix randomly, where the dense one randomly obtains a binary coding matrix and the sparse one randomly generates a ternary one. Hadamard ECOC [2] uses Hadamard matrix to build coding matrix. Hadamard ECOC can ensure enough separability between the row and column of the coding matrix.

All aforementioned encoding strategies are data-independent which leads to a longer code words or worse performance. Therefore, more and more attention has been paid to construct the code matrix by a data-driven way. Discriminant ECOC (DECOC) [15] utilizes sequential forward floating search and mutual information to generate the tree by a top-down strategy. Based on the partition of tree, a coding matrix is constructed. Hierarchical ECOC (HECOC) [11] and Maximum Margin Tree ECOC (M^2ECOC) [17] use different criteria: support vector domain description and maximum margin to estimate inter-class separability and further generate the partition of tree. Subclass problem-dependent ECOC (Subclass ECOC) [7] splits the linearly non-separable problem into linearly separable ones on the basis of DECOC.

Decoding strategy emphasizes on enhance the error-correcting capability. Traditional decoding strategies are based on the distance between code word and output code word, such as Hamming distance (HD) [13] and Euclidean distance [8]. Allwein et al. [1] shown the advantage of using a loss-based function to decode. Loss-weighted [6] decoding applies the performance of base classifiers to adjust the classification decision.

Other strategy aims to improving the performance of ECOC beyond the encoding and decoding. ECOC-optimizing node embedding [14] algorithm iteratively adds a new classifier to classify the most confusing binary problems. Through selecting the most representative base classifiers from a group of similar base classifiers, Anderson and Siome [16] avoided the efficiency problem of ECOC. Liu et al. [12] mined the relationships among the base classifiers and proposed using a unified objective function to represent the relationships and boost the learning performances.

In this paper, we focus on encoding strategy and propose Subclass Maximum Margin Tree Error Correcting Output Codes (SM^2ECOC) which emphasizes on

Table 1. Coding matrix for a four-class problem

(a) Binary

	h_1	h_2	h_3	h_4
C_1	+1	-1	-1	-1
C_2	-1	+1	-1	-1
C_3	-1	-1	+1	-1
C_4	-1	-1	-1	+1

(b) Ternary

	h_1	h_2	h_3	h_4	h_5	h_6
C_1	+1	+1	+1	0	0	0
C_2	-1	0	0	+1	+1	0
C_3	0	-1	0	-1	0	+1
C_4	0	0	-1	0	-1	-1

making each binary problem in coding matrix tend to be linearly separable. SM^{2}ECOC firstly adopts the hierarchical clustering algorithm to split the original classes into a series of simple and linearly separable subclasses. Then it uses the margin to estimate the separability among the classes and further guide the construction of coding matrix. Consequently, the binary problems in coding matrix are more likely reorganized to be linearly separable ones which further makes the selection of base classifiers more flexible.

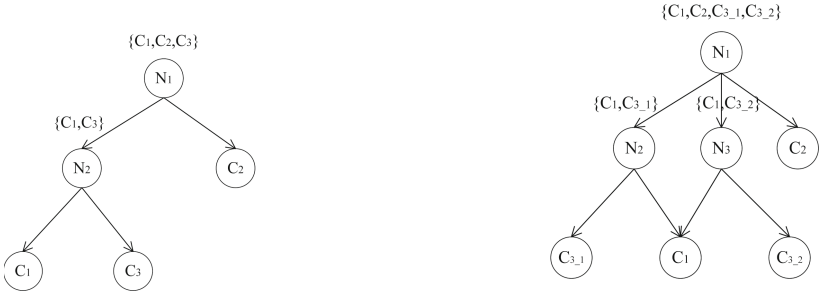
The paper is organized as follows. Section 2 introduces the related encoding algorithms Subclass ECOC. Section 3 introduces SM^{2}ECOC approach in detail. Section 4 shows the compared experiments. Finally, Sect. 5 concludes the paper.

2 Related Encoding Algorithms

Subclass ECOC [7] considers the linearly non-separable problems in coding matrix. On the basis of DECOC, when the partition is linearly non-separable, Subclass ECOC uses k-means to split the linearly non-separable partition into simpler and smaller sub-partitions. As Fig. 1 shown, when $\{C_1, C_3\}$ is linearly non-separable, different from DECOC, Subclass ECOC splits $\{C_1, C_3\}$ into two linearly separable partition $\{C_1, C_{3\_1}\}$ and $\{C_1, C_{3\_2}\}$. Therefore, through several times of decompositions, Subclass ECOC can transform a linearly non-separable binary problem into linearly separable ones. However, Subclass ECOC usually leads to an unstable splitting result because it uses the unstable k-means clustering algorithm hierarchically.

3 Subclass Maximum Margin Tree ECOC (SM^{2}ECOC)

The complexity of the binary problems in coding matrix has an important impact on the performance of ECOC. When the binary problems are linearly non-separable, even the non-linear base classifiers can not guarantee that they can handle each binary problem well. Subclass ECOC solves linearly non-separable problem by using k-means hierarchically in encoding step. Different from Subclass ECOC, SM^{2}ECOC splits subclasses before encoding and uses a more stable



(a) DECOC (b) Subclass ECOC

Fig. 1. Illustration of building the tree in DECOC and Subclass ECOC

clustering method. As a result, SM^{2}ECOC tries to recombine classes in order to make each binary problem tend to be linearly separable. SM^{2}ECOC involves two steps: splitting subclasses and building coding matrix. The splitting subclasses step splits original classes into a series of simple and linearly separable subclasses. The building coding matrix step uses M^{2}ECOC to explore data-dependent information from subclasses and obtain the coding matrix.

(a) Before (b) After

Fig. 2. Effect of DIANA algorithm DIANA

Splitting Subclasses: We use Divisive Analysis (DIANA) [9] to split original classes, which is a divisive hierarchical clustering algorithm. Different with k-means, DIANA can get a stable clustering result without specifying the number of clusters. Concretely, DIANA firstly splits data into two clusters with the biggest differences, and then for each cluster, repeats the process until the termination condition is satisfied. A threshold *termination* controls the partition granularity of each cluster. The number of the samples in each cluster becomes larger when the value of the *termination* gets bigger. The effect of DIANA is shown in Fig. 2. The original three classes are divided into eight linearly separable subclasses. Therefore, each binary problem in coding matrix is much more easier to be linearly separable.

Since that DIANA is an unsupervised algorithm, one cluster sometimes involves different classes. If the proportion of the data in one cluster compared to the whole data of this class is less than a threshold *drop*, we will consider these data as noises and drop them from the cluster, which can avoid generating redundant subclasses. Furthermore, if the data in one class are split into different subclasses, we will give them new labels which should contain the original class label information in order to backtrack the true label in prediction.

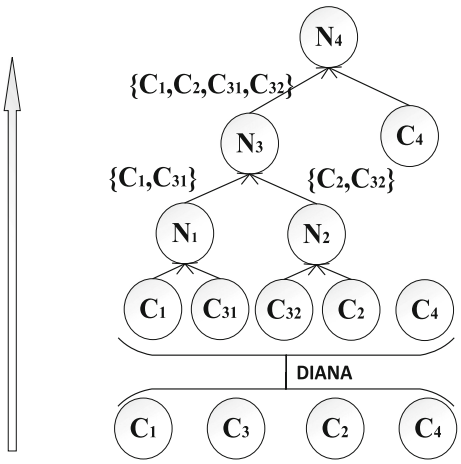


Fig. 3. Procedure of building a bottom-up maximum margin

Building Coding Matrix: After the splitting step, we obtain a series of linearly separable subclasses. In order to obtain the coding matrix, we use the margin as the separability measure standard to guide the construction of coding matrix. Margin is one of the basic concepts in Support Vector Machine (SVM) [4], which is defined as follow:

$$margin = \frac{2}{||\boldsymbol{w}||} \tag{1}$$

where $\boldsymbol{w}$ is the normal vector of hyperplane. The maximum margin can be computed between subclasses according to (1), and then a maximum margin matrix can be obtained through all maximum margins:

$$\begin{bmatrix} 0 & m_{12} & \dots & m_{1(k-1)} & m_{1k} \\ m_{21} & 0 & \dots & m_{2(k-1)} & m_{2k} \\ \dots & \dots & \dots & \dots & \dots \\ m_{(k-1)1} & m_{(k-1)2} & \dots & 0 & m_{(k-1)k} \\ m_{k1} & m_{k2} & \dots & m_{k(k-1)} & 0 \end{bmatrix}$$

where m_{ij} is the maximum margin between the ith and jth subclasses.

According to the maximum margin matrix, a bottom-up strategy is utilized to build the maximum margin tree. Concretely, for subclasses with the maximal

Algorithm 1. The Produce of SM^{2}ECOC

Input: The data set S has N samples
Threshold t for clustering method
Threshold d for dropping step
Output: The final subclasses *subclasses*

//Finding subclasses
cluster=Clustering(S, t);
//Ignoring the noise
subclasses=drop(*cluster*,d);
Relabeling the samples
while num(*subclasses*)>0 **do**
  // Computing maximum margin among *subclasses* according to (1)
  maximum_margin_matrix=*compute_maximum_margin*(*subclasses*)
  max_margin=max(*maximum_margin_matrix*)
  //Merging the subclasses corresponding to the maximal maximum margin
  new_subclass={row(*max_margin*),column(*max_margin*)}
  //Recording the left subclasses
  $left\_classes$=*subclasses* − *new_subclass*
  //Creating a partition of the tree
  partlength(part+1)=*new_subclass*
  subclasses={*new_subclass*, $left\_classes$}
end while
Obtaining a coding matrix according to (2)

maximum margin, we merge them into one until there is only one subclass left. As Fig. 3) shown, the original four classes are divided into five subclasses after splitting. $\{C_1\}$, $\{C_{31}\}$ and $\{C_{32}\}$, $\{C_2\}$ are merged as a new subclass orderly. Repeating the above process, we take $\{C_1, C_{31}\}$ and $\{C_2, C_{32}\}$ as another new subclass. Finally, all subclasses will be integrated as a new subclass.

We can obtain a coding matrix M according to the maximum margin tree as follows:

$$M(r,l) = \begin{cases} 1 & C_r \in P_l^{left} \\ 0 & C_r \notin P_l \\ -1 & C_r \in P_l^{right} \end{cases} \tag{2}$$

where $M(r,l)$ indicates the element lying in row r, column l of coding matrix. C_r stand for the class r, P_l^{left} is the left and P_l^{right} is the right partition of the lth partition.

Finally, we can obtain a coding matrix in which the binary problems are more likely to be linearly separable. The Produce of SM^{2}ECOC is as Algorithm 1 shown.

4 Experiments and Analyses

In this section, SM^{2}ECOC is compared with DECOC [15], HECOC [11], M^{2}ECOC [17] and Subclass ECOC [7] to validate its superiority.

In the experiments, ten multi-class UCI data sets [3] are used. Each data set is randomly split into two non-overlapping training and testing sets ten times. Seventy percent of the samples constitutes the training set and the rest constitutes the testing set. Moreover, the parameter set $\Theta = \{\Theta_{size}, \Theta_{perf}, \Theta_{impr}\}$ in Subclass ECOC is fixed to $\Theta_{size} = \frac{\lfloor J \rfloor}{50}$, $\Theta_{perf} = 0$ and $\Theta_{impr} = 0.95$ according to [7]. In SM^{2}ECOC, the threshold *termination* is selected according to the accuracy of classification from the interval $\{N, \lfloor N/2 \rfloor, \lfloor N/3 \rfloor ... \lfloor N/10 \rfloor\}$, where N is the size of data. The threshold *drop* is selected from the interval $\{0.05, 0.1\}$ also according to the accuracy of classification.

Linear base classifier Nearest Mean Classifier (NMC) and non-linear base classifier SVM are applied as the base classifier. In the radial basis function kernel, the regularization parameter C is set to 1 [7] and the width σ is selected from the interval $\{2^{-6}, 2^{-5}, ..., 2^5, 2^6\}$. Hamming distance (HD) [13] is used to as decoding strategy.

Table 2. Classification results (mean $\pm$ std) of NMC and HD on ten datasets ($\bullet$/$\circ$ indicates that our algorithm is significantly better or worse than other algorithms based on the t-test at 5% significance level)

Data set	DECOC	Subclass ECOC	HECOC	M^{2}ECOC	SM^{2}ECOC
Balance	80.32 ± 3.72	81.60 ± 5.27	81.60 ± 3.16	$83.42 \pm 3.24\bullet$	$\mathbf{84.01 \pm 2.72}$
Cmc	$46.29 \pm 1.19\bullet$	$46.31 \pm 1.18\bullet$	$45.48 \pm 2.38\bullet$	$46.97 \pm 1.60\bullet$	$\mathbf{50.88 \pm 0.90}$
Ecoli	$74.20 \pm 8.27\bullet$	$77.90 \pm 4.07\bullet$	$71.90 \pm 13.40\bullet$	$83.80 \pm 2.67\bullet$	$\mathbf{86.00 \pm 2.71}$
Glass	$42.15 \pm 11.40\bullet$	$49.08 \pm 6.97\bullet$	$42.31 \pm 10.00\bullet$	$49.23 \pm 4.53\bullet$	$\mathbf{62.15 \pm 5.04}$
Iris	$79.78 \pm 8.35\bullet$	$85.11 \pm 3.93\bullet$	$86.22 \pm 5.52\bullet$	$86.00 \pm 5.55\bullet$	$\mathbf{92.00 \pm 4.34}$
Lenses	$78.75 \pm 8.44\bullet$	$78.75 \pm 8.44\bullet$	$71.25 \pm 11.90\bullet$	$80.00 \pm 8.74\bullet$	$\mathbf{92.50 \pm 8.74}$
Tea	$53.26 \pm 5.64\bullet$	$55.87 \pm 5.80\bullet$	$54.13 \pm 6.44\bullet$	$56.09 \pm 5.21\bullet$	$\mathbf{61.52 \pm 5.33}$
Thyriod	$92.97 \pm 3.40\bullet$	$93.44 \pm 3.59\bullet$	$94.06 \pm 2.42\bullet$	$94.53 \pm 2.68\bullet$	$\mathbf{97.34 \pm 1.29}$
Vehicle	$40.04 \pm 3.37\bullet$	$43.94 \pm 5.22\bullet$	$40.67 \pm 3.40\bullet$	$49.25 \pm 4.86\bullet$	$\mathbf{66.22 \pm 3.26}$
Wine	$\mathbf{97.36 \pm 3.11}$	$\mathbf{97.36 \pm 3.11}$	95.09 ± 3.47	93.40 ± 3.59	94.72 ± 3.42
Average	68.51 ± 5.69	70.94 ± 4.76	68.27 ± 6.21	72.36 ± 4.27	$\mathbf{78.73 \pm 3.78}$
Win/tie/loss	8/2/0	8/2/0	8/2/0	9/1/0	/

From the Tables 2 and 3, it can be seen that SM^{2}ECOC can reach better or comparable performance, especially with linear base classifier. Meanwhile, the standard deviations and average accuracies are listed in the bottom of the tables. The pairwise t-tests [10] at 5% significance level are also shown. The win/loss with marker $\bullet$/$\circ$ indicates that SM^{2}ECOC can achieves significantly better/worse performance than the compared algorithms respectively. Otherwise, a tie with no marker. As the tables shown, SM^{2}ECOC achieves statistically better or comparable performance which just validates the effectiveness of SM^{2}ECOC and accords with our conclusions.

Table 3. Classification results(mean ± std) of SVM and HD on ten datasets (•/◦ indicates that our algorithm is significantly better or worse than other algorithms based on the t-test at 5% significance level)

Data set	DECOC	Subclass ECOC	HECOC	M^{2}ECOC	SM^{2}ECOC
Balance	76.84 ± 1.99•	77.17 ± 2.03•	89.79 ± 1.02	90.16 ± 0.92	**90.27 ± 0.94**
Cmc	51.97 ± 1.46•	52.83 ± 1.35	**54.52 ± 1.19**	53.21 ± 1.27	53.51 ± 1.45
Ecoli	80.30 ± 15.10	83.10 ± 6.59•	70.30 ± 15.50•	86.90 ± 2.59	**87.80 ± 1.87**
Glass	62.31 ± 9.84	64.15 ± 9.97	61.85 ± 5.88•	62.15 ± 3.78•	**68.00 ± 2.49**
Iris	72.89 ± 4.42•	74.67 ± 2.81•	**95.11 ± 3.60**	94.89 ± 1.83	94.89 ± 1.83
Lenses	56.25 ± 8.84•	58.75 ± 6.04•	62.50 ± 0.00•	75.00 ± 10.20	**82.50 ± 8.74**
Tea	44.35 ± 5.99•	47.83 ± 5.02•	51.30 ± 3.58•	55.43 ± 4.94•	**61.52 ± 5.98**
Thyriod	94.37 ± 1.32•	94.37 ± 1.32•	95.78 ± 1.06•	96.09 ± 1.11	**97.03 ± 1.37**
Vehicle	72.76 ± 2.37•	72.76 ± 2.37•	74.53 ± 1.93	75.08 ± 1.76	**75.35 ± 1.51**
Wine	96.23 ± 1.99•	97.36 ± 1.82•	98.30 ± 1.65	98.30 ± 1.97	**99.25 ± 0.97**
Average	70.83 ± 5.33	72.30 ± 3.93	75.40 ± 3.54	78.72 ± 3.04	**81.01 ± 2.72**
Win/tie/loss	8/2/0	8/2/0	5/5/0	2/8/0	/

5 Conclusion

In this paper, we present an encoding algorithm SM^{2}ECOC which focuses on making each binary problem linearly separable in coding matrix. SM^{2}ECOC includes two steps: splitting subclasses and building coding matrix. The splitting subclasses step splits original classes into a series of linearly separable subclasses by DIANA algorithm. Then, the building coding matrix step uses the maximum margin between subclasses to guide the construction of coding matrix. Experimental results show the superior performance of SM^{2}ECOC compared with some related algorithms.

Acknowledgments. This work is supported by the National Key R&D Program of China (No. 2017YFB1002801), the National Natural Science Foundations of China (Grant Nos. 61375057, 61300165 and 61403193), the Natural Science Foundation of Jiangsu Province of China (Grant No. BK20131298) and Fundamental Research Funds for the Central Universities (SJLX_160053) and Research Innovation Program for College Graduates of Jiangsu Province of China (SJLX_160053). It is also supported by Collaborative Innovation Center of Wireless Communications Technology.

References

1. Allwein, E.L., Schapire, R.E., Singer, Y.: Reducing multiclass to binary: a unifying approach for margin classifiers. J. Mach. Learn. Res. **1**(Dec), 113–141 (2000). https://doi.org/10.1162/15324430152733133
2. An-rong, Y., Xiang, X., Jing-ming, K.: Application of hadamard ECOC in multiclass problems based on SVM. Chin. J. Electron. **36**(1), 122–126 (2008). https://doi.org/10.3321/j.issn:0372-2112.2008.01.022

3. Asuncion, A., Newman, D.: UCI machine learning repository (2007)
4. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995). https://doi.org/10.1007/BF00994018
5. Dietterich, T.G., Bakiri, G.: Solving multiclass learning problems via error-correcting output codes. J. Artif. Intell. Res. **2**, 263–286 (1995). https://doi.org/10.1613/jair.105
6. Escalera, S., Pujol, O., Radeva, P.: Loss-weighted decoding for error-correcting output coding. In: VISAPP, no. 2, pp. 117–122 (2008)
7. Escalera, S., Tax, D.M., Pujol, O., Radeva, P., Duin, R.P.: Subclass problem-dependent design for error-correcting output codes. IEEE Trans. Pattern Anal. Mach. Intell. **30**(6), 1041–1054 (2008). https://doi.org/10.1109/TPAMI.2008.38
8. Hastie, T., Tibshirani, R., et al.: Classification by pairwise coupling. Ann. Stat. **26**(2), 451–471 (1998). https://doi.org/10.1214/aos/1028144844
9. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data: An Introduction to Cluster Analysis, vol. 344. Wiley, Hoboken (2009)
10. Kuncheva, L.I.: Combining Pattern Classifiers: Methods and Algorithms. Wiley, Hoboken (2004)
11. Lei, L., Xiao-Dan, W., Xi, L., Ya-Fei, S.: Hierarchical error-correcting output codes based on SVDD. Pattern Anal. Appl. **19**(1), 163–171 (2016)
12. Liu, M., Zhang, D., Chen, S., Xue, H.: Joint binary classifier learning for ECOC-based multi-class classification. IEEE Trans. Pattern Anal. Mach. Intell. **38**, 2335–2341 (2016). https://doi.org/10.1109/TPAMI.2015.2430325
13. Nilsson, N.J.: Learning Machines: Foundations of Trainable Pattern-Classifying Systems. McGraw-Hill, New York (1965)
14. Pujol, O., Escalera, S., Radeva, P.: An incremental node embedding technique for error correcting output codes. Pattern Recogn. **41**(2), 713–725 (2008). https://doi.org/10.1016/j.patcog.2007.04.008
15. Pujol, O., Radeva, P., Vitria, J.: Discriminant ECOC: a heuristic method for application dependent design of error correcting output codes. IEEE Trans. Pattern Anal. Mach. Intell. **28**(6), 1007–1012 (2006). https://doi.org/10.1109/TPAMI.2006.116
16. Rocha, A., Goldenstein, S.K.: Multiclass from binary: expanding one-versus-all, one-versus-one and ECOC-based approaches. IEEE Trans. Neural Netw. Learn. Syst. **25**(2), 289–302 (2014). https://doi.org/10.1109/TNNLS.2013.2274735
17. Zheng, F., Xue, H., Chen, X., Wang, Y.: Maximum margin tree error correcting output codes. In: Booth, R., Zhang, M.-L. (eds.) PRICAI 2016. LNCS (LNAI), vol. 9810, pp. 681–691. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42911-3_57