Journal of Computational Biophysics and Chemistry

Research

Cite This: *J. Comput. Biophys. Chem.* **2024**, *23* (10), 1339–1358

Graph-Based Bidirectional Transformer Decision Threshold Adjustment Algorithm for Class-Imbalanced Molecular Data

Nicole Hayes[*], Ekaterina Merkurjev[*,†,¶] and Guo-Wei Wei[*,‡,§]

[*]*Department of Mathematics, Michigan State University, MI 48824, USA*
[†]*Department of Computational Mathematics, Science and Engineering, Michigan State University, MI 48824, USA*
[‡]*Department of Electrical and Computer Engineering, Michigan State University, MI 48824, USA*
[§]*Department of Biochemistry and Molecular Biology, Michigan State University, MI 48824, USA*
[¶]Corresponding author. E-mail: *merkurje@msu.edu*

ABSTRACT: Data sets with imbalanced class sizes, where one class size is much smaller than that of others, occur exceedingly often in many applications, including those with biological foundations, such as disease diagnosis and drug discovery. Therefore, it is extremely important to be able to identify data elements of classes of various sizes, as a failure to do so can result in heavy costs. Nonetheless, many data classification procedures do not perform well on imbalanced data sets as they often fail to detect elements belonging to underrepresented classes. In this work, we propose the BTDT-MBO algorithm, incorporating Merriman–Bence–Osher (MBO) approaches and a bidirectional transformer, as well as distance correlation and decision threshold adjustments, for data classification tasks on highly imbalanced molecular data sets, where the sizes of the classes vary greatly. The proposed technique not only integrates adjustments in the classification threshold for the MBO algorithm in order to help deal with the class imbalance, but also uses a bidirectional transformer procedure based on an attention mechanism for self-supervised learning. In addition, the model implements distance correlation as a weight function for the similarity graph-based framework on which the adjusted MBO algorithm operates. The proposed method is validated using six molecular data sets and compared to other related techniques. The computational experiments show that the proposed technique is superior to competing approaches even in the case of a high class imbalance ratio.

KEYWORDS: Imbalanced data; molecular data; transformer; graph-based; data classification.

1. INTRODUCTION

In data classification, class imbalanced data is distinguished by an underrepresentation of one class (the "minority" class) in the data compared to the other class or classes (the "majority" class(es)). As a result, data classification algorithms trained on imbalanced data tend to become biased with regard to the majority class(es) and often misclassify data points from the minority group, which is frequently the class of most interest.[1,2] This scenario is quite common in classification tasks, and it can be especially problematic if the minority class represents an important or critical outcome, such as in drug discovery, disease diagnosis, fraud detection, energy management, and other applications involving rare events.[2] For example, drug discovery is a very exacting problem in science. In drug screening, the number of inactive molecules is typically thousands of times larger than that of active molecules. A successful drug must be highly potent, be free of serious side effects or off-target binding, and exhibit minimal or no toxicity.[1] As a result, the majority of drug candidates are either inactive or unqualified, leading to highly imbalanced data sets.[3,4] Specifically, imbalanced data occur during virtual screening: identifying potential drug candidates from large compound libraries. Effective handling of imbalanced data can improve the identification of active compounds.[5–8] Another situation in which we deal with imbalanced data is quantitative structure–activity relationship modeling, which predicts a compound's activity using

Received: 10 June 2024
Accepted: 16 August 2024
Published: 19 September 2024

DOI: 10.1142/S2737416524500479
J. Comput. Biophys. Chem. **2024**, *23* (10), 1339–1358

the chemical composition.[9,10] Balancing the data can lead to more reliable predictions.[11]

Therefore, addressing imbalanced data is crucial for effective machine learning in many applications such as drug design and discovery. By employing a fusion of algorithm-level and data-level strategies, and focusing on appropriate evaluation metrics, researchers can build models that better identify rare but valuable compounds, leading to more successful outcomes in drug discovery and development.

Techniques to deal with imbalanced data can be categorized into[2,12–14] data pre-processing or resampling methods,[15,16] model-level techniques,[2,17] ensemble algorithms[18] and cost-sensitive methods.[19] Strategies in the first group include resampling methods, which attempt to make the training set's class distribution more balanced using sampling techniques, and feature selection methods, which are carried out in the feature space rather than the sample space.[2] Examples of resampling procedures include oversampling,[20,21] undersampling[22,23] and hybrid-sampling algorithms.[15,24,25] Respectively, these procedures attempt to rebalance the sample space by creating new samples from the minority group, discarding representatives from the majority group, or a combination of both. Methods belonging to the second group, the model-level techniques, attempt to make adjustments to steps of the algorithm itself so that there is less bias towards classes of larger size. Ensemble algorithms use the output of multiple classifiers to make predictions, which can improve the performance of individual classifiers.[2,26] Lastly, cost-sensitive procedures instead assign a larger misclassification cost to classes of smaller size.[19]

Some cost-sensitive approaches to imbalanced data classification involve thresholding, which is applied in a post-processing step.[27] In particular, thresholding algorithms are cost-sensitive techniques which assign a probability to each of the test elements. For a threshold of 0.5, which is the default probability threshold value in most classifiers,[28] any test element with a probability of at least 0.5 will be assigned to one class; otherwise, it will be assigned to another class. In the case of highly imbalanced data classification, the threshold of 0.5 may not be optimal; thus, threshold algorithms attempt to find a more optimal threshold boundary for the machine learning task. Usually, a balanced accuracy metric is used; some examples include the ROC curve (receiver operating characteristic curve),[29] the G-Mean,[30] F1-score,[28] the Matthews correlation coefficient[31] and the balanced accuracy.[5] In these cases, the goal is to maximize the balanced accuracy metric. Overall, one approach for image classification and feature extraction is detailed in Ref. 32, which proposes a convolution-based efficient transformer image feature extraction network. Moreover, Wang *et al.*[33] introduce a micro-directional propagation model using deep learning, and Zhu[34] focus on an adaptive agent decision model derived using autonomous learning deep reinforcement learning. In addition, Yin *et al.*[35] detail feature extraction modules, while Peng *et al.*[36] present a systematic review of picture fuzzy decision-making methods. Applications to sideband harmonics, internet market design, large-scale sliding puzzles are found in Refs. 37, 38 and 39.

However, most of the cost-sensitive techniques for imbalanced data classification that search for the optimal threshold can be computationally extensive because retraining of the classifier is often required. Inspired by the work of Esposito *et al.*,[1] in this paper, we detail an algorithm for imbalanced data classification which does not require any retraining in its procedure. In order to motivate the proposed procedure, we turn to the authors' previous work,[40] which demonstrated the success of their BT-MBO method for molecular data classification with very small labeled sets; refer Ref. 41 for a survey of machine learning techniques for data with limited labeled elements in molecular science. The BT-MBO algorithm, which admits two-dimensional molecular data in a special format called the simplified molecular input line entry specification (SMILES)[42] format, is composed of a bidirectional transformer[43] and a graph-based and modified Merriman–Bence–Osher (MBO) algorithm[44,45] in series, utilizing the transformer model to generate molecular fingerprints which then serve as features for the MBO algorithm. The MBO method used in the BT-MBO model[40] generates a probability distribution over all classes for each data point, and it subsequently classifies each data element by choosing the class for which that element has the highest probability. However, in the case of imbalanced data, this procedure may cause the technique to miscategorize data elements in the minority group, since the probability distribution may be biased toward the majority classes. Thus, in this work, we propose the BTDT-MBO method (bidirectional transformer with MBO techniques using a varying decision threshold), which is a novel adaptation of the BT-MBO technique for use on highly imbalanced molecular data.

In particular, the proposed BTDT-MBO method incorporates a decision threshold adjustment in the MBO algorithm, which utilizes the probability distribution generated by the algorithm for each data point. Instead of following the typical MBO classification

DOI: 10.1142/S2737416524500479

procedure, the BTDT-MBO algorithm tests a set of thresholds and chooses the threshold that produces the highest ROC-AUC score (area under the ROC curve) for the unlabeled points. The new thresholding step introduced in the BTDT-MBO method does not require any additional iterations of the MBO algorithm; rather, at the end of each iteration, we repeat the new thresholding step for each threshold. We refer to the modified MBO algorithm with decision threshold adjustment as the DT-MBO method. Another adaptation introduced in the BTDT-MBO method is the inclusion of the distance correlation[46,47] as a weight function in the DT-MBO algorithm. The default weight function used in our BTDT-MBO method is a Gaussian kernel; in addition to experiments using a Gaussian weight function, we also carried out experiments using a distance correlation weight function.

Given our previous BT-MBO method's predictive success for the challenging scenario of scarcely labeled data,[40] we focus on the difficult task of data classification of highly imbalanced molecular data sets in this paper. Specifically, all six data sets used for benchmarking the proposed BTDT-MBO algorithm have an imbalance ratio (IR) of 16.5 or greater. We compare our results using BTDT-MBO to those of the GHOST procedure[1] applied to four common classification algorithms: random forest, gradient boosting decision trees, extreme gradient boosting and logistic regression. Specifically, the GHOST technique[1] is an automated procedure that can be used in any data classification procedure and that adjusts the decision threshold of a classifier using the training set's prediction probabilities (i.e., without needing to retrain the classifier); it achieved success in classifying imbalanced molecular data, demonstrated on data with varying levels of class imbalance. Detailed results on the six highly imbalanced data sets, as well as a discussion, are given in Sec. 3. The computational experiments indicate that our BTDT-MBO method obtains superior results to those of the comparison techniques.

Contributions

We present the main contributions of this work:

- We present a new technique, the BTDT-MBO method, for molecular data classification, especially designed to be very useful even in cases of highly imbalanced data sets. This procedure incorporates Merriman–Bence–Osher techniques, as well as bidirectional transformers, distance correlation and adjustments in the classification threshold.
- The proposed BTDT-MBO algorithm is able to perform extremely well even in the cases of molecular data that is highly imbalanced and where the class sizes vary significantly. This is extremely important, since highly imbalanced data sets are often found in practice.
- The proposed method utilizes several techniques to achieve good performance when using data with high class imbalance ratios, including adjustments in the classification threshold.
- The proposed technique is evaluated using experiments on six data sets. It is shown that the BTDT-MBO method performs more accurately than the comparison methods in almost all cases. The ROC-AUC score is used as a metric.

2. METHODS

In this section, we derive the proposed techniques. In particular, in Sec. 2.1, we outline the details of the proposed BTDT-MBO method (bidirectional transformer with MBO techniques[44,45] using a varying decision threshold), which adapts some of the techniques of our previous work.[40] In Sec. 2.2, we further discuss the distance correlation[46,47] as a potential similarity metric for use in the BTDT-MBO method.

2.1. BTDT-MBO algorithm

The authors' previous work[40] demonstrated the success of their proposed BT-MBO method in molecular classification problems with small sets of labeled nodes, with the BT-MBO procedure outperforming state-of-the-art techniques on some benchmark data sets when only 1% of all data elements are viewed as labeled. This work proposes a method, called BTDT-MBO, that adapts some of the procedures of the BT-MBO procedure for the data classification task on molecular data sets where the classes of the data set vary vastly in size.

Inspired in part by the success of Esposito *et al.*[1] in adjusting the classification threshold for machine learning classifiers to make predictions about imbalanced molecular data, the proposed BTDT-MBO method utilizes the aspects of the BT-MBO algorithm from Ref. 40 along with a decision threshold adjustment at each iteration of the method. We denote the adjusted MBO method with varied decision threshold by DT-MBO. The proposed BTDT-MBO algorithm consists of a bidirectional transformer (created by Chen *et al.*[43]) and a DT-MBO algorithm in series, where the fingerprints (BT-FPs) generated by the

DOI: 10.1142/S2737416524500479
J. Comput. Biophys. Chem. **2024**, *23* (10), 1339–1358

transformer serve as input features to the DT-MBO algorithm. Figure 1 illustrates the steps of the BTDT-MBO algorithm. The BTDT-MBO algorithm is further outlined in Algorithm 2, and details of the DT-MBO method are given in Algorithm 1.

2.1.1. *MBO scheme with decision threshold adjustment*

Here, we establish the similarity graph-based structure our BTDT-MBO scheme uses, which adapts techniques outlined in the literature.[44,48,49] First, consider a graph involving the vertex set V representing the data set nodes. We can then define a weight function $w: V\times V\to\mathbb{R}$, whose value represents the degree of similarity between two vertices of the graph. Specifically, $w(i, j)$ measures the degree of similarity of data elements i and j of the data set. While there are many choices for the weight function, it should be constructed so that the value of $w(i, j)$ corresponds to the extent of resemblance or closeness of the data elements i and j. A popular weight function is the Gaussian weight function:

$$w(i,j)=\exp\left(-\frac{d(i,j)^2}{\sigma^2}\right), \tag{1}$$

where $d(i, j)$ represents the distance between vertices i and j (computed using some prespecified metric such as the Euclidean distance), and $\sigma>0$.[44,49]

In this paper, we utilize the Gaussian weight function for some of our BTDT-MBO experiments, and for other experiments, we use the distance correlation[46] as a weight function. The distance correlation is defined and further discussed in Sec. 2.2.

Given a weight function w, let the weight matrix $\mathbf{W}$ be the matrix $\mathbf{W}_{ij}=w(i, j)$. Then let the degree of a vertex $i\in V$ be defined as $d_i=\sum_{j\in V} w(i, j)$. If we let $\mathbf{D}$ be a diagonal matrix containing diagonal elements $\{d_i\}$, then we can define the graph Laplacian as follows:

$$\mathbf{L}=\mathbf{D}-\mathbf{W}. \tag{2}$$

Occasionally, the graph Laplacian is normalized so that the diagonal entries of the normalized Laplacian are unit values. Of course, the non-diagonal entries are scaled accordingly.

For outlining the Merriman–Bence–Osher (MBO) framework, formerly introduced in Ref. 45 to generate approximations to mean curvature flow and then applied using a graph-based setting to various applications such as hyperspectral imaging in works such as Refs. 44, 48, 50, 51 and 52, we consider a general data classification problem with m classes. Overall, the MBO algorithm ultimately aims to compute a label matrix $\mathbf{U}=(\mathbf{u}_1,...,\mathbf{u}_N)^T$; a row $\mathbf{u}_i\in\mathbb{R}^m$ of the label matrix contains the probability values of the data element i belonging to each of the m classes. For instance, the first element of $\mathbf{u}_i$ represents the probability that node i is a member of the first class, the second element of $\mathbf{u}_i$ represents the probability that node i is a member of the second class, and so on. Therefore, each vector $\mathbf{u}_i$ can be viewed as a component of the mth Gibbs simplex:

$$\sum^m := \left\{(y_1,\ldots,y_m)\in[0,1]^m \text{ such that} \sum_{k=1}^m y_k=1\right\}. \tag{3}$$

The vertices of mth Gibbs simplex are unit vectors $\{\mathbf{e}_i\}$, where the probability of belonging to class i is 1, while the probability of belonging to any other class is 0. Thus, the vertices $\{\mathbf{e}_i\}$ correspond to data elements that belong exclusively to each of the m classes. In our data classification problem with labeled and unlabeled data elements, the labeled data elements are assigned to their corresponding vertices of the Gibbs simplex.

The process by which the MBO algorithm generates the optimal label matrix $\mathbf{U}$ relies on energy minimization, a time-splitting scheme, and transference to a graph-based setting as described above. A general graph-based machine learning algorithm for data classification can be formulated as minimizing the energy consisting of the sum of a regularization term that incorporates weights in addition to a fidelity part that integrates the labeled nodes.[49]

As Garcia and coauthors demonstrated in Ref. 44, one successful choice for the regularization term R is the Ginzburg–Landau functional,[53,54] often utilized in image processing tasks and associated applications due to its relation to the total variation functional.[48,55] For the fidelity term, one can choose an L_2 fit to all labeled nodes. In the continuous setting, L_2 gradient descent can then be applied to minimize the energy. This results in a modified Allen–Cahn equation containing an additional forcing part. Next, a time-splitting scheme produces a technique that alternates between two steps: a diffusion step, which uses the heat equation with an addition term, and a thresholding step. Finally, one can use the techniques outlined in Refs. 44, 48 and 56 to transfer the procedure to a graph-based setting, in which the thresholding step is replaced by projection to the Gibbs simplex (3), followed by displacing to the nearest vertex in the mth Gibbs simplex. Additionally, to adapt to the graph setting, one can

DOI: 10.1142/S2737416524500479
J. Comput. Biophys. Chem. **2024**,*23*(10),1339–1358

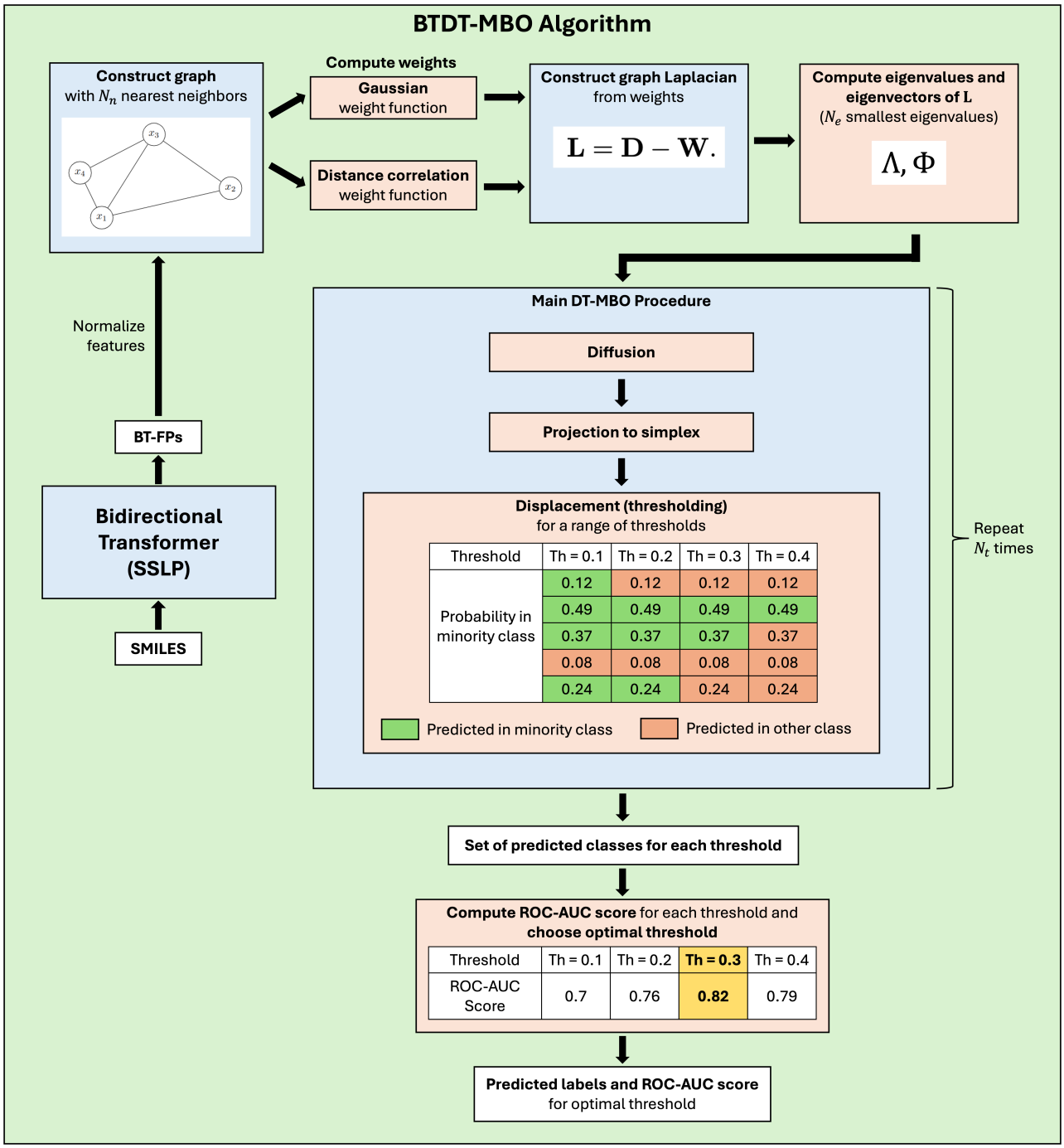

Fig. 1. (Color online) Flowchart illustrating the BTDT-MBO method.

replace the Laplace operator by the graph Laplacian (2) or some normalization of the graph Laplacian.

Thus, each iteration of the MBO algorithm ends with the aforementioned displacement step, where each row of the label matrix is displaced to its closest vertex in the simplex. In the algorithm, the closest vertex is determined by the class for which the data element corresponding to that row has the highest probability. In this paper, we propose that this procedure can instead be formulated using a decision threshold, which utilizes the probability that a data element belongs to the minority class. Given a decision threshold and a row of the label matrix, if the probability corresponding to the minority class is greater than that threshold, that row will be displaced to the vertex for the minority class. Otherwise, the typical displacement

DOI: 10.1142/S2737416524500479
J. Comput. Biophys. Chem. **2024**, *23* (10), 1339–1358

Algorithm 1. DT-MBO Algorithm (using techniques from Refs. 49 and 56)

Require: Labeled set $\mathscr{L}=\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, with fingerprint $\mathbf{x}_i$ and label y_i, N= size of data set, N_n=number of nearest neighbors, N_e=number of eigenvectors, dt, C, N_t=maximum number of iterations, N_p=number of labeled elements, N_l=number of labeled collections, N_s, m=number of classes, τ_{low}=lowest threshold, τ_{high}=highest threshold, $k_{\min}$=index of minority group/cluster.

Ensure: Average optimal ROC-AUC score over the N_l labeled sets.

1. Compute the N_n-nearest neighbor graph, unless one is using the Nyström technique[61–63] for large data. In the latter case, go to Step 3.
2. Construct the graph Laplacian **L**.
3. Calculate the smallest N_e eigenvalues Λ and the corresponding eigenvectors Φ of **L**. One can also approximate them using the Nyström method[61–63] in case of large data sets.
4. The main procedure of the proposed method:

for $l=1\rightarrow N_l$ **do**
Let the $N\times 1$ vector Γ vector be defined so that $\Gamma_j=1$ for labeled points and 0 for unlabeled points.
for $i=1\rightarrow N$ **do**
Initialize each value of a row of **U** to a random number in [0,1], except for rows corresponding to labeled nodes, in which case, set $\mathbf{u}_i^0 \leftarrow e_k$, where k is the true class.
$\mathbf{u}_i^0 \leftarrow projectToSimplex(\mathbf{u}_i^0)$. Here, $\mathbf{u}_i$ is the ith row of the matrix $\mathbf{U}^0$.
end for
$n_\tau \leftarrow (\tau_{\text{high}}-\tau_{\text{low}})/0.05+1$, $\mathbf{U}_\tau \leftarrow \mathbf{0}$, an $N\times m\times n_\tau$ array of zeros.
$\mathbf{A}\leftarrow \Phi'\mathbf{U}^0$, $\mathbf{B}\leftarrow \mathbf{0}$, $\mathbf{E}\leftarrow 1+(dt/N_s)\Lambda$
for $n=1\rightarrow N_t$ **do**
for $k=1\rightarrow N_s$ **do**
for $j=1\rightarrow m$ **do**
$\mathbf{A}_j \leftarrow (\mathbf{A}_j-(dt/N_s)\mathbf{B}_j)./\mathbf{E}$
end for
$\mathbf{U}^n \leftarrow \Phi\mathbf{A}$
$\mathbf{B}\leftarrow \mathbf{C}(\Phi'(\Gamma\cdot(\mathbf{U}^n-\mathbf{U}^0)))$, where row-wise multiplication is performed.
end for
$\mathbf{U}^{n+1}\leftarrow projectToSimplex(\mathbf{U}^n)$
for $j=1\rightarrow n_\tau$ **do**
$\tau_j=\tau_{\text{low}}+0.05(j-1)$
for $i=1\rightarrow N$ **do**
if $\mathbf{u}_{ik_{\min}}^{n+1}\geq\tau_j$ **then**
$\mathbf{u}_i^{n+1}\leftarrow \mathbf{e}_{k_{\min}}$, where $\mathbf{e}_{k_{\min}}$ is the simplex vertex corresponding to the minority class.
else
$\mathbf{u}_i^{n+1}\leftarrow \mathbf{e}_k$, where $\mathbf{e}_k$ is the closest simplex vertex to $\mathbf{u}_i^{n+1}$.
end if
end for
$\mathbf{U}_{\tau j}\leftarrow \mathbf{U}^{n+1}$, where $\mathbf{U}_{\tau j}$ is the jth $N\times m$ slice of $\mathbf{U}_\tau$.
end for
end for
Compute ROC-AUC score for each threshold, and choose the threshold that yields the highest score.
end for

5. Compute the average of the N_l Optimal ROC-AUC scores.

procedure is followed, with the row being displaced to the vertex for which it has the highest probability.

To formulate our decision threshold technique, we are inspired by Esposito *et al.*,[1] which demonstrated success in classifying imbalanced molecular data by developing a procedure that adjusts and optimizes the decision threshold of machine learning classifiers. For binary data, the default decision threshold is usually 0.5, where the model assigns a sample to the minority class if its probability of belonging to the minority class is greater than 0.5 (this is identical to assigning the sample to the class for which it has the highest probability). However, machine learning classifiers are likely to misclassify minority points since the probability

Algorithm 2. BTDT-MBO Method (using related procedures as in Ref. 40)

Require: Unlabeled set $\mathscr{U}=\{s_i\}_{i=1}^N$, where s_i is a molecular compound's SMILES string, a chosen pretrained model from the three described in Sec. 2.1.2, a dictionary which designates an integer number to each SMILES element.

Ensure: $\mathscr{F}_{\text{BT}}=\{\mathbf{x}_i\}_{i=1}^N$, a set of normalized fingerprints.

1. Binarize the data consisting of SMILES strings.
2. Loading step: load the chosen model (which is pre-trained).
3. Extracting step: Extract the hidden information from $\mathscr{U}$ via the chosen pretrained model:

Initialize the empty dictionary D_{hidden} (of length N) containing hidden features.

for $i=1\rightarrow N$ **do**

Compute tensor $\mathbf{t}_i$ (of the same dimension as s_i), containing the dictionary values of the characters in the analogous locations in s_i.

Let $\mathbf{t}_i$ be an input to the bidirectional encoder transformer from the chosen pretrained model. Obtain the inner state from the last hidden layer $\mathbf{T}_i$, whose dimensions are (l_i, 1, 512) (where l_i is the length of $\mathbf{t}_i$).

Reshape $\mathbf{T}_i$ into a tensor $\mathbf{V}_i$ with the following dimensions: (l_i, 512).

Set $\mathbf{V}_i$ as ith element of the matrix D_{hidden}.

end for

4. Generating step: Produce the fingerprints from the hidden information:

Initialize the list of fingerprints $\mathscr{F}_{\text{BT}}$ (of length N).

for $i=1\rightarrow N$ **do**

Compute the first row $\mathbf{x}_i$ from the ith element of the matrix D_{hidden}.

Set $\mathbf{x}_i$ as the ith fingerprint in $\mathscr{F}_{\text{BT}}$.

end for

5. Scale fingerprints $\mathscr{F}_{\text{BT}}=\{\mathbf{x}_i\}_{i=1}^N$ so that they have unit variance and zero mean.
6. Send scaled fingerprints $\mathscr{F}_{\text{BT}}=\{\mathbf{x}_i\}_{i=1}^N$ to the DT-MBO algorithm, described in Algorithm 1.

distribution over the two classes for a given data point may be skewed toward the majority class,[57] so Esposito *et al.*[1] tested potential decision thresholds from 0.05 to 0.5 with a spacing of 0.05. Similarly, in this work, we augment the MBO method by a process that optimizes the decision threshold for the displacement step, resulting in our proposed decision-threshold-MBO (DT-MBO) algorithm.

Two of the inputs of the DT-MBO algorithm, which is integrated into the proposed BTDT-MBO procedure, are a minimum threshold and a maximum threshold to test. In this study, we test a minimum threshold of 0.05 and a maximum threshold of 0.55, with thresholds between these two values spaced by 0.05. This allows us to choose a more optimal threshold than the usual one of 0.5.

Overall, we summarize the main iterative steps of the DT-MBO procedure for the data classification task used in this work below, with more details given in Algorithm 1. The process is adapted from the MBO algorithm used in our previous work,[40] with an additional decision threshold step in the displacement step. The below steps require a given labeled/unlabeled partition of a data set as well as a list of thresholds to test.

1. Diffusion step: compute $\mathbf{U}^{n+\frac{1}{2}}$ using the heat equation with an additional term using

$$\mathbf{U}^{n+\frac{1}{2}}=\mathbf{U}^n-\mathrm{d}t\mathbf{L}\mathbf{U}^{n+\frac{1}{2}}+\mu\cdot(\mathbf{U}^n-\mathbf{U}_{\text{labeled}}),$$

applied N_s times. Here, $dt>0$, and $\mathbf{U}_{\text{labeled}}$ is a matrix where a row corresponding to a labeled element is an indicator vector $\mathbf{e}_k$ if the node is of class k, and a zero vector for non-labeled points. Moreover, μ is a vector whose entries are equal to a constant for labeled points, and zero otherwise. We set $N_s=3$.

2. Projection step: compute $\mathbf{U}^{n+1}$ via projecting (each row) of the matrix $\mathbf{U}^{n+\frac{1}{2}}$ onto (3).
3. Displacement step: Replace every row of the result from step 2 by its assigned vertex $\mathbf{e}_k$ in the mth Gibbs simplex (3) depending on the decision threshold. This procedure is executed for every threshold in the list.
   (a) If the probability that the row belongs to the minority class is greater than (or equal to) the decision threshold, the row is replaced by the vertex corresponding to the minority class.
   (b) Otherwise, the row is replaced by the vertex for which it has the highest probability.

After repeating the above steps for a number of iterations, our model generates a predicted label matrix for each threshold. To select the optimal decision threshold from the set of thresholds for a given labeled/

DOI: 10.1142/S2737416524500479
J. Comput. Biophys. Chem. **2024**, *23* (10), 1339–1358

unlabeled partition of a data set, our model then computes the ROC-AUC score for each label matrix and subsequently chooses the threshold whose predicted labels yield the highest ROC-AUC score. Notably, the new thresholding procedure only involves alterations to the displacement step and does not require additional iterations of the diffusion or projection to simplex steps, thus saving computational time, in contrast to many cost-sensitive methods which require the complete retraining of the classifier. Section 3.3 contains discussion regarding this threshold optimization and its results.

2.1.2. *Bidirectional transformer*

As in our previous work,[40] the BTDT-MBO algorithm utilizes a bidirectional transformer to convert the SMILES input for each molecular compound to a vector in a latent space, which is then obtained as a molecular fingerprint, named BT-FP, for that compound. After being normalized, the BT-FPs for the compounds in a given data set comprise the features passed to the DT-MBO method for that data set. Specifically, the BTDT-MBO algorithm uses a particular bidirectional transformer model introduced by Chen *et al.*[43] involving an attention structure for self-supervised learning (SSL). The model, the self-supervised learning platform (SSLP), enables the generation of BT-FPs without the need for data labels. Moreover, the model used for BTDT-MBO can be chosen out of three SSLP models constructed by Chen *et al.*[43] The three SSLP models include one pretrained on the ChEMBL data set.[58] The other two models include one pretrained on the union of the ChEMBL and PubMed[59] data sets and one model trained on the union of the ChEMBL, PubMed and ZINC[60] data sets. Additionally, there is an option to fine-tune the selected self-supervised learning platform for specific predictive tasks.

In our prior work,[40] we demonstrated the predictive success of the BT-MBO model utilizing the self-supervised learning platform trained solely on ChEMBL for scarcely labeled molecular data. Consequently, the proposed BTDT-MBO model in this work uses the same SSLP trained on ChEMBL. Similarly, our BT-MBO model in Ref. 40 performed well without fine-tuning the SSLP, so we also bypass the fine-tuning step for the proposed BTDT-MBO model, instead passing our data directly to the pretrained SSLP.

2.1.3. *Overall procedure*

The overall two stages of the BTDT-MBO procedure are outlined as follows:

- Molecular data in the form of SMILES strings are passed to the self-supervised learning platform pretrained on ChEMBL data set.[58]
- The resulting BT-FPs are the rescaled so that they have zero mean as well as unit variance and then served as an input to the DT-MBO method, outlined in Algorithm 1.

More details of the stages are given in Algorithm 2. Notably, because the SSLP does not require data labels, this procedure can be applied to data with any number of classes. Paired with our DT-MBO method as outlined in Sec. 2.1.1 and Algorithm 1, our BTDT-MBO model can thus be used for problems with any number of clusters or classes.

2.2. Distance correlation

As discussed in Sec. 2.1.1, the similarity graph-based framework on which the DT-MBO scheme operates requires a weight function to compute the degree of similarity between two data elements. Various weight functions can be used, but a suitable weight function is defined so that a high degree of similarity between two elements is reflected by a large weight function output, and a low degree of similarity is reflected by a small weight function output.[64]

As in the authors' previous work[40] using the BT-MBO method, our proposed BTDT-MBO method utilizes a Gaussian kernel (1) as one of the ways to compute weights for the MBO scheme. By construction, when the distance between two data elements is 0, the Gaussian weight function evaluated at those two data elements equals 1. As the distance between two data elements approaches infinity, the Gaussian weight function approaches 0. In other words, the Gaussian weight function outputs values close to 0 for dissimilar data elements and 1 for identical data elements, with a weight function output closer to 1 reflecting greater similarity.

In this work, we further propose distance correlation[46] as a potential weight function for the DT-MBO scheme, motivated in part by the success of Hozumi *et al.*[47] in applying distance correlation to feature clustering. Also, as will be illustrated below, the distance correlation weight function scales similarly to the Gaussian weight function, with potential outputs

between 0 and 1, and higher weight function values corresponding to greater similarity, supporting our substitution. We hope to provide a new interpretation of the MBO method by incorporating the distance correlation as a weight function.

Distance correlation[46] is a measure of dependence between random vectors that generalizes of the idea of correlation. In particular, distance correlation is defined for pairs of vectors in arbitrary dimension, and the distance correlation of two vectors is 0 if and only if the two vectors are independent. We use the distance correlation between two vectors as defined in Ref. 46 and use the same notation below as in Ref. 47. First, given a vector $\mathbf{z}^i$, $i=1,2,...,I$, we can compute a distance matrix with entries defined by

$$a^i_{jk} = \left\| z^i_m - z^i_k \right\|, \quad m,k = 1,2,\ldots,M. \tag{4}$$

Here, $||\cdot||$ represents the Euclidean norm. Also, the doubly centered distance for $\mathbf{z}^i$ is given by

$$A^i_{jk} := a_{jk} - \bar{a}_{j\cdot} - \bar{a}_{\cdot k} + \bar{a}_{\cdot\cdot}, \tag{5}$$

where $\bar{a}_{j\cdot}$ represents the jth row mean. In addition, $\bar{a}_{\cdot_k}$ represents the kth column mean, and $\bar{a}_{\cdot\cdot}$ represents the grand mean of the distance matrix for $\mathbf{z}^i$.

Given $\mathbf{z}^i$ and $\mathbf{z}^j$, we can define the squared distance covariance as follows:

$$\mathrm{dCov}^2(\mathbf{z}^i,\mathbf{z}^j) := \frac{1}{M^2}\sum_j\sum_k A^i_{jk}A^j_{jk}. \tag{6}$$

Now, the distance correlation between $\mathbf{z}^i$ and $\mathbf{z}^j$ is defined as follows:

$$\mathrm{dCor}(\mathbf{z}^i,\mathbf{z}^j) := \frac{\mathrm{dCov}^2(\mathbf{z}^i,\mathbf{z}^j)}{\mathrm{dCov}(\mathbf{z}^i,\mathbf{z}^i)\mathrm{dCov}(\mathbf{z}^j,\mathbf{z}^j)}. \tag{7}$$

Note that the distance correlation $\mathrm{dCor}(\mathbf{z}^i,\mathbf{z}^j)$ can take on values in the range [0,1] (see a complete proof of this fact in Ref. 46). We have $\mathrm{dCor}(\mathbf{z}^i,\mathbf{z}^j)=0$ when vectors $\mathbf{z}^i$ and $\mathbf{z}^j$ are independent (or $\mathrm{dCov}^2(\mathbf{z}^i,\mathbf{z}^j)=0$), and we have $\mathrm{dCor}(\mathbf{z}^i,\mathbf{z}^j)=1$ when $\mathbf{z}^i$ and $\mathbf{z}^j$ are linearly dependent. Additionally, a higher squared distance covariance between two vectors corresponds to a distance correlation closer to 1 for those vectors.

Thus, the distance correlation (7) satisfies the conditions for a similarity weight function for the MBO scheme. As discussed above, the range of the distance correlation is identical to that of the Gaussian kernel, so computationally implementing the distance correlation as a weight function in our DT-MBO method is straightforward. Specifically, only the construction of the graph Laplacian $\mathbf{L}$ in Step 2 of Algorithm 1 differs when using the distance correlation to compute the graph weights. Furthermore, Székely *et al.*[46] proved relationships between the distance correlation and the standard bivariate normal distribution which support this substitution in our algorithm. Namely, if X and Y have standard normal distributions with correlation ρ, then $\mathrm{dCor}(X,Y) \le |\rho|$.

3. RESULTS AND DISCUSSION

In this section, we discuss the computational experiments and results. The code used for the DT-MBO algorithm was written in MATLAB. To carry out computational experiments and optimize parameters, we utilized the High Performance Computing Center (HPCC) at Michigan State University. For all jobs submitted through HPCC, we specified two CPUs per task. Distance correlation weights were computed using the `distance_correlation` function from the dcor library in Python.

3.1. Data sets

The data sets used in this work are characterized by a high level of imbalance in class sizes, measured by the imbalance ratio (IR). The IR of a data set with two classes is defined as the ratio of the number of points in the majority class to the number of points in the minority class.[1] Out of the public data sets Esposito *et al.*[1] used for benchmarking their proposed thresholding models, we selected data sets from each of the following classes to benchmark our methods:

- DS1 Data Sets (obtained from Ref. 1, introduced in Ref. 65 and refined in Ref. 66): Each of the DS1 data sets corresponds to a particular ChEMBL target and contains 100 associated diverse active compounds against that target. For each data set, we randomly chose 2,000 inactives from a set of 10,000 assumed inactives drawn from ZINC,[67,68] reproducing the data set construction procedure from Ref. 1. These 10,000 compounds were selected in Ref. 66 to have similar property distributions to the set of actives, with similarity measured by an atom-count fingerprint. Every DS1 data set has an IR of 20.0 by construction.

DOI: 10.1142/S2737416524500479
J. Comput. Biophys. Chem. **2024**, *23* (10), 1339–1358

Table 1. Information about the data used for benchmarking the model (results are shown in Figs. 3 and 7). All data sets group molecules into two classes. The first four data sets are from Refs. 1 and 69 and the last two data sets are from Refs. 1, 65 and 66.

Data Set	Data Set Grouping	# Compounds	Imbalance Ratio (IR)	Labels
CHEMBL1909150	DrugMatrix	842	16.9	Active, Inactive
CHEMBL1909157	DrugMatrix	842	18.6	Active, Inactive
CHEMBL1909132	DrugMatrix	842	20.0	Active, Inactive
CHEMBL1909134	DrugMatrix	842	16.5	Active, Inactive
CHEMBL100	DS1	2100	20.0	Active, Inactive
CHEMBL8	DS1	2100	20.0	Active, Inactive

- DrugMatrix Assays (obtained from Ref. 1): Each of the 44 DrugMatrix assays benchmarked in Ref. 1 contains results for 842 compounds tested in a particular assay, with each data set labeled by the assay tested. Each molecule was labeled as active or inactive based on its "activity comment" value in the source data. In this paper, we evaluate our proposed procedure on four of these data sets, with IRs ranging from 16.5 to 20.0. The source data was retrieved from ChEMBL and was originally recorded in the DrugMatrix database.[69]

A brief summary of the data sets used for testing the BTDT-MBO model is given in Table 1.

The authors' BT-MBO method from Ref. 40 performed better than the state-of-the-art techniques for scarcely labeled molecular data. Inspired by this success, in this work, we focus on the challenging task of making predictions on highly imbalanced data sets. All data sets used for benchmarking the proposed BTDT-MBO method have an IR of 16.5 or greater.

3.2. Evaluation metrics

There are various existing evaluation metrics for data classification tasks. One such metric is model accuracy, which computes the fraction of data points correctly classified by the model. However, this metric does not capture many nuances of model performance. Particularly when dealing with imbalanced data, as Esposito *et al.* discussed in Ref. 1, the minority class may be the class of most interest, but points from the minority class are more likely to be misclassified by the model than points from the majority class(es). In highly imbalanced data sets, such as the ones used in this work, there are comparatively very few points in the minority class. Even if the model misclassifies all of the points in the minority class, the performance of the technique could still be very high, and the model could mistakenly appear to be suitable for the highly imbalanced classification task. Thus, to better quantify the performance of our proposed method, we use another evaluation metric, which computes the area under the receiver operating characteristic curve (the ROC curve). This is also known as the ROC-AUC score.

For the data classification problem, the ROC curve plots the true positive rate versus the false positive rate of the classifier for various thresholds. The ROC curve captures useful information about the classifier, particularly on highly imbalanced data — because the minority (or positive, in our case) class can be easily misclassified, both the true positive rate (TPR) and the false positive rate (FPR) need to be considered. Recall that a classification threshold is used to predict a given data point's class using its probability distribution over all classes. If the probability that the data point is affiliated with the minority class is greater than the threshold, then that point is allotted to the minority class. For a threshold equal to 0, the model would predict that every data point belonged to the minority class. In this case, the model would successfully classify all of the true positive data points (i.e., a TPR of 1), but it would also incorrectly classify all of the negative data points as positive (i.e., an FPR of 1). As the threshold increased, fewer data points would be classified as positive. For a threshold equal to 1, the model would classify all data points as negative, producing a TPR and an FPR of 0.

The ROC-AUC score quantifies the relationship between the TPR and FPR by computing the area under the ROC curve. For an area close to 1, it is theoretically possible to choose an "optimal" threshold that yields a high TPR and a low FPR (i.e., produces a point close to the upper-left corner of the ROC plot). In general, an AUC of 0.5 suggests that algorithm has no ability to distinguish between the classes, 0.7 to 0.8 indicates an acceptable technique, 0.8 to 0.9 describes an excellent technique, and greater than 0.9 designates

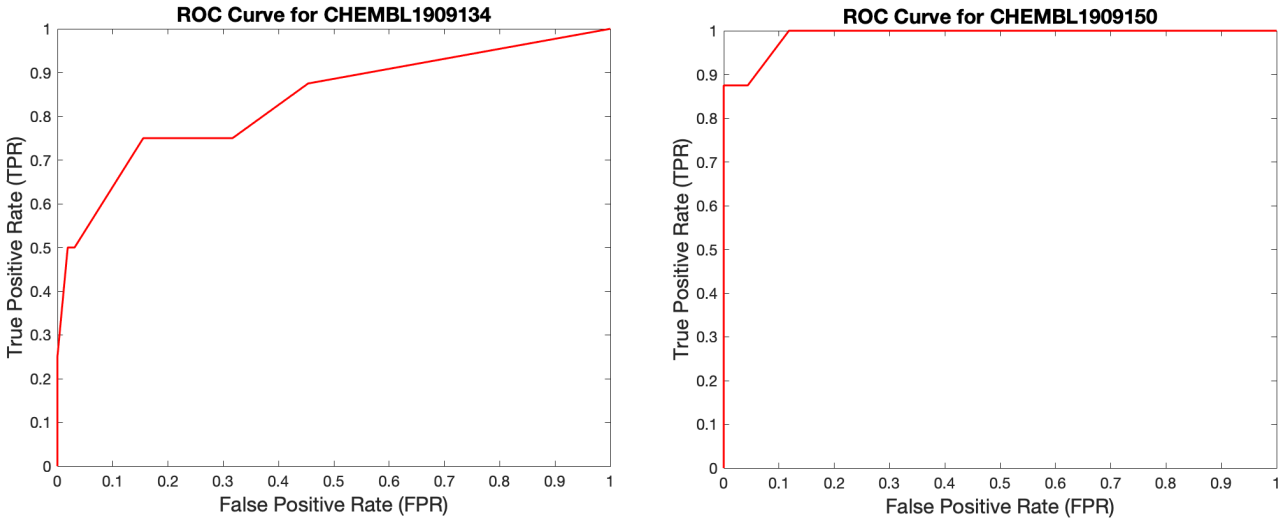

Fig. 2. Example ROC curves for two of the data sets (CHEMBL1909134 and CHEMBL1909150) used for benchmarking the proposed method. Curves were constructed in MATLAB for a random labeled/unlabeled split of each data set using a set of increasing thresholds from 0 to 1, with each successive threshold increasing by 0.05. The average ROC-AUC scores for both data sets over all 50 random partitions can be seen in Fig. 3.

an outstanding technique. To calculate the ROC-AUC score in our experiments for the proposed BTDT-MBO method, we used the `perfcurve` function in MATLAB. There also exist functions that calculate the ROC-AUC score for the multiclass case, so our evaluation procedure can be applied for any number of classes.

To further illustrate the construction of the ROC curves as well as their reflection of the classification performance of models, we have included example ROC curves in Fig. 2 for two data sets used in benchmarking the proposed BTDT-MBO model. These plots were generated using a random labeled/unlabeled split of each data set, and the thresholds used to construct each plot ranged from 0 to 1, with each threshold increasing by 0.05. The ROC curve on the right for the CHEMBL1909150 data set demonstrates near-ideal performance on this data partition, with a point very close to the upper-left corner of the plot, corresponding to an FPR close to 0 and a TPR close to 1. Indeed, as displayed in Fig. 3 and discussed in Sec. 3.3, our model's average ROC-AUC score for 50 random partitions of the CHEMBL1909150 data set was 0.978, displaying outstanding classification performance. The ROC curve on the left for the CHEMBL1909134 data set clearly has a lower ROC-AUC value than the curve for the CHEMBL1909150 data set, but the model still shows some ability to discriminate between the two classes. As shown in Fig. 3, our model's average ROC-AUC score over all partitions on the CHEMBL1909134 data set was 0.792, which is not surprising given its example ROC curve. Do note, however, that our proposed model performed superior to the comparison methods on this data set, which is further discussed in Sec. 3.3.

3.3. Model performance and discussion

To evaluate the performance of our BTDT-MBO method for highly imbalanced molecular data sets, we compare our models against the GHOST algorithm from Ref. 1. The GHOST algorithm is designed to be paired with any machine learning classifier, and the code used in Ref. 1 ran experiments using GHOST with random forest (RF), extreme gradient boosting (XGB), logistic regression (LR) and gradient boosting (GB) classifiers. In this work, all models used 80% of the data as training (or labeled, in the MBO case) to mirror the experiments by Esposito *et al.*[1]

Figures 3 and 4 display the results of the BTDT-MBO procedure compared with the GHOST method from Ref. 1. We generated results using GHOST paired with LR, RF, GB and XGB. The GHOST model allows the user to select the desired classification metric to use for optimization from two options: the Cohen's kappa, which is a balanced classification metric that was used as the default metric in Ref. 1, or the ROC-AUC score. All of the experiments in this paper were evaluated using the ROC-AUC score.

Overall, the experimental results indicate that the BTDT-MBO algorithm obtains higher ROC-AUC

DOI: 10.1142/S2737416524500479
J. Comput. Biophys. Chem. **2024**, *23* (10), 1339–1358

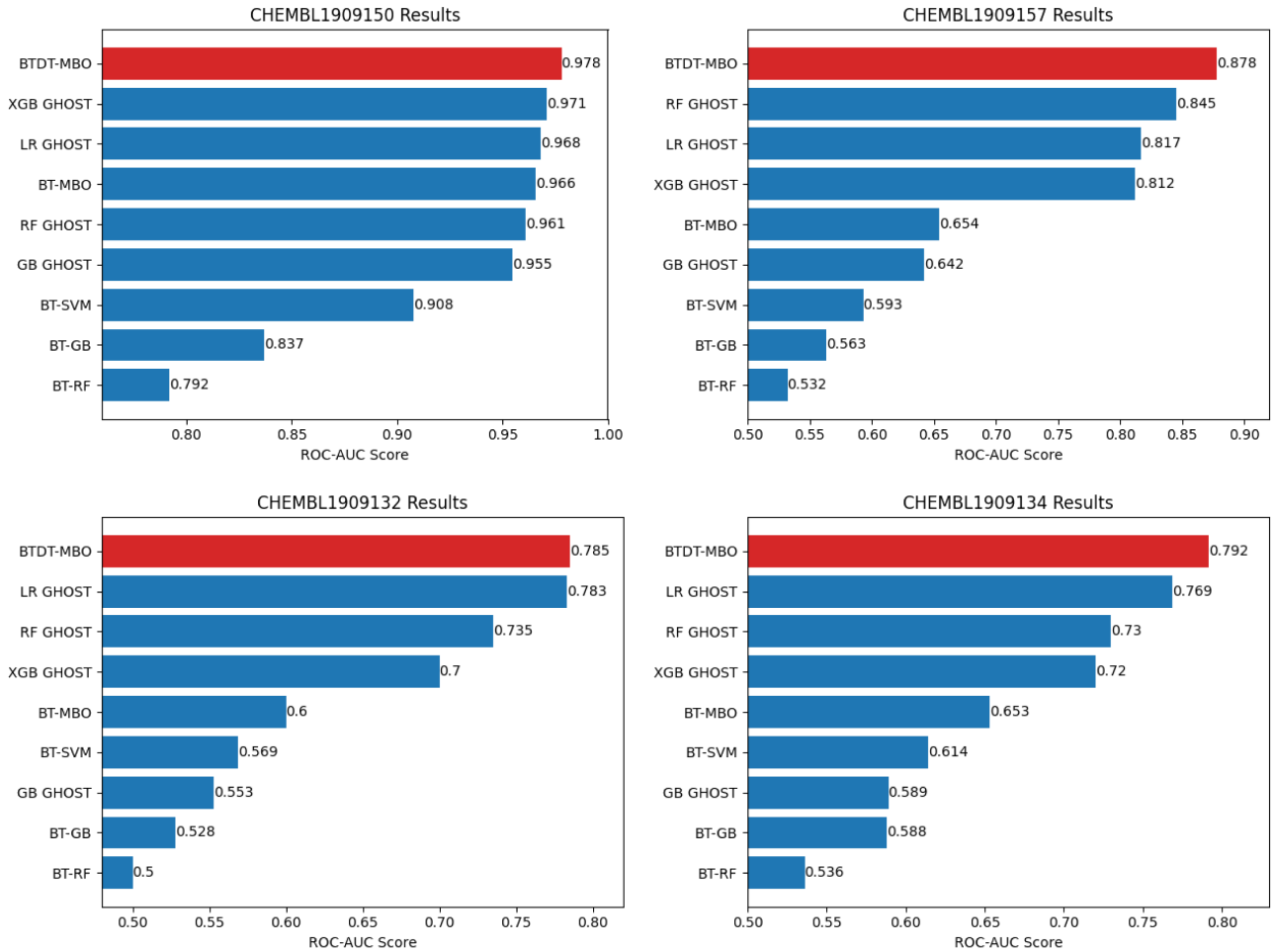

Fig. 3. (Color online) Comparison to other techniques on DrugMatrix data sets. The results of our proposed method are in red, while those of other algorithms are in blue. The imbalance ratios for the pictured data sets vary from 16.5 to 20.0. Detailed information about the overall size and composition of the comparison data sets is described in Sec. 3.1. The performance metric is the ROC-AUC score averaged over 50 random training-testing (or labeled/unlabeled) splits of the given data, with 80% of the data being labeled in each case. The BTDT-MBO result for each data set is the highest of the BTDT-MBO model using a Gaussian weight function and the BTDT-MBO model using a distance correlation weight function. Some comparison results were generated using the GHOST algorithm[1] via random forests (RF), extreme gradient boosting (XGB), logistic regression (LR) and gradient boosting (GB). Additional comparison results were generated using the BT-MBO algorithm[40] as well as BT-GB, BT-RF and BT-SVM models (consisting of the BT-FPs passed to gradient boosting, random forest and support vector machine algorithms, respectively).

scores than the comparison methods for all data, except CHEMBL8, for which the result of the proposed algorithm is second best, but almost the same as the best. We first outline the details of the experiments in this work, after which we provide a summary as well as a further discussion of results.

To create labeled and unlabeled partitions in the data input to our DT-MBO method, we randomly selected 80% of the points to be labeled and denoted the rest as unlabeled. Then, we applied the DT-MBO method, which propagates the known labels to the unlabeled nodes for a prescribed number of iterations as detailed in Sec. 2.1.1. We used the resulting predicted labels to calculate the optimal ROC-AUC score for this partition (over all thresholds). In highly imbalanced data sets, which often contain very few points in the minority class, results can vary widely depending on the particular training/testing or labeled/unlabeled partition. To account for this variability, as in Ref. 1, we repeated each experiment 50 times, with a new random partition for each experiment, and averaged the ROC-AUC scores over the 50 trials. We used a similar splitting and averaging procedure to generate the GHOST comparison results; however, Esposito *et al.*[1] recorded

DOI: 10.1142/S2737416524500479

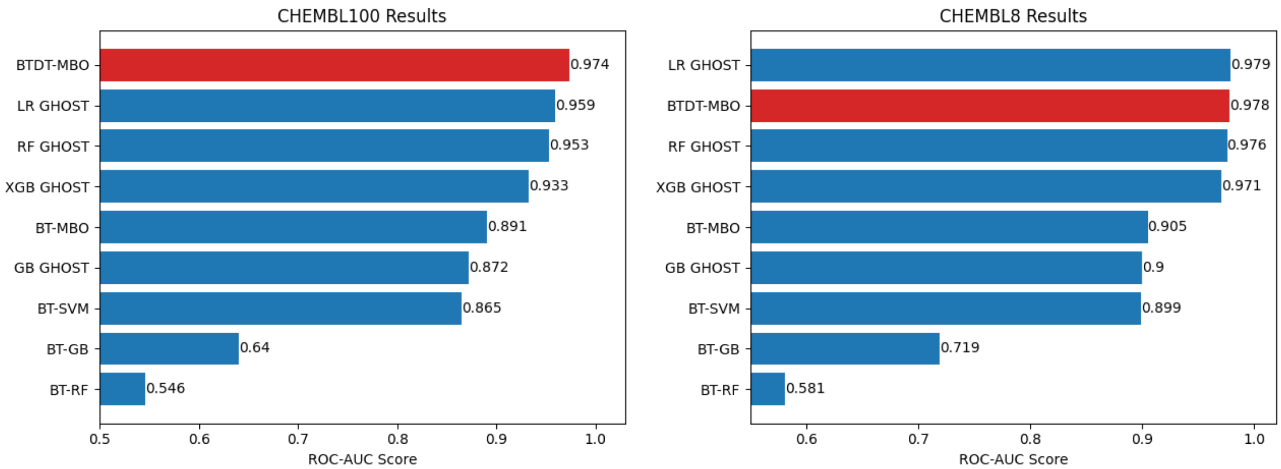

Fig. 4. (Color online) Comparison to other techniques on two DS1 data sets. The results of our proposed method are in red, while those of other algorithms are in blue. The imbalance ratios for the pictured data sets are both 20.0. Detailed information about the overall size and composition of the comparison data sets is described in Sec. 3.1. The metric is the ROC-AUC score averaged over 50 splits of the data, with 80% of the data being labeled in each case. The BTDT-MBO result for each data set is the highest of the BTDTMBO model using a Gaussian weight function and the BTDT-MBO model using a distance correlation weight function. Some comparison results were generated using the GHOST algorithm[1] via random forests (RF), extreme gradient boosting (XGB), logistic regression (LR) and gradient boosting (GB). Additional comparison results were generated using the BT-MBO algorithm[40] as well as BT-GB, BT-RF and BT-SVM models (consisting of the BT-FPs passed to gradient boosting, random forest and support vector machine algorithms, respectively).

the random seeds used for each training/testing data split in their experiments, and we used the same 50 random seeds when running the GHOST tests. Results in Figs. 3 and 4 indicate the average ROC-AUC score over 50 different splits of the data.

Regarding parameters, the proposed BTDT-MBO model requires specification of various parameters, whose functions can be seen in Algorithms 2 and 1. The bidirectional transformer[43] requires the user to choose one of the three pretraining data sets as outlined in Sec. 2.1.2. In this work, all BTDT-MBO experiments use the transformer model trained only on the ChEMBL data set,[58] based on its success in predicting scarcely labeled molecular data in experiments from the authors' previous work[40] using the BT-MBO method. The transformer model does not require any additional parameter specifications, as all other parameters are implicit in the pretrained models.[43,70]

The DT-MBO algorithm relies on several parameters, many of which are discussed in our previous work,[40] but some of which have been added in this work to implement the decision threshold adjustment. Some parameters were defined manually prior to running all experiments, and other parameters were tuned in our experiments. Here, we provide a brief description of all parameters and their settings or tuning. First, to create the framework in which the DT-MBO algorithm operates, we construct a N_n-neighbor graph. After, the graph Laplacian $\mathbf{L}$ (2) is constructed, and the first (smallest) N_e eigenvalues and corresponding eigenvectors of $\mathbf{L}$ are computed. Overall, both N_n and N_e were designated as hyperparameters and tuned in our experiments. Other hyperparameters for the DT-MBO method that were tuned are the C and dt parameters in the diffusion step, as well as the number of iterations, N_t.

Regarding other parameters, we let $N_s = 3$ and $N_l = 50$. Additionally, the number of labeled points N_p was set at 80% of each data set for all tasks. Finally, the decision threshold adjustment steps in our algorithm require the specification of a minimum threshold and a maximum threshold to test. For all BTDT-MBO experiments, we used a minimum threshold of 0.05 and a maximum threshold of 0.55.

As previously stated, the proposed BTDT-MBO algorithm obtained higher ROC-AUC scores than the compared methods in all but one of the six highly imbalanced data sets included in this work. Figure 3 displays our model's results (using the best-performing of the BTDT-MBO with Gaussian weight function and BTDT-MBO with distance correlation weight function) compared with the GHOST algorithm's results on four highly imbalanced DrugMatrix data sets. The BTDT-MBO procedure obtained a higher ROC-AUC

DOI: 10.1142/S2737416524500479
J. Comput. Biophys. Chem. **2024**, *23* (10), 1339–1358

score than all GHOST models for all four of the DrugMatrix data sets. As indicated in Table 1, each of the four DrugMatrix data sets contained 842 compounds and had IRs of at least 16.5. Our model achieved its best predictive performance on the CHEMBL1909150 data set with an average ROC-AUC score of 0.978, indicating an outstanding ability to discriminate between the active and inactive classes. Furthermore, our model outperformed all GHOST models, which also demonstrated very high ROC-AUC scores on this data set. For CHEMBL1909157, our procedure obtained excellent results, with an ROC-AUC score of 0.866. Some of the GHOST models also achieved ROC-AUC scores greater than 0.8, but the GHOST model using GB performed significantly worse. Our model similarly outperformed all four GHOST comparison models on the CHEMBL1909132 and CHEMBL1909134 data sets.

Figure 4 shows our model's results on two highly imbalanced DS1 data sets, again compared with the GHOST algorithm paired with LR, RF, XGB and GB. Recall from Sec. 3.1 that all DS1 data sets have an IR of 20.0 by construction. For CHEMBL100, our method achieved an outstanding ROC-AUC score of 0.974, while the highest-performing GHOST model (using LR) earned an ROC-AUC score of 0.959. While our BTDT-MBO method did not achieve the highest ROC-AUC score of comparison algorithms on CHEMBL8, it obtained the second-best ROC-AUC score, performing almost as well as the top-performing GHOST model using LR. All models in this case demonstrated ROC-AUC scores of at least 0.9, with the highest method earning a score of 0.979 and BTDT-MBO scoring 0.978.

As outlined in Sec. 2.2, in addition to the default Gaussian kernel used in our BTDT-MBO method, we also conducted experiments that instead used the distance correlation to compute the weights between data elements. In our experiments, the BTDO-MBO method using a Gaussian weight function yielded higher ROC-AUC scores than the BTDT-MBO method using distance correlation for five out of the six data sets included in this work, with the BTDT-MBO algorithm using distance correlation achieving the best results on the CHEMBL1909157 DrugMatrix data set. However, the BTDT-MBO algorithm using a distance correlation weight function performed similarly to (and frequently almost as well as) the BTDT-MBO method using a Gaussian kernel on all six data sets, particularly on the DS1 data sets used for benchmarking. For the CHEMBL100 DS1 data set, the BTDT-MBO model using distance correlation yielded an ROC-AUC score of 0.970, slightly lower than BTDT-MBO using Gaussian weights (ROC-AUC = 0.974) but still higher than all comparison methods. Similarly, for the CHEMBL8 DS1 data set, BTDT-MBO using distance correlation scored 0.973, which was again lower than BTDT-MBO using a Gaussian weight function (ROC-AUC = 0.978) but higher than two comparison methods. All tested models scored very highly on this data set.

Our model using distance correlation saw similar results on the DrugMatrix data sets. In fact, as mentioned above, BTDT-MBO using distance correlation achieved the best overall result of all tested models on the CHEMBL1909157 DrugMatrix data set, earning an ROC-AUC score of 0.878. BTDT-MBO using a Gaussian kernel scored an ROC-AUC value of 0.866; thus, both BTDT-MBO methods beat all tested GHOST models. For the CHEMBL1909150 data set, BTDT-MBO using distance correlation scored an ROC-AUC value of 0.975, higher than all comparison methods and only lower than our BTDT-MBO model using a Gaussian, which scored 0.978. On the CHEMBL1909132 DrugMatrix data set, BTDT-MBO using distance correlation earned an ROC-AUC score of 0.743, which places it below BTDT-MBO using Gaussian weights (ROC-AUC = 0.785) and GHOST using LR, but above the three other GHOST comparison methods. Our model using distance correlation saw similar relative results for the CHEMBL1909134 DrugMatrix data set, with an average ROC-AUC score of 0.766. The model earned the third-highest score of all methods tested for this data set, lower than our BTDT-MBO method using Gaussian weights (ROC-AUC = 0.792) and very close to GHOST paired with LR, which scored a 0.769. Overall, these results suggest that the distance correlation can be useful for quantifying similarity in molecular data sets, including those that are highly imbalanced. Furthermore, the distance correlation may be considered as a potential alternative weight function to the Gaussian function in similarity graph-based settings, particularly when the data is not close to a normal distribution.

To further visualize our proposed model's performance on the benchmark data sets, we additionally utilized the residue-similarity (R-S) scores and associated R-S plots introduced by Hozumi *et al.*[47] For a given data point in a particular class, its residue score is defined as the sum of the distances from that point to points in other classes. The point's similarity score is defined as the average distance from that point to points in its own class. In other words, the residue score measures how well data points in a given class are

DOI: 10.1142/S2737416524500479

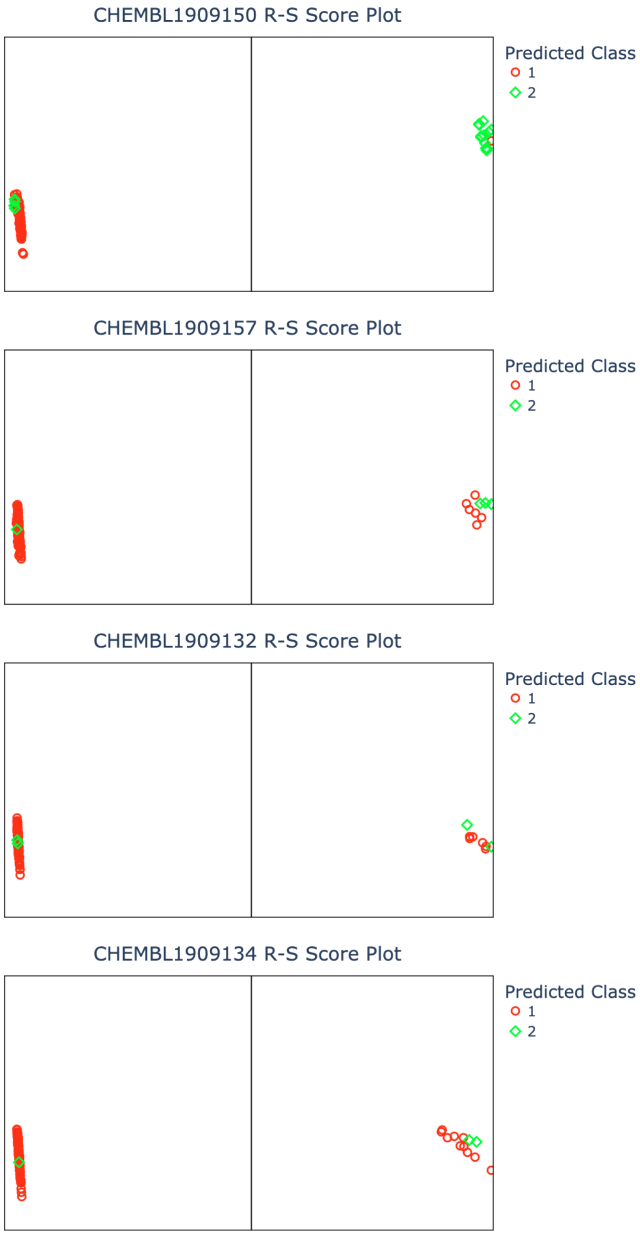

Fig. 5. (Color online) R-S score plots for the four DrugMatrix data sets. The plots display R-S scores of unlabeled points from a random labeled/unlabeled partition of each data set. The *x*- and *y*-axes of the plots represent residue and similarity scores, respectively. From left to right, the panels plot points in class 1 (i.e., inactive compounds) and class 2 (i.e., active compounds). Each point is colored based on its class predicted by the proposed BTDT-MBO model.

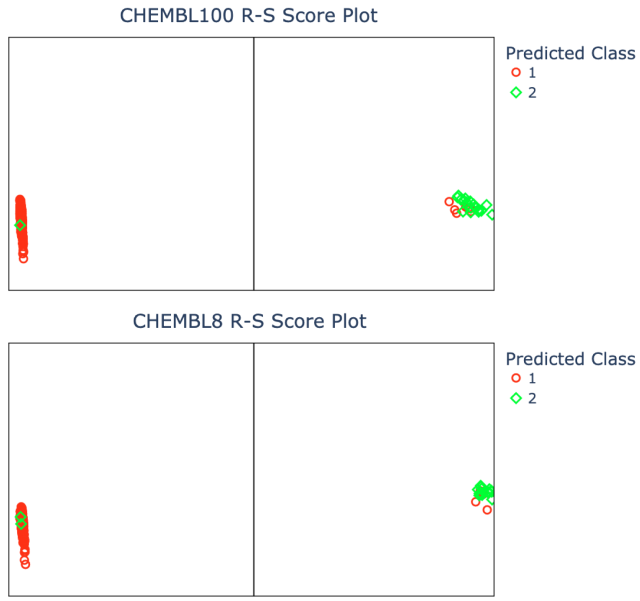

Fig. 6. (Color online) R-S score plots for the two DS1 data sets. The plots display R-S scores of unlabeled points from a random labeled/unlabeled partition of each data set. The *x*- and *y*-axes of the plots represent residue and similarity scores, respectively. From left to right, the panels plot points in class 1 (i.e., inactive compounds) and class 2 (i.e., active compounds). Each point is colored based on its class predicted by the proposed BTDT-MBO model.

separated from other classes, and the similarity score measures how well data points in a given class are clustered together. Given a training/testing (or labeled/unlabeled) split of a data set, one can construct R-S score plots visualizing the elements of each class separately. In Figs. 5 and 6, we plot R-S scores for a random labeled/unlabeled split of the six data sets used in our work. Points are separated into panels by their true class label, and the color of each point corresponds to its predicted class using our proposed BTDT-MBO model. We use the optimal weight function for each data set as discussed above.

Another novel addition to the proposed BTDT-MBO method is its threshold optimization step, the details of which are given in Sec. 2.1.1. This step chose the decision threshold that yielded the highest ROC-AUC score for a given partition of a particular data set. Because the partitions were created randomly, the class distribution of the unlabeled and labeled portions of a data set could vary. Additionally, due to the high level of class imbalance in the data, there may be very few data points from the minority class in the labeled or unlabeled set for a particular partition. Therefore, it is not surprising that in our experiments, the optimal thresholds for the splits varied considerably. However, the optimal thresholds often tended to be lower than the typical threshold of 0.5. For example, in one experiment on the CHEMBL1909157 data set (i.e., testing over 50 random labeled/unlabeled partitions), the lowest optimal threshold for the 50 random splits was 0.05, and the highest optimal threshold was 0.45. The median optimal threshold over all 50 splits was 0.15.

DOI: 10.1142/S2737416524500479
J. Comput. Biophys. Chem. **2024**, *23* (10), 1339–1358

These results reinforce the impact and the advantage of adjusting the decision threshold on classifier performance for highly imbalanced data.

3.4. Statistical testing

In this paper, we perform statistical analysis using the experimental results; we perform the Friedman and the post-hoc Nemenyi test in order to achieve a more thorough evaluation. The results are shown in Table 2 and Fig. 7, including the mean rank values, where smaller ranks demonstrate more competitive techniques. The critical distance was around 4.29, with the significance level being 0.05. Overall, the Friedman test's null hypothesis is that all algorithms have the same performance. If the *p*-value is less than 0.05, the null hypothesis is rejected, which it was in our testing. Moreover, the Nemenyi post-hoc test is utilized to identify which procedures are significantly different from each other. Overall, it is shown that the proposed algorithm is ranked consistently the best among comparison methods.

3.5. Future work and limitations

While the proposed method has many advantages, such as the ability to perform accurately with highly imbalanced data, it has certain limitations similarly to any other algorithm. For example, the proposed procedure requires the calculation of a small number of the

Table 2. Results for statistical testing.

(a) Reference for (b) and (c)

	Method
'1'	BTDT-MBO (Proposed)
'2'	LR-GHOST
'3'	RF-GHOST
'4'	XGB-GHOST
'5'	BT-MBO
'6'	GB-GHOST
'7'	BT-SVM
'8'	BT-GB
'9'	BT-RF

(b) Results of Friedman's Test

Chi-sq statistic	*p*-value
112.39130	2.4237e-07

(c) Mean rank values of the Friedman/Nemenyi test

'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'
1.16667	2.16667	3.16667	3.66667	4.83333	6.33333	6.66667	8.00000	9.00000

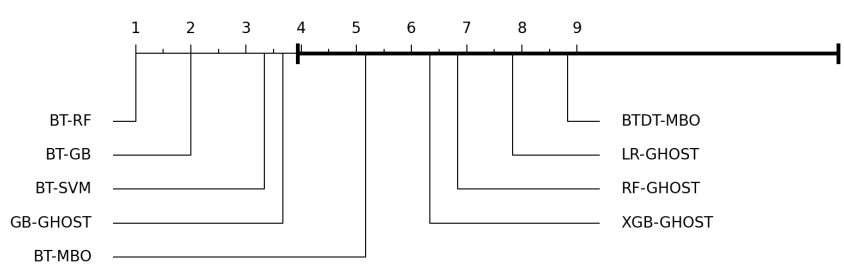

Fig. 7. The average rank diagram of the Friedman/Nemenyi test.

graph Laplacian's eigenvectors and corresponding eigenvalues. This can obviously be computationally expensive for certain data sets. One possible direction in such a scenario is to construct an approximation of the full graph using other techniques, e.g., sampling-based procedures, in particular the Nyström Extension technique, detailed in sources such as Refs. 61, 62 and 63. Such a technique can be utilized to very efficiently compute approximations of the eigenvalues and eigenvectors for the proposed procedure even for large data sets.

Future work regarding this study includes exploring imposing class size constraints into the technique. Moreover, we would like to explore integrating other correlation techniques, as well as incorporating extended-connectivity fingerprints or autoencoders into the algorithm structure.

4. CONCLUSION

Given the prevalence of molecular data where the class sizes vary vastly, the ability of an algorithm to accurately predict the class of data elements in imbalanced data sets is extremely important. However, many classification techniques are not accurate on highly imbalanced data as they tend to overestimate majority classes. This paper presents a BTDT-MBO method for molecular data classification in case of imbalanced data sets with highly varied class sizes. The method, which integrates Merriman–Bence–Osher procedures as well as a bidirectional transformer, incorporates adjustments in the data classification threshold for machine learning classifiers to handle the high class imbalance. To illustrate the advantages of the proposed technique, particular attention is given to highly imbalanced data sets.

Numerical experiments indicate that the BTDT-MBO procedure performs very well even with high class imbalance ratios; in particular, the computational experiments on DS1 data sets and DrugMatrix data sets demonstrate that the results of the proposed procedure are almost always more accurate than the comparison algorithms such as Ref. 1, as indicated by Figs. 2 and 3. The ROC-AUC score was used as a metric. Overall, the new model serves as a powerful machine learning tool for data sets with a high class imbalance.

This work additionally investigates the distance correlation as a choice for a weight function in the proposed BTDT-MBO algorithm, serving as an alternative to the Gaussian kernel often used to compute graph weights. Overall, the results for the BTDT-MBO technique using a distance correlation weight function are similar to those using Gaussian weights, with the distance correlation model performing better for one data set; these results suggest further utility of the distance correlation in imbalanced molecular data classification and similarity graph-based methods. In particular, data that is far from a normal distribution may benefit from the use of a distance correlation weight function rather than a Gaussian weight function.

FUNDING INFORMATION

The work of this paper is supported partly by the NSF DMS-2052983 grant. The third author was supported partly by NIH R01GM126189, R01AI164266 and R35GM148196 grants, NSF grants DMS-1761320, DMS-2052983, and IIS-1900473, NASA grant 80NSSC21M0023, Michigan Economic Development Corporation, MSU Foundation, Pfizer, and Bristol-Myers Squibb 65109.

CONFLICT OF INTEREST

All authors declare that they have no conflict of interest.

AUTHOR'S CONTRIBUTIONS

Nicole Hayes was responsible for the software, computational experiments and writing. Ekaterina Rapinchuk and Guo-Wei Wei were responsible for methodology, reviewing, editing, some writing and supervision.

DATA AVAILABILITY

The data sets in this paper are publicly available at https://github.com/rinikerlab/GHOST/tree/main/data.

STATEMENT OF USAGE OF ARTIFICIAL INTELLIGENCE

The authors have not received any help from artificial intelligence such as ChatGPT while preparing this paper.

ORCID

Nicole Hayes https://orcid.org/0000-0003-1772-0306
Ekaterina Merkurjev https://orcid.org/0000-0002-2489-8332
Guo-Wei Wei https://orcid.org/0000-0001-8132-5998

DOI: 10.1142/S2737416524500479
J. Comput. Biophys. Chem. **2024**, *23* (10), 1339–1358

References

1. Esposito, C.; Landrum, G. A.; Schneider, N.; Stiefl, N.; Riniker, S. GHOST: Adjusting the Decision Threshold to Handle Imbalanced Data in Machine Learning. *J. Chem. Inf. Model.* **2021**, *61*, 2623–2640.
2. Haixiang, G.; Li, Y.; Shang, J.; Mingyun, G.; Yuanyue, H.; Gong, B. Learning from Class-Imbalanced Data: Review of Methods and Applications. *Expert Syst. Appl.* **2016**, *73*, 220–239.
3. Lopez-del Rio, A.; Picart-Armada, S.; Perera-Lluna, A. Balancing Data on Deep Learning-based Proteochemometric Activity Classification. *J. Chem. Inf. Model.* **2021**, *61*, 1657–1669.
4. Guan, S.; Fu, N. Class Imbalance Learning with Bayesian Optimization Applied in Drug Discovery. *Sci. Rep.* **2022**, *12*, 2069.
5. Korkmaz, S. Deep Learning-based Imbalanced Data Classification for Drug Discovery. *J. Chem. Inf. Model.* **2020**, *60*, 4180–4190.
6. Xu, S.; Yuan, H.; Li, L.; Yang, K.; Zhao, L. Computational Screening of Potential Fatty Acid Synthase Inhibitors as Broad-Spectrum Antiviral Agents. *J. Comput. Biophys. Chem.* **2024**, *23*, 223–241.
7. Tian, Y.; Li, Y.; Zheng, X.; Peng, Q.; Shen, S. Pharmacophore-Based Virtual Screening Toward the Discovery of Novel GLUT1 Inhibitors. *J. Comput. Biophys. Chem.* **2022**, *21*, 951–966.
8. Nair, V.; Pradeep, S.; Nair, V.; Pournami, P.; Gopakumar, G.; Jayaraj, P. Deep Sequence Models for Ligand-Based Virtual Screening. *J. Comput. Biophys. Chem.* **2022**, *21*, 207–217.
9. Casanova-Alvarez, O.; Morales-Helguera, A.; Cabrera-Pérez, M. A.; Molina-Ruiz, R.; Molina, C. A Novel Automated Framework for QSAR Modeling of Highly Imbalanced Leishmania High-throughput Screening Data. *J. Chem. Inf. Model.* **2021**, *61*, 3213–3231.
10. Gu, Q.; Hou, J.; Gao, H.; Shi, M.; Zhuang, Y.; Wu, Q.; Zheng, L. 3D-QSAR and Molecular Docking Studies of Novel GPR52 Agonists. *J. Comput. Biophys. Chem.* **2023**, *22*, 615–626.
11. Soares, T.; Nunes-Alves, A.; Mazzolari, A.; Ruggiu, F.; Wei, G.-W.; Merz, K. The (Re)-Evolution of Quantitative Structure–Activity Relationship (QSAR) Studies Propelled by the Surge of Machine Learning Methods. *J. Chem. Inf. Model.* **2022**, *62*, 5317–5320.
12. Kaur, H.; Pannu, H.; Malhi, A. A Systematic Review on Imbalanced Data Challenges in Machine Learning: Applications and Solutions. *ACM Comput. Surv.* **2019**, *52*, 1–36.
13. Sun, Y.; Wong, A.; Kamel, M. Classification of Imbalanced Data: A Review. *Int. J. Pattern Recogn. Artif. Intell.* **2009**, *23*, 687–719.
14. Ganganwar, V. An Overview of Classification Algorithms for Imbalanced Datasets. *Int. J. Emerg. Technol. Adv. Eng.* **2012**, *2*, 42–47.
15. Estabrooks, A.; Jo, T.; Japkowicz, N. A Multiple Resampling Method for Learning from Imbalanced Data Sets. *Comput. Intell.* **2004**, *20*, 18–36.
16. García, V.; Sánchez, J.; Mollineda, R. On the Effectiveness of Preprocessing Methods When Dealing with Different Levels of Class Imbalance. *Knowl.-Based Syst.* **2012**, *25*, 13–21.
17. Fernández, A.; García, S.; Galar, M.; Prati, R.; Krawczyk, B.; Herrera, F.; Fernández, A.; García, S.; Galar, M.; Prati, R. Algorithm-level Approaches. In *Learning from Imbalanced Data Sets*. Springer, Cham, 2018, pp. 123–146.
18. Galar, M.; Fernandez, A.; Barrenechea, E.; Bustince, H.; Herrera, F. A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-based Approaches. *IEEE Trans. Syst., Man, Cybern. C Appl. Rev.* **2011**, *42*, 463–484.
19. Elkan, C. The Foundations of Cost-sensitive Learning. *Int. Joint Conf. Artificial Intelligence*. 2001, pp. 973–978.
20. Cao, H.; Li, X.-L.; Woon, D. Y.-K.; Ng, S.-K. Integrated Oversampling for Imbalanced Time Series Classification. *IEEE Trans. Knowl. Data Eng.* **2013**, *25*, 2809–2822.
21. Nekooeimehr, I.; Lai-Yuen, S. Adaptive Semi-unsupervised Weighted Oversampling (A-SUWO) for Imbalanced Datasets. *Expert Syst. Appl.* **2016**, *46*, 405–416.
22. Kumar, N.; Rao, K.; Govardhan, A.; Reddy, K.; Mahmood, A. Undersampled K-means Approach for Handling Imbalanced Distributed Data. *Prog. Artif. Intell.* **2014**, *3*, 29–38.
23. Anand, A.; Pugalenthi, G.; Fogel, G.; Suganthan, P. An Approach for Classification of Highly Imbalanced Data Using Weighting and Undersampling. *Amino Acids* **2010**, *39*, 1385–1391.
24. Song, J.; Huang, X.; Qin, S.; Song, Q. A Bi-directional Sampling based on K-means Method for Imbalance Text Classification. *2016 IEEE/ACIS 15th Int. Conf. Computer and Information Science*. 2016, pp. 1–5.
25. Cateni, S.; Colla, V.; Vannucci, M. A Method for Resampling Imbalanced Datasets in Binary Classification Tasks for Real-world Problems. *Neurocomputing* **2014**, *135*, 32–41.
26. López, V.; Fernández, A.; García, S.; Palade, V.; Herrera, F. An Insight Into Classification with Imbalanced Data: Empirical Results and Current Trends on Using Data Intrinsic Characteristics. *Inf. Sci.* **2013**, *250*, 113–141.
27. Sheng, V.; Ling, C. Thresholding for Making Classifiers Cost-sensitive. In *Proc. AAAI Conf. Artificial Intelligence*. 2006, pp. 476–481.
28. Zou, Q.; Xie, S.; Lin, Z.; Wu, M.; Ju, Y. Finding the Best Classification Threshold in Imbalanced Classification. *Big Data Res.* **2016**, *5*, 2–8.

DOI: 10.1142/S2737416524500479

29. Fawcett, T. An Introduction to ROC Analysis. *Pattern Recogn. Lett.* **2006**, *27*, 861–874.
30. Johnson, J.; Khoshgoftaar, T. Thresholding Strategies for Deep Learning with Highly Imbalanced Big Data. In *Deep Learning Applications, Volume 2*. Springer, Singapore, 2021, pp. 199–227.
31. Esposito, C.; Wang, S.; Lange, U.; Oellien, F.; Riniker, S. Combining Machine Learning and Molecular Dynamics to Predict P-glycoprotein Substrates. *J. Chem. Inf. Model.* **2020**, *60*, 4730–4749.
32. Yin, L.; Wang, L.; Lu, S.; Wang, R.; Yang, Y.; Yang, B.; Liu, S.; Alsanad, A.; Alqahtani, S.; Yin, Z.; Li, X.; Chen, X.; Zheng, W. Convolution-Transformer for Image Feature Extraction. *Comput. Model. Eng. Sci.* **2024**, *141*, 87–106.
33. Wang, B.; Zheng, W.; Wang, R.; Lu, S.; Yin, L.; Wang, L.; Yin, Z.; Chen, X. Stacked Noise Reduction Auto Encoder–OCEAN: A Novel Personalized Recommendation Model Enhanced. *Systems* **2024**, *12*, 188.
34. Zhu, C. An Adaptive Agent Decision Model Based on Deep Reinforcement Learning and Autonomous Learning. *J. Logist. Inform. Serv. Sci.* **2023**, *10*, 107–118.
35. Yin, L.; Wang, L.; Lu, S.; Wang, R.; Ren, H.; Alsanad, A.; Alqahtani, S.; Yin, Z.; Li, X.; Zheng, W. AFBNet: A Lightweight Adaptive Feature Fusion Module for Super-Resolution Algorithms. *Comput. Model. Eng. Sci.* **2024**, *140*, 2315–2347.
36. Peng, J.; Chen, X.; Wang, X.; Wang, J.; Long, Q.; Yin, L. J. Picture Fuzzy Decision-Making Theories and Methodologies: A Systematic Review. *Int. J. Syst. Sci.* **2023**, *54*, 2663–2675.
37. Jiao, N.; Wang, S.; Ma, J.; Liu, T.; Zhou, D. Sideband Harmonic Suppression Analysis Based on Vector Diagrams for CHB Inverters Under Unbalanced Operation. *IEEE Trans. Ind. Electron.* **2024**, *71*, 427–437.
38. Cheng, Y.; Deng, X.; Qi, Q.; Yan, X. Truthfulness of a Network Resource-Sharing Protocol. *Math. Oper. Res.* **2023**, *48*, 1522–1552.
39. Wang, G.; Li, R. DSolving: A Novel and Efficient Intelligent Algorithm for Large-scale Sliding Puzzles. *J. Exp. Theor. Artif. Intell.* **2017**, *29*, 809–822.
40. Hayes, N.; Merkurjev, E.; Wei, G. Integrating Transformer and Autoencoder Techniques with Spectral Graph Algorithms for the Prediction of Scarcely Labeled Molecular Data. *Comput. Biol. Med.* **2023**, *153*, 106479.
41. Dou, B.; Zhu, Z.; Merkurjev, E.; Ke, L.; Chen, L.; Jiang, J.; Zhu, Y.; Liu, J.; Zhang, B.; Wei, G.-W. Machine Learning Methods for Small Data Challenges in Molecular Science. *Chem. Rev.* **2023**, *123*, 8736–8780.
42. Weininger, D. SMILES, a Chemical Language and Information System. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 31–36.
43. Chen, D.; Zheng, J.; Wei, G.-W.; Pan, F. Extracting Predictive Representations from Hundreds of Millions of Molecules. *J. Phys. Chem. Lett.* **2021**, *12*, 10793–10801.
44. Garcia-Cardona, C.; Merkurjev, E.; Bertozzi, A.; Flenner, A.; Percus, A. Multiclass Data Segmentation Using Diffuse Interface Methods on Graphs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 1600–1613.
45. Merriman, B.; Bence, J. K.; Osher, S. J. Motion of Multiple Junctions: A Level Set Approach. *J. Comput. Phys.* **1994**, *112*, 334–363.
46. Székely, G.; Rizzo, M.; Bakirov, N. Measuring and Testing Dependence by Correlation of Distances. *Ann. Stat.* **2007**, *35*, 2769–2794.
47. Hozumi, Y.; Tanemura, K. A.; Wei, G.-W. Preprocessing of Single Cell RNA Sequencing Data Using Correlated Clustering and Projection. *J. Chem. Inf. Model.* **2024**, *64*, 2829–2838.
48. Merkurjev, E.; Kostic, T.; Bertozzi, A. An MBO Scheme on Graphs for Classification and Image Processing. *SIAM J. Imaging Sci.* **2013**, *6*, 1903–1930.
49. Merkurjev, E.; Nguyen, D. D.; Wei, G.-W. Multiscale Laplacian Learning. *Appl. Intell.* **2022**, *53*, 15727–15746.
50. Meng, Z.; Merkurjev, E.; Koniges, A.; Bertozzi, A. Hyperspectral Image Classification Using Graph Clustering Methods. *Image Process. On Line* **2017**, *7*, 218–245.
51. Merkurjev, E.; Bertozzi, A.; Chung, F. A Semi-supervised Heat Kernel Pagerank MBO Algorithm for Data Classification. *Commun. Math. Sci.* **2018**, *16*, 1241–1265.
52. Merkurjev, E.; Sunu, J.; Bertozzi, A. Graph MBO Method for Multiclass Segmentation of Hyperspectral Stand-off Detection Video. *IEEE Int. Conf. Image Processing*. 2014, pp. 689–693.
53. Bethuel, F.; Brezis, H.; Hélein, F. Asymptotics for the Minimization of a Ginzburg-Landau Functional. *Calc. Var. Partial Differ. Equ.* **1993**, *1*, 123–148.
54. Jerrard, R.; Soner, H. Limiting Behavior of the Ginzburg-Landau Functional. *J. Funct. Anal.* **2002**, *192*, 524–561.
55. Esedoglu, S.; Tsai, Y.-H. R. Threshold Dynamics for the Piecewise Constant Mumford–Shah Functional. *J. Comput. Phys.* **2006**, *211*, 367–384.
56. Merkurjev, E.; Garcia-Cardona, C.; Bertozzi, A. L.; Flenner, A.; Percus, A. Diffuse Interface Methods for Multiclass Segmentation of High-dimensional Data. *Appl. Math. Lett.* **2014**, *33*, 29–34.
57. King, G.; Zeng, L. Logistic Regression in Rare Events Data. *Pol. Anal.* **2001**, *9*, 137–163.
58. Gaulton, A. *et al.* The ChEMBL Database in 2017. *Nucl. Acids Res.* **2017**, *45*, D945–D954.

59. Kim, S.; Thiessen, P.; Bolton, E.; Chen, J.; Fu, G.; Gindulyte, A.; Han, L.; He, J.; He, S.; Shoemaker, B.; Wang, J.; Yu, B.; Zhang, J.; Bryant, S. PubChem Substance and Compound Databases. *Nucl. Acids Res.* **2016**, *44*, D1202–D1213.
60. Irwin, J.; Shoichet, B. ZINC — A Free Database of Commercially Available Compounds for Virtual Screening. *J. Chem. Inf. Model.* **2005**, *45*, 177–182.
61. Fowlkes, C.; Belongie, S.; Chung, F.; Malik, J. Spectral Grouping Using the Nyström Method. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 214–225.
62. Fowlkes, C.; Belongie, S.; Malik, J. Efficient Spatiotemporal Grouping Using the Nyström Method. In *Proc. 2001 IEEE Computer Society Conf. Computer Vision and Pattern Recognition. CVPR 2001*. 2001, pp. I–I.
63. Belongie, S.; Fowlkes, C.; Chung, F.; Malik, J. Spectral Partitioning with Indefinite Kernels Using the Nyström Extension. *European Conf. Computer Vision*. 2002, pp. 531–542.
64. Von Luxburg, U. A Tutorial on Spectral Clustering. *Stat. Comput.* **2007**, *17*, 395–416.
65. Riniker, S.; Fechner, N.; Landrum, G. Heterogeneous Classifier Fusion for Ligand-based Virtual Screening: Or, How Decision Making by Committee can be a Good Thing. *J. Chem. Inf. Model.* **2013**, *53*, 2829–2836.
66. Riniker, S.; Landrum, G. Open-source Platform to Benchmark Fingerprints for Ligand-based Virtual Screening. *J. Cheminformatics* **2013**, *5*, 26.
67. Irwin, J.; Sterling, T.; Mysinger, M.; Bolstad, E.; Coleman, R. ZINC: A Free Tool to Discover Chemistry for Biology. *J. Chem. Inf. Model.* **2012**, *52*, 1757–1768.
68. Sterling, T.; Irwin, J. ZINC 15 — Ligand Discovery for Everyone. *J. Chem. Inf. Model.* **2015**, *55*, 2324–2337.
69. DrugMatrix. https://ntp.niehs.nih.gov/data/drugmatrix, Accessed on 3 March 2024.
70. Gao, K.; Chen, D.; Robison, A. J.; Wei, G.-W. Proteome-informed Machine Learning Studies of Cocaine Addiction. *J. Phys. Chem. Lett.* **2021**, *12*, 11122–11134.

DOI: 10.1142/S2737416524500479