

Article

Beta Distribution-Based Cross-Entropy for Feature Selection

Weixing Dai  and Dianjing Guo *

School of Life Science and State Key Laboratory of Agrobiotechnology, G94, Science Center South Block, The Chinese University of Hong Kong, Shatin 999077, Hong Kong, China

* Correspondence: djguo@cuhk.edu.hk; Tel.: +852-26035745

Received: 15 June 2019; Accepted: 5 August 2019; Published: 7 August 2019



Abstract: Analysis of high-dimensional data is a challenge in machine learning and data mining. Feature selection plays an important role in dealing with high-dimensional data for improvement of predictive accuracy, as well as better interpretation of the data. Frequently used evaluation functions for feature selection include resampling methods such as cross-validation, which show an advantage in predictive accuracy. However, these conventional methods are not only computationally expensive, but also tend to be over-optimistic. We propose a novel cross-entropy which is based on beta distribution for feature selection. In beta distribution-based cross-entropy (BetaDCE) for feature selection, the probability density is estimated by beta distribution and the cross-entropy is computed by the expected value of beta distribution, so that the generalization ability can be estimated more precisely than conventional methods where the probability density is learnt from data. Analysis of the generalization ability of BetaDCE revealed that it was a trade-off between bias and variance. The robustness of BetaDCE was demonstrated by experiments on three types of data. In the exclusive or-like (XOR-like) dataset, the false discovery rate of BetaDCE was significantly smaller than that of other methods. For the leukemia dataset, the area under the curve (AUC) of BetaDCE on the test set was 0.93 with only four selected features, which indicated that BetaDCE not only detected the irrelevant and redundant features precisely, but also more accurately predicted the class labels with a smaller number of features than the original method, whose AUC was 0.83 with 50 features. In the metabonomic dataset, the overall AUC of prediction with features selected by BetaDCE was significantly larger than that by the original reported method. Therefore, BetaDCE can be used as a general and efficient framework for feature selection.

Keywords: cross-entropy; beta distribution; feature selection; machine learning; data mining

1. Introduction

When a machine learning algorithm learns from high-dimensional data, a huge amount of training samples is typically required to ensure the predictive power. This problem is known as the “curse of dimensionality” [1]. Feature selection is, therefore, commonly used to control the dimensionality in order to make the machine learning model more robust and the data more interpretable. Feature selection is widely used in various fields such as economics [2], fault diagnosis [3], DNA microarrays [4], etc.

Feature selection algorithms can be broadly divided into three categories: filter, wrapper, and embedded methods [5]. Filter methods select features without any learning algorithms and evaluate features individually with evaluation criterion such as mutual information [6], pair *t*-scores [7], and relief-based methods [8]. By contrast, wrapper methods evaluate subsets of features by learning algorithms such as random forest (RF), support vector machine (SVM), k-nearest neighbor (KNN), and neural network (NN), together with searching algorithms such as sequential forward selection [9], genetic algorithm [10], and fuzzy C-means [11,12], to search for the optimal subset of features.

Embedded methods are quite similar to wrapper methods except that an intrinsic building metric is used in the model during training. For example, the least absolute shrinkage and selection operator (LASSO) method uses a linear model to shrink many features to zero by L1 penalty. Other embedded methods include Bolasso [13], FeaLect [14], and support vector machine–recursive feature elimination (SVM-RFE) [15–17]. Among the three categories, filter methods are the most computationally efficient but usually have the worst performance. They are generally used as a preprocessing method for feature selection. Wrapper and embedded methods perform better than filter methods, but with a higher computational cost.

The evaluation function is the basis of feature selection that determines the quality of the final models [18]. Different from filter methods, most methods of high performance, including wrapper and embedded methods, are based on evaluation of the accuracy by cross-validation [19,20]. Other evaluation functions based on information theory include Akaike information criterion (AIC) and Bayesian information criterion (BIC), which are derived via the maximum entropy principle [21,22] and minimum description length [23] respectively. AIC and BIC both have a term of accuracy, and the difference lies in the penalty for the number of features. Under some circumstances, AIC is asymptotically equivalent to leave-one-out cross-validation [24,25]. However, cross-validation itself can result in over-optimistic estimates and poor generalization ability, thus leading to poor performance for feature selection [26,27].

Many recent studies on feature selection focused on the methods of optimizing feature subsets. New searching methods were applied to select the optimal feature subset. For example, whale optimization-related approaches were used for wrapper feature selection [28]. Other new approaches include evolutionary population dynamics and grasshopper optimization [29], binary dragonfly optimization [30], artificial bee colony optimization [31], etc. Combined methods of optimization for feature selection were applied in several studies. For example, Mafarja et al. combined the whale optimization with simulated annealing [32]. Faris et al. combined binary salp swarm algorithm with crossover scheme for feature selection [33]. However, few studies were conducted on evaluation function, which is another important element for feature selection other than the searching method.

Hence, we propose a new algorithm, named beta distribution-based cross-entropy (BetaDCE), for feature selection. BetaDCE tries to estimate probability density of prediction using beta distribution, and it selects feature subsets by cross-entropy that maximizes its generalization ability. In Section 2, we discuss the basic theory. The results of feature selection by BetaDCE and the discussion are given in Sections 3 and 4, respectively.

2. Materials and Methods

2.1. Beta Distribution-Based Cross-Entropy

The main objective of BetaDCE is feature selection, although it can also be used for prediction. Like KNN, BetaDCE tries to make full use of the information of adjacent samples. However, the characteristic of BetaDCE differing from KNN is that the probability density in the neighborhood is precisely modeled by beta distribution.

To evaluate feature subsets by BetaDCE, each training sample corresponds to an optimized number of nearest neighbors (K) which may differ from that of other samples. The parameter K is optimized by beta distribution K , and the expected value of probability is then used to compute the cross-entropy as the evaluation function for feature selection (Figure 1).

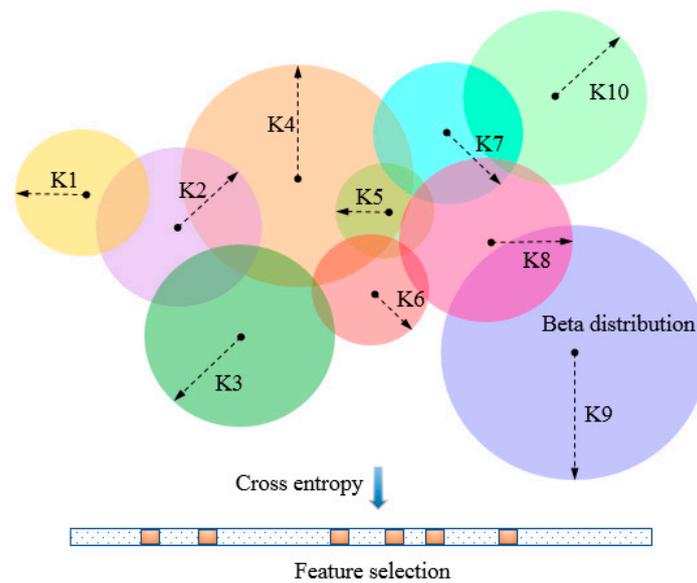


Figure 1. Description of beta distribution-based cross-entropy (BetaDCE). Each training sample corresponds to an optimized K , which is determined by beta distribution. The expected value of probability is then used to compute the cross-entropy as the evaluation function for feature selection.

2.1.1. Evaluation Function of BetaDCE

Suppose in a binary classification problem there are K nearest neighbors for the investigated sample, where K_A and K_B nearest neighbors belong to class A and B respectively. Although other definitions such as Mahalanobis distance are available, Euclidean distance is used here to define the nearest neighbors for simplicity. Let P_A be the probability of the sample belonging to class A . It can be concluded that P_A conforms to beta distribution with shape parameters of $K_A + 1$ and $K_B + 1$ in case no prior knowledge is provided. The expected value for P_A can be analytically expressed as

$$E[P_A] = \frac{K_a + 1}{K_a + K_b + 2} \tag{1}$$

Similarly, the equation of the expected value for P_B can also be obtained. Therefore, the expected value (E) for the investigated sample can be written as

$$E = \begin{cases} E[P_A], & E[P_A] \geq E[P_B] \\ E[P_B], & E[P_A] < E[P_B] \end{cases} \tag{2}$$

The overall cross-entropy for all the training samples can be calculated using the following formula:

$$H = -\frac{1}{n_A} \sum_{i=1}^{n_A} \ln E_i - \frac{1}{n_B} \sum_{i=1}^{n_B} \ln E_i \tag{3}$$

where E_i is the expected value for sample i , while n_A and n_B are the total numbers of samples for classes A and B , respectively. Equation (3) is used as the evaluation function of BetaDCE to optimize feature subsets.

2.1.2. Iteration and Optimization

In the procedure of BetaDCE, a feature subset is initialized and then updated iteratively until a stopping criterion is reached. We define an epoch as a circle of iteration. Epoch i means that all subsets consisting of i features in the training dataset have had an opportunity to update the optimized feature subset.

In epoch i , all subsets consisting of i features are evaluated and then sorted by the values of the evaluation function defined by Equation (3). If the minimum value of the evaluation function in the current epoch is not smaller than 0.95 of that in the previous one, we update the optimized subset with one in the current epoch and exit the iteration. Otherwise, we update the optimized subset with the first m unique features from the list of sorted feature subsets and go to the next epoch. The number of unique features m is determined as follows:

$$\begin{aligned} & \text{Max } m \\ & \text{s.t. } \frac{m!}{(i+1)!(m-i-1)!} \leq N_e \end{aligned} \quad (4)$$

where i is the index of epoch, and N_e is the maximum number of feature subsets to be evaluated in every epoch, which is a predefined constant to control the amount of computation.

2.2. Investigated Datasets

Three different kinds of datasets were investigated to test the performance of BetaDCE. Firstly, an exclusive or-like (XOR-like) simulated dataset was used to test the ability for detecting redundant features by BetaDCE. Then, BetaDCE was used for feature selection in the training set of a leukemia dataset and prediction of class labels in the independent test set. Finally, in a metabonomic dataset, we applied BetaDCE to determine the number of features, which was extracted from raw features by partial least square (PLS) and used to predict whether chemotherapy would yield enough of a positive response on patients with gastric cancer.

To intuitively demonstrate its performance in feature selection, the model was tested on an XOR-like dataset. A raw training set containing 280 samples was generated for binary classification, where each class consisted of 140 samples. Samples of the two classes were crossed in four clusters in an XOR-like style, with normal distribution for samples in each cluster, to keep it non-linear. The raw training set was described by two features, and an additional redundant feature was then added, which was repeated 100,000 times to generate multiple training sets, with each containing one redundant feature, and the raw training set.

The leukemia dataset was taken from a collection of leukemia patients reported by Golub et al. [34]. The goal of this study was to develop a systematic approach to distinguish acute lymphoblastic leukemia (ALL) from acute myeloid leukemia (AML) based on the simultaneous measure of thousands of gene expressions using DNA microarrays. The training set consisted of 27 ALL and 11 AML samples. An independent collection of 34 leukemia samples (20 ALL and 14 AML) was used for testing. Each sample was described by 7129 probes from 6817 human genes.

The metabonomic dataset used in this study was about the chemosensitivity prediction of cisplatin plus 5-fluorouracil in a human xenograft model of gastric cancer [35]. In total, 9265 raw features were generated from the HPLC/MS profile and then extracted by PLS for prediction. There were 60 patients in the dataset, where 17 were sensitive to chemotherapy and the remaining were considered as the insensitive group.

2.3. Algorithm Implementation

We developed a Python implementation of BetaDCE, which takes advantage of multiple central processing unit (CPU) cores to significantly accelerate the training process, and the open source code is now available [36].

Our server had 80 Xeon E7-8870 CPU cores (2.10 GHz) and 120 GB of memory. The training time of BetaDCE for the leukemia dataset was 149.6 min on a single CPU, and it was shortened to 4.1 min when 80 CPUs were used. As a comparison, the training time of RF + sequential floating forward selection (SFFS) for the leukemia dataset was 380.3 min.

3. Results

3.1. Generalization Ability of BetaDCE

The generalization ability of a supervised learning method is related to both bias and variance. Higher bias leads to underfit, while higher variance leads to overfit. As for BetaDCE, the expectation value E combines the information of both bias and variance and can be a criterion for the evaluation of generalization ability.

Suppose there are two potential numbers of nearest neighbors, $K1 = 4$ and $K2 = 8$, to be determined. Of the four neighbors for $K1$, three are from the positive class, and one is from the negative class. For $K2$, six are from the positive class, and two are from negative class (Figure 2A). $K1$ and $K2$ obviously have the same bias toward the positive class (both are 3:1); however, the probability density of $K2$ is sharper, which results in lower variance than that of $K1$ (Figure 2B). E is the expectation value analyzed from the probability density to combine bias and variance. As a result, E for $K2$ is bigger than that for $K1$; therefore, we should choose $K2$ as the better number of nearest neighbors for better generalization ability.

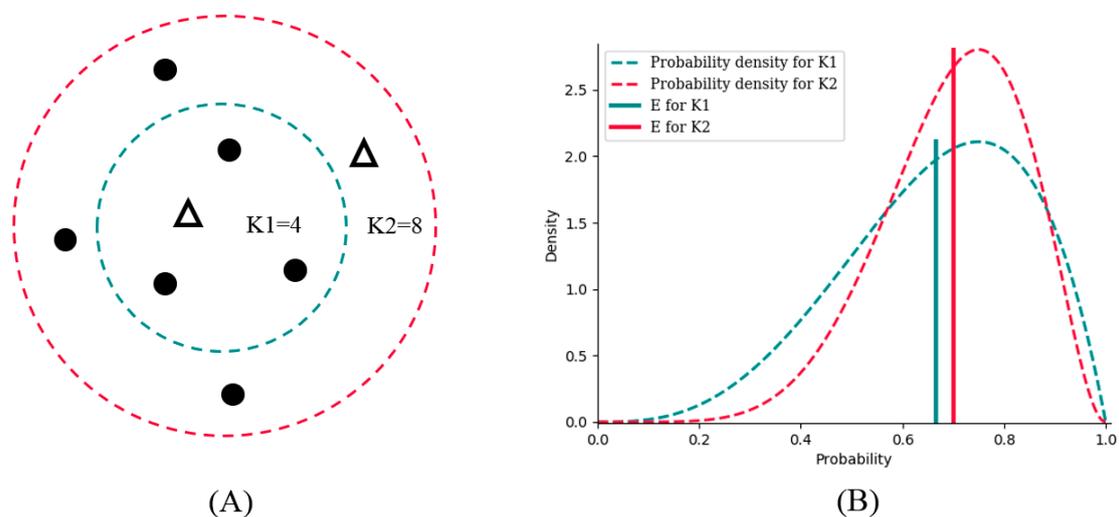


Figure 2. The generalization ability of BetaDCE. (A) An example comparing two numbers of nearest neighbors. $K1$ and $K2$ are two potential numbers of nearest neighbors, with the same bias, to be determined. (B) The probability density and E (expected value) for $K1$ and $K2$. Although the biases of $K1$ and $K2$ are the same, the probability density and E indicate that we should choose $K2$ for better generalization ability.

By analysis of probability density, BetaDCE thereby combines information of both bias and variance in the model to ensure its generalization ability.

3.2. Analysis of XOR-Like Dataset

The XOR-like dataset was used to test the performance for feature selection. Here, BetaDCE was compared to KNN and SVM in terms of detecting the redundant feature in the dataset.

To compare the performance, the algorithms were firstly evaluated on the raw training set. Then, they were compared on the datasets with redundant features by the percentage of redundant features detected. For KNN, the accuracy of leave-one-out cross-validation was used to evaluate the performance of features, and the parameter K was optimized automatically in cross-validation. For SVM, the radial basis function kernel was used as the kernel function. The penalty parameter C and kernel coefficient were optimized simultaneously in cross-validation. The accuracy of five-fold cross-validation was used to evaluate the performance of features. For BetaDCE, the objective function was directly used for evaluation without the procedure of cross-validation, which can be safely done

without data leakage since our task here is to test whether the model can identify the redundant features correctly or not, rather than to make a prediction in a test set.

Figure 3 shows the distribution of E on the raw two-dimensional data without additional redundant features during the training of BetaDCE. E clearly served as a satisfactory boundary for classification. It should be noted that the number of nearest neighbors of each sample can be determined automatically in an easy way during training; therefore, no other parameters are needed for training BetaDCE. The number of nearest neighbors varied considerably from sample to sample, with a minimal value of one and a maximal value of 68, but most of them were smaller than 10 (Figure 4).

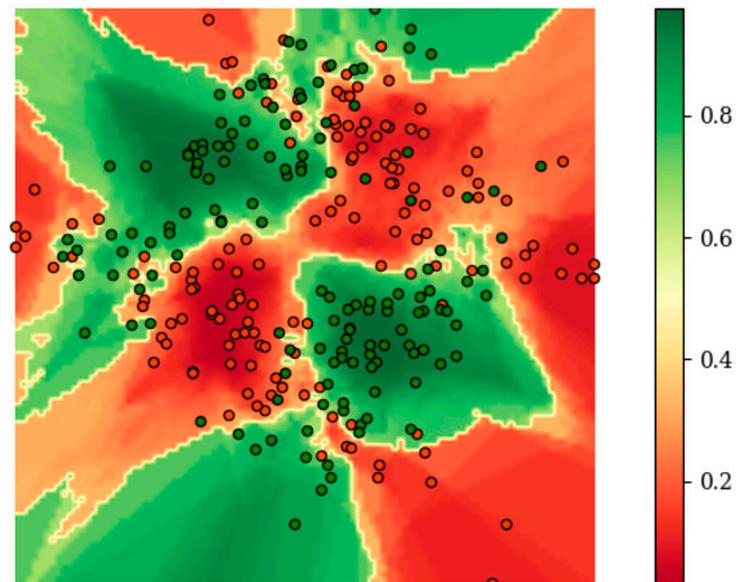


Figure 3. Distribution of E on the raw two-dimensional data of the exclusive or-like (XOR-like) dataset during training of BetaDCE.

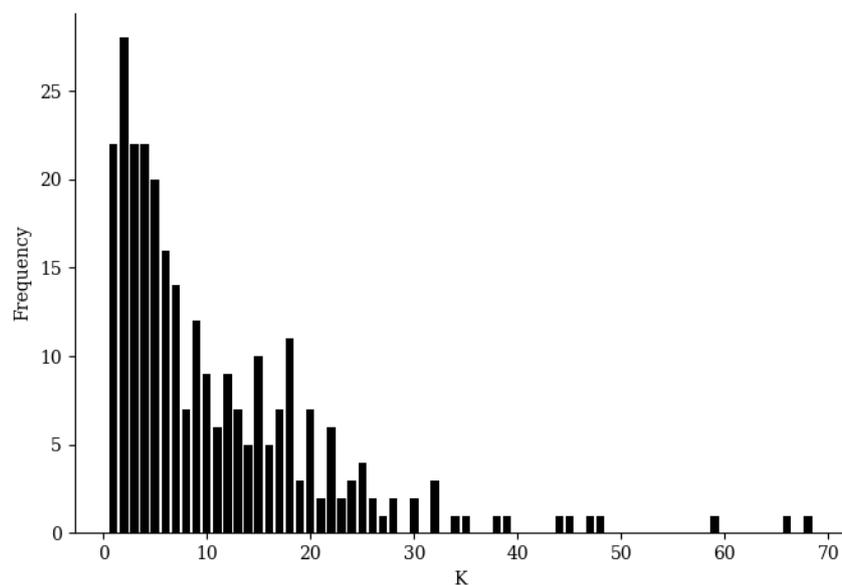


Figure 4. Statistics of number of nearest neighbors (K) for BetaDCE training on the raw XOR-like dataset without redundant features.

A permutation test was adopted to test the performance in detecting redundant features in the dataset. The redundant feature was randomly permuted 100,000 times to calculate the objective function (BetaDCE) or accuracy of cross-validation (KNN and SVM). In the test, all three methods were able to detect the raw two-dimensional features. In total, 722 and 28,809 redundant features were not detected by KNN (Figure 5B) and SVM (Figure 5C), respectively, while all redundant features were detected by BetaDCE (Figure 5A). The overall false discovery rates were 0.7%, 28.8%, and 0% for KNN, SVM, and BetaDCE, respectively (Table 1). To test the superiority of BetaDCE to KNN and SVM, we calculated the false discovery rate every 1000 permutations for all the methods and set the null hypothesis that the false discovery rate of BetaDCE was not better than that of KNN (SVM). As a result, both null hypotheses for KNN and SVM were rejected. The false discovery rate of BetaDCE was significantly smaller than that of KNN ($p = 3.8 \times 10^{-21}$) or SVM ($p = 5.7 \times 10^{-124}$), which indicated that the performance of BetaDCE for feature selection was much better than that of KNN and SVM with cross-validation.

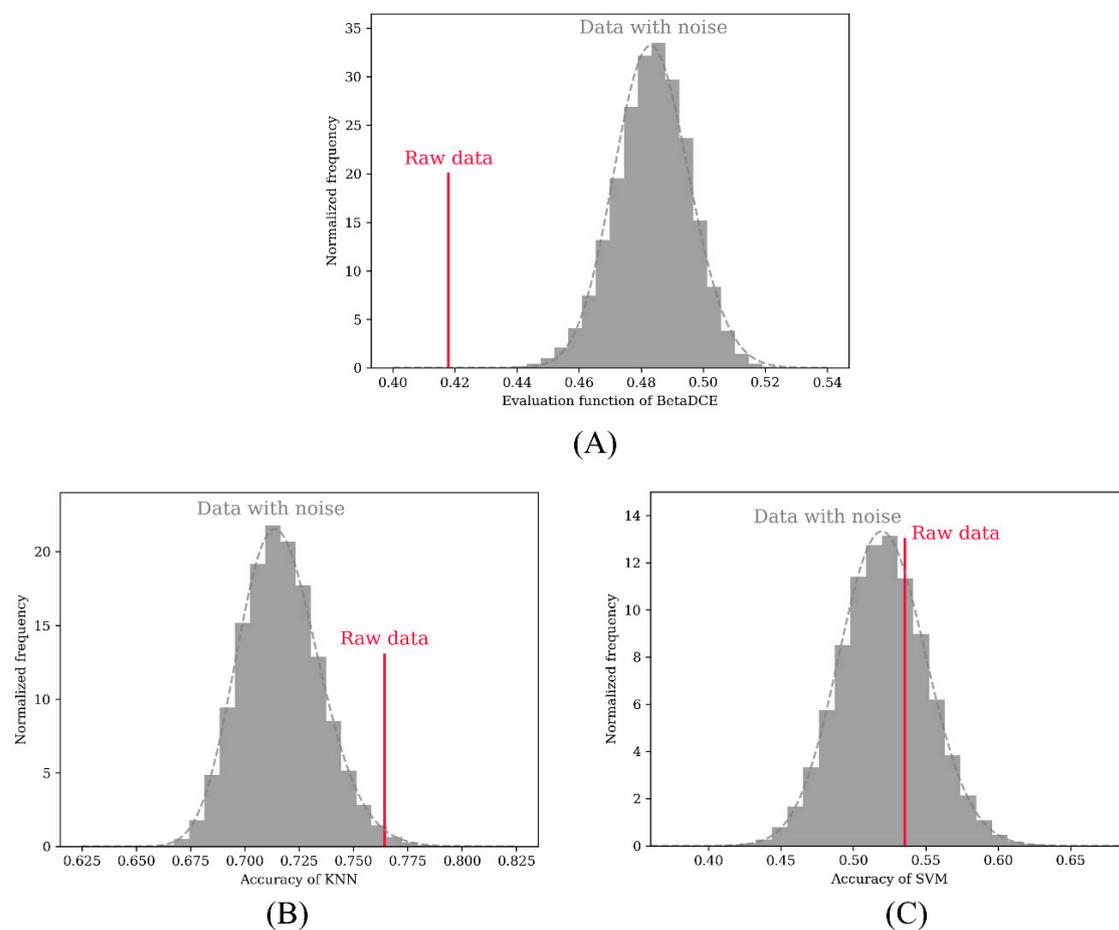


Figure 5. Permutation test for detecting redundant features on the XOR-like dataset. (A) Permuted result of BetaDCE. The statistics were based on the evaluation function. (B) Permuted result of k-nearest neighbor (KNN). The statistics were based on the accuracy of leave-one-out cross-validation. (C) Permuted result of support vector machine (SVM). The statistics was based on the accuracy of five-fold cross-validation.

Table 1. Comparison between beta distribution-based cross-entropy (BetaDCE) and other methods for detecting redundant features on an exclusive or-like (XOR-like) dataset. KNN—k-nearest neighbor; SVM—support vector machine.

Methods	Evaluation Function	False Discovery Rate (1-Specificity)
KNN	Cross-validation	0.7%
SVM	Cross-validation	28.8%
BetaDCE	Equation (3)	0%

3.3. Analysis of Leukemia Dataset

3.3.1. Predictive Result of BetaDCE

As described previously, 38 samples of leukemia were analyzed to test whether it was possible to classify them as ALL or AML based on 7129 genes using DNA microarrays. An independent test set of 34 leukemia samples was then used to evaluate the performance of the classification model trained. The performance of BetaDCE was compared with Golub’s method, as well as least absolute shrinkage and selection operator (LASSO), and random forest (RF) with sequential floating forward selection (SFFS).

In Golub’s work [34], a total of 50 genes most closely correlated with AML–ALL distinction in the training set were chosen as the set of informative genes. The class predictor was based on “weighted votes” of the informative genes. Before training of BetaDCE, all the genes were standardized to unit variance by subtracting the mean and then dividing by the standard deviation. The evaluation function decreased during the first few epochs, and the stopping criterion was reached at the fourth epoch (Figure 6). Figure 7A,B show the bias and variance trade-off for BetaDCE on the leukemia dataset. As can be seen, the variance was dominant when the number of nearest neighbors was small, and the bias became the main contribution to predictive results for big numbers of nearest neighbors. The cross-entropy by BetaDCE is a trade-off between bias and variance. The area under the curve (AUC) of BetaDCE on the test set was 0.93, which was larger than that of Golub’s method (0.83), RF + SFFS (0.81), and LASSO (0.81). The comparison between BetaDCE and other methods is shown in Table 2, which indicated that BetaDCE could provide higher predictive performance with a much smaller number of selected features.

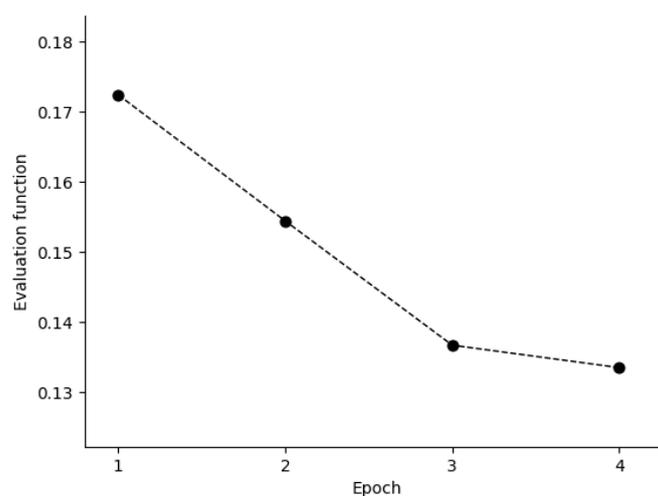


Figure 6. Relationship between evaluation function of BetaDCE and training epoch. Epoch i means that all subsets consisting of i features in the training dataset have had an opportunity to update the optimized subset.

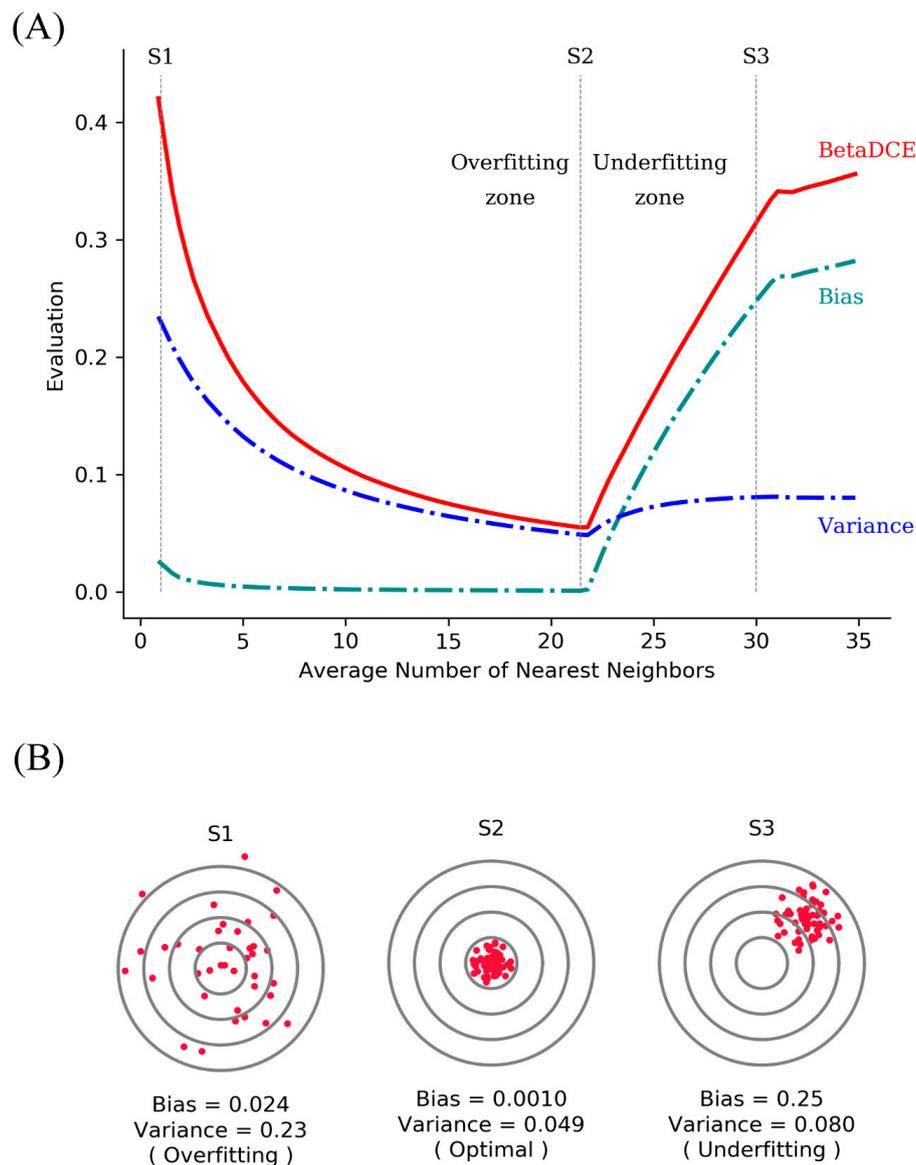


Figure 7. Bias and variance trade-off for BetaDCE on leukemia dataset. (A) Relationship among the evaluations of BetaDCE, bias, and variance. Here, this was the bias and variance trade-off when the number of nearest neighbors was optimized for the optimal feature subset. Variance was shown in the form of standard deviation so that it was comparable with bias and the value of BetaDCE. Red solid line: evaluation by BetaDCE. Blue dashed dotted line: bias. Green dashed dotted line: variance. The average numbers of nearest neighbors for S1, S2, and S3 were one, 21.4, and 30 respectively. S2 was the optimal parameter dividing the whole space into overfitting zone (left) and overfitting zone (right). (B) The bulls-eye diagrams for customized points S1, S2, and S3. Data were simulated according to the size of bias and variance.

Table 2. Comparison between BetaDCE and other methods on the leukemia dataset. RF—random forest; SFFS—sequential floating forward selection; LASSO—least absolute shrinkage and selection operator; AUC—area under the curve.

Methods	AUC on Test Set	Number of Selected Features
Reported by Golub [34]	0.83	50
RF + SFFS	0.81	3
LASSO	0.81	111
BetaDCE	0.93	4

3.3.2. Interpretation of the Features Selected by BetaDCE

One advantage of feature selection is that it eases interpretation of the system revealed by data. For the leukemia dataset, BetaDCE finally selected four genes which were crucial for the distinction of AML and ALL. The accession numbers of selected genes were M54995_at, M84526_at, X16662_at, and M22612_f_at. Detailed information about the selected genes is shown in Table 3.

Table 3. Description of genes selected by BetaDCE.

Gene Accession Number	Description of the Gene
M84526_at	DF: D component of complement (adipsin)
X16662_at	ANX8: Annexin VIII
M54995_at	PPBP: Connective tissue activation peptide III
M22612_f_at	PRSS1: Protease; serine; 1 (trypsin 1)

Furthermore, we measured the predictive power of each input feature by measuring the area under the curve (AUC) of the receive operating characteristic (ROC) curve. Figure 8 shows the AUCs of all the features in the leukemia dataset. From the figure, we can see that, among the four genes selected by BetaDCE, AUCs of M54995_at (0.95) and M84526_at (0.92) were high, whereas AUCs of X16662_at (0.50) and M22612_f_at (0.51) were at the average level.

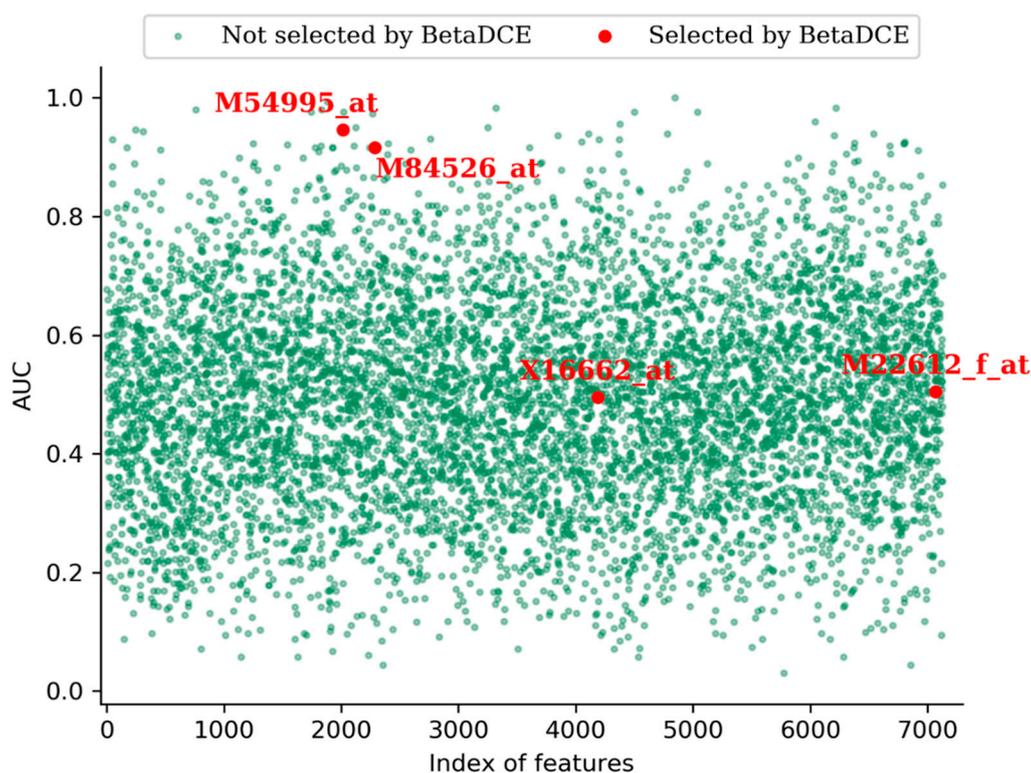


Figure 8. Predictive power of the features in the leukemia dataset. The predictive power of a feature is measured by the area under the curve (AUC). Red points are the genes selected by BetaDCE, and the corresponding gene accession numbers are shown. The order of features listed in the figure is the same as that in Golub's work.

3.4. Analysis of the Metabonomic Dataset

The metabonomic dataset was randomly split into a training set (2/3) and a test set (1/3), as described by Wang et al. [35]. The splits of the dataset were repeated 1000 times, and predictions on test sets were evaluated by ROC curves. Raw features were generated by the HPLC/MS profile and extracted by

PLS. Then, BetaDECE or cross-validation was used to determine the number of features for predictive model. As indicated in Figure 9, the average true positive rate by BetaDCE was higher than that by cross-validation for most points of false positive rate. A paired comparison by AUC indicated that the performance of feature selection by BetaDCE was significantly better than that by cross-validation ($p = 2.4 \times 10^{-112}$).

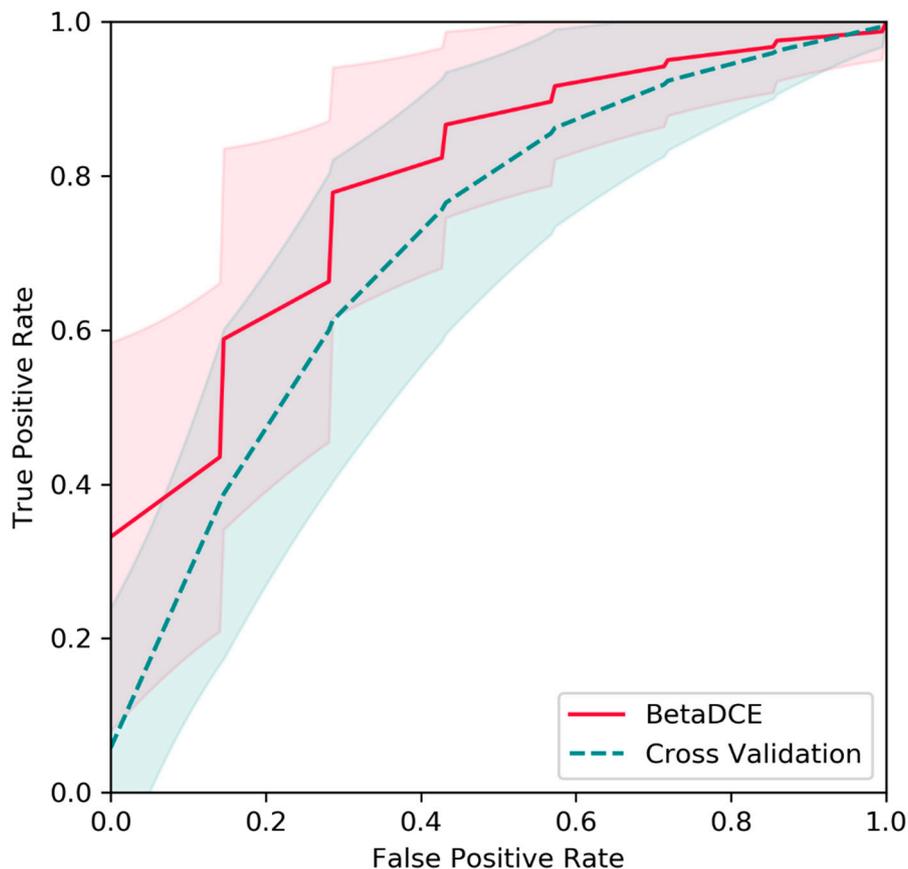


Figure 9. Comparison of BetaDCE and cross-validation for feature selection on the metabonomic dataset. The predictive results based on 1000 random splits were compared in terms of receive operating characteristic (ROC) curves. The red and blue filled areas are the standard deviations for BetaDCE and cross-validation, respectively.

4. Discussion

Cross-entropy is widely used as a loss function in machine learning, such as logistic regression and deep neural networks. It minimizes the difference between the estimated probability distribution and the true distribution from the perspective of information theory. The estimated probability distribution needed for cross-entropy is generally predicted from the training samples. However, the prediction of probability distribution from the limited number of samples could be much more difficult than the original problem. The variance may be high, especially when the number of samples is small compared to the number of features.

In view of such difficulties, BetaDCE uses beta distribution as the estimated probability distribution to calculate cross-entropy. Meanwhile, the expected value of beta distribution is analytically available; thus, estimation of cross-entropy is easy. Therefore, BetaDCE is efficient because no resampling methods such as cross-validation are needed, and the parameter K is optimized automatically during training without the requirement of additional parameters for optimization.

The limitation of BetaDCE is that, when the number of features in the investigated dataset is huge compared to number of samples, the solution might be biased from the optimal one. Under such circumstances, we should enlarge the constant N_e to increase the probability of getting to the optimal solution at the expense of greater computational effort. This limitation is essentially the problem of wrapper methods.

5. Conclusions

In summary, we proposed BetaDCE as a novel approach for feature selection. The characteristic of BetaDCE is that the cross-entropy of a predictive model is computed directly by the expected value of beta distribution, so that the generalization ability can be estimated more precisely than conventional methods such as cross-validation. We analyzed the generalization ability of BetaDCE by trade-off between bias and variance. The performance of BetaDCE was validated on several datasets. In the XOR-like dataset, the false discovery rate of BetaDCE was significantly smaller than that of other methods. For the leukemia dataset, the AUC of BetaDCE on the test set was 0.93 with only four selected features, indicating that the performance of prediction by BetaDCE can be ensured with a smaller number of features than the original method, whose AUC was 0.83 with 50 features. In the metabonomic dataset, the overall AUC of the prediction with features selected by BetaDCE was significantly larger than that by the original reported method.

Although the robustness of BetaDCE was validated in this study, its limitation is that the optimal solution cannot be promised if there are a huge number of features. In future work, one practical way of ensuring the optimal solution is to implement the algorithm on a graphics processing unit (GPU) for efficient parallelization. Another way is to organize the algorithm of BetaDCE as a constrained mathematical program, so that it can be solved as an embedded method.

Author Contributions: Conceptualization, W.D.; methodology, W.D.; software, W.D.; validation, W.D.; formal analysis, W.D.; writing—original draft preparation, W.D.; writing—review and editing, D.G.; supervision, D.G.; project administration, D.G.; funding acquisition, D.G.

Funding: This research was funded by State Key Laboratory of Agrobiotechnology, grant number 8300052. And the APC was funded by a direct grant from the Chinese University of Hong Kong.

Acknowledgments: We thank Mofan Zhou from Griffith University for his constructive remarks. We thank the reviewers for their insightful comments and suggestions.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Bellman, R.E. *Dynamic Programming*; Princeton University Press: Princeton, NJ, USA, 1957.
2. Yamada, K.; Takayasu, H.; Takayasu, M. Estimation of economic indicator announced by government from social big data. *Entropy* **2019**, *20*, 852. [[CrossRef](#)]
3. Rauber, T.W.; de Assis Boldt, F.; Varejão, F.M. Heterogeneous feature models and feature selection applied to bearing fault diagnosis. *IEEE Trans. Ind. Electron.* **2015**, *62*, 637–646. [[CrossRef](#)]
4. Hira, Z.M.; Gillies, D.F. A review of feature selection and feature extraction methods applied on microarray data. *Adv Bioinform.* **2015**, *2015*, 198363. [[CrossRef](#)]
5. Guyon, I.; Elisseeff, A. An introduction to variable and feature selection. *JMLR* **2003**, *3*, 1157–1182.
6. Yang, Y.M.; Pedersen, J.O. A comparative study on feature selection in text categorization. In Proceedings of the 14th International Conference on Machine Learning, Nashville, TN, USA, 14–18 June 1997; pp. 412–420.
7. Bø, T.; Jonassen, I. New feature subset selection procedures for classification of expression profiles. *Genome Biol.* **2002**, *3*, research0017.1–research0017.11. [[CrossRef](#)]
8. Urbanowicz, R.J.; Meeker, M.; Cava, W.L. Relief-based feature selection: Introduction and review. *J. Biomed. Inform.* **2018**, *85*, 189–203. [[CrossRef](#)]
9. Ambrose, C.; McLachlan, G.J. Selection bias in gene extraction on the basis of microarray gene-expression data. *Proc. Natl. Acad. Sci. USA* **2002**, *99*, 6562–6566. [[CrossRef](#)]

10. Perez, M.; Marwala, T. Microarray data feature selection using hybrid genetic algorithm simulated annealing. In Proceedings of the IEEE 27th Convention of Electrical and Electronics, Eilat, Israel, 14–17 November 2012; pp. 1–5.
11. Dunn, J.C. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *J. Cybern.* **1973**, *3*, 32–57. [[CrossRef](#)]
12. Bezdek, J.C. *Pattern Recognition with Fuzzy Objective Function Algorithms*; Kluwer Academic Publishers: Norwell, MA, USA, 1981.
13. Bach, F.R. Bolasso: Model consistent Lasso estimation through the bootstrap. In Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 5–9 July 2008; pp. 33–40.
14. Zare, H.; Haffari, G.; Gupta, A.; Brinkman, R.R. Scoring relevancy of features based on combinatorial analysis of Lasso with application to lymphoma diagnosis. *BMC Genom.* **2013**, *14*, 1–14. [[CrossRef](#)]
15. Liu, Q.; Sung, A.H.; Chen, Z.; Liu, J.; Huang, X.; Deng, Y. Feature selection and classification of MAQC-II breast cancer and multiple myeloma microarray gene expression data. *PLoS ONE* **2009**, *4*, e8250. [[CrossRef](#)]
16. Tang, E.K.; Suganthan, P.N.; Yao, X. Feature selection for microarray data using least squares SVM and particle swarm optimization. In Proceedings of the IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, La Jolla, CA, USA, 14–15 November 2005; pp. 9–16.
17. Tang, Y.; Zhang, Y.Q.; Huang, Z. Development of two-stage SVM-RFE gene selection strategy for microarray expression data analysis. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2007**, *4*, 365–381. [[CrossRef](#)]
18. Liu, H.; Motoda, H. *Feature Selection for Knowledge Discovery and Data Mining*; Springer Science & Business Media: Berlin, Germany, 2012.
19. Ustunkar, G.; Ozogur-Akyuz, S.; Weber, G.W.; Friedrich, C.M.; Son, Y.A. Selection of representative SNP sets for genome-wide association studies: A metaheuristic approach. *Optim. Lett.* **2012**, *6*, 1207–1218. [[CrossRef](#)]
20. Zhang, Y.D.; Dong, Z.C.; Phillips, P.; Wang, S.; Ji, G.; Yang, J.; Yuan, T.F. Detection of subjects and brain regions related to Alzheimer’s disease using 3D MRI scans based on eigenbrain and machine learning. *Front. Comput. Neurosci.* **2015**, *9*, 66–80. [[CrossRef](#)]
21. Akaike, H. Prediction and entropy. In *A Celebration of Statistics*; Atkinson, A.C., Fienberg, S.E., Eds.; Springer: Berlin/Heidelberg, Germany, 1985; pp. 1–24.
22. Burnham, K.P.; Anderson, D.R. *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2002.
23. Rissanen, J. Modeling by shortest data description. *Automatica* **1978**, *14*, 465–658. [[CrossRef](#)]
24. Stone, M. An asymptotic equivalence of choice of model by cross-validation and Akaike’s criterion. *J. R. Stat. Soc. Ser. B* **1977**, *39*, 44–47. [[CrossRef](#)]
25. Fang, Y. Asymptotic equivalence between cross-validations and Akaike Information Criteria in mixed-effects models. *J. Data Sci.* **2011**, *9*, 15–21.
26. Anderssen, E.; Dyrstad, K.; Frank Westad, F.; Martens, H. Reducing over-optimism in variable selection by cross-model validation. *Chemometr. Intell. Lab.* **2006**, *84*, 69–74. [[CrossRef](#)]
27. Efron, B. Estimating the error rate of a prediction rule: Improvement on cross-validation. *J. Am. Stat. Assoc.* **1983**, *78*, 316–331. [[CrossRef](#)]
28. Mafarja, M.; Mirjalili, S. Whale optimization approaches for wrapper feature selection. *Appl. Soft Comput.* **2018**, *62*, 441–453. [[CrossRef](#)]
29. Mafarja, M.; Aljarah, I.; Heidari, A.A.; Hammouri, A.I.; Faris, H.; Al-Zoubi, A.M.; Mirjalili, S. Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems. *Knowl. Based Syst.* **2018**, *145*, 25–45. [[CrossRef](#)]
30. Mafarjaa, M.; Aljarah, I.; Heidari, A.A.; Faris, H.; Fournier-Viger, P.; Li, X.; Mirjalili, S. Binary dragonfly optimization for feature selection using time-varying transfer functions. *Knowl. Based Syst.* **2018**, *161*, 185–204. [[CrossRef](#)]
31. Hancer, M.; Xue, B.; Zhang, M.; Karaboga, M.; Akay, B. Pareto front feature selection based on artificial bee colony optimization. *Inform. Sci.* **2018**, *422*, 462–479. [[CrossRef](#)]
32. Mafarja, M.M.; Mirjalili, S. Hybrid whale optimization with simulated annealing for feature selection. *Neurocomputing* **2017**, *260*, 302–312. [[CrossRef](#)]
33. Faris, H.; Mafarja, M.M.; Heidari, A.A.; Aljarah, I.; Al-Zoubi, A.M.; Mirjalili, S.; Fujita, H. An efficient binary Salp Swarm Algorithm with crossover scheme for feature selection problems. *Knowl. Based Syst.* **2018**, *154*, 43–67. [[CrossRef](#)]

34. Golub, T.R.; Slonim, D.K.; Tamayo, P.; Huard, C.; Gaasenbeek, M.; Mesirov, J.P.; Coller, H.; Loh, M.L.; Downing, J.R.; Caligiuri, M.A.; et al. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science* **1999**, *286*, 531–537. [[CrossRef](#)]
35. Wang, X.; Yan, S.K.; Dai, W.X.; Liu, X.R.; Zhang, W.D.; Wang, J.J. A metabonomic approach to chemosensitivity prediction of cisplatin plus 5-fluorouracil in a human xenograft model of gastric cancer. *Int. J. Cancer* **2010**, *127*, 2841–2850. [[CrossRef](#)]
36. Dai, W.; Guo, D. The Open Source Code. Available online: github.com/mlalgorithm/betadce (accessed on 15 June 2019).



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).