Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
ISMB 2020

Subject Section

CoRAE: Concrete Relaxation Autoencoder for Differentiable Gene Selection and Pan-Cancer Classification

Abdullah Al Mamun, Ananda Mohan Mondal*

School of Computing and Information Sciences
Florida International University, Miami, USA

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Selecting relevant features from a high-dimensional dataset is a critical study. It aims to select a small subset of features that will increase accuracy and decrease the cost of data classification or clustering. Due to high-dimension with a low number of samples in omics data, classification models encounter over-fitting problem. Thus, there is an urgent need for efficient feature selection methods that will be capable of selecting relevant features. In recent years, standard autoencoder and its variations have been used to select latent features to increase the classification performance. However, these methods are unable to provide which original features are contributing to these latent features. In this paper, we introduced a novel global feature selection method based on concrete relaxation discrete random variable selection, which can efficiently identify a subset of most significant features that have an effective contribution in data reconstruction and classification. The proposed method is a variation of standard autoencoder where a concrete feature selection layer is added in the encoder and a standard neural network is used as a decoder.
Results: We evaluated the proposed method using coding and non-coding gene expression profiles of 33 different cancers from TCGA. It significantly outperforms state-of-the-art methods in identifying top coding and non-coding genes. Later, expression values of selected genes are used to train a linear classifier to distinguish 33 cancer types where features selected by CoRAE shows highest performance in terms of four evaluation metrics: accuracy 99%, precision 98%, recall 98%, and f1 score 99%.

Availability: Source code and a sample dataset are available at https://github.com/pwaabdullah/CoRAE
Contact: amondal@fiu.edu
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

A major issue while working with recent omics data is the availability of greater number of features in comparison to the available number of samples leading to a highly imbalanced feature-sample ratio. It may be great to argue that the larger the feature set, the better the classification that is possible. However, on the contrary, in a general setting, not all of these features will be necessary for optimal classification. Only a selected number of significant features, when used with the classifier, can lead to optimal classification leading to distinguishing samples that belong to different classes. A large part of the remaining features are not too significant and could be either noise, irrelevant to the study or even redundant (Pirgazi *et al.*, 2019). The use of such insignificant features can lead to unwanted computational complexities and hamper the performance of the system. This is more pronounced when working with data having high dimensionality. Thus, it is essential to identify the set of significant features that can provide us with the optimal classification and clustering. For this to be accomplished, we need a robust method that can eliminate the redundant features and noise that do not have any information about the labels leaving us with only relevant features (Liu and Motoda, 2012).

Any dataset with N number of features has 2^N possible subset of features (Pirgazi *et al.*, 2019). In the presence of such a large number of possible combinations, finding the best subset of N features is computationally challenging and expensive (Liang *et al.*, 2018). An optimally selected set of features optimizes the performance of the models and also helps in alleviating the effect of *overfitting* and *high-dimensionality*. Along with the above benefits, selecting the appropriate features helps in easier interpretation of the model and thus its predictions. Also, the use of the gratuitous features can significantly impact the training speeds and the accuracy of the learning models. Overall, appropriate feature selection can provide the following advantages: (a) reduced cost for computation and storage, (b) adequate use of the available sample set for improved performance, (c) improved timing for classification and thus predictions, (d) easier interpretation of the data and thus the final predictions.

Filter, wrapper, embedded methods are the three general classes/types of feature selection techniques. Numerous algorithms have been proposed for each of these types of feature selection methods. The filtering method works by ranking the features using a statistical score that is assigned to each of them depending on their relevance to the class type. In both univariate and multivariate filter methods, the interactions among features are disregarded in the selection process. Studies like the ones in Pearson correlation coefficient(PCC), t-statistics(TS) (Speed, 2003), F-Test (Ding and Peng, 2005), and ANOVA (Ding and Li, 2015) are examples where the filtering method is used. It is observed that these methods are effective for selecting features in high-dimensional data because of the reduced computation expenses. However, they fail to provide good accuracy as discussed in (Sun *et al.*, 2018).

As an enhancement, the wrapper method is proposed with a learning algorithm and a classifier to find a suitable subset of features. Initially, a random solution is generated following which, an objective function is maximized using black-box optimization methods (Rau *et al.*, 2019) like simulated annealing (Jeong *et al.*, 2018), particle swarm optimization (Xue *et al.*, 2012), genetic algorithm (Wu *et al.*, 2011) and, ant colony optimization (Kabir *et al.*, 2012). The iterative evaluation of every candidate subset of the feature by the method leads to the identification of a strong relationship between features, however with an increase in the computational expense. Embedded methods are very efficient as they are a part of the learning phase. This helps reduce computation costs. Well-known example of the embedded method are LASSO (Tibshirani, 1996), recursive feature ellimation with state vector machine estimator (SVM-RFE) (Mamun and Mondal, 2019; Guyon *et al.*, 2002; Fang, 2019), random forest (Pouyan and Kostka, 2018; Ram *et al.*, 2017), Adabost (Wang, 2012), KNN (Le *et al.*, 2019), and autoencoder (Lu *et al.*, 2019).

In general, the use of feature selection is worthwhile when using the whole set of features is difficult or the cost of execution is high. As an example, the gene expression dataset contains more than 60 thousand features with a very low number of samples. Consequently, it is important to answer the impending questions: *Is it possible to identify important genes whose expressions can classify disease or cancer types?* Feature selection works differently compared to the standard dimension reduction techniques such as principal component analysis (PCA) (Hotelling, 1933), and autoencoders (Hinton and Salakhutdinov, 2006). The above-mentioned methods can preserve maximum variance with a highly reduced number of features. However, these methods do not provide the original features of the dataset making it difficult to eliminate redundant or irrelevant features from the dataset.

In this paper, we propose a novel feature subset selection method that increases the power of a standard deep autoencoder for differentiable feature selection. Our major contributions in this paper are two-fold: (a) we created a new variant of the standard autoencoder by introducing the use of a concrete relaxation discrete random variable selection layer for encoding that allows selection of user-defined number of original features, (b) classification of 33 different cancers by selecting features from cancer patients only, and (c) modifying the codebase of a standard autoencoder to realize the proposed approach. Our evaluation shows the improved performance of the proposed approach in comparison to the existing state-of-the-art methods, LASSO and SVM-RFE. Idea of concrete distribution is adapted from (Maddison *et al.*, 2016; Kingma and Welling, 2013), and reparameterization technique to minimize the loss and reconstruction error from (Abid *et al.*, 2019).

2 Materials and Methods

2.1 Coding and Non-coding Gene Expression

To validate the proposed idea, TCGA RNAseq cancer expression profiles and clinical data for 33 cancers (n=9566 cancer patients) were downloaded from UCSC Xena database (https://xenabrowser.net). This dataset contains expression profiles of around $60k$ RNAs including coding and non-coding (miRNA, lncRNA, etc). In this study, expression profiles of mRNA and lncRNA are considered for model evaluation. The number of mRNA and lncRNA are 18,731 and 12,309 respectively. Due to their varying expression levels, separate experiments were conducted for mRNA and lncRNA. This study was based on cancer patients only. So normal samples available in the same cancer are removed. The final dataset contains 9566 cancer patients as shown in Table 1. Each RNA expression was further processed using a min-max normalization method to achieve good training performance.

2.2 Concrete Relaxation Autoencoder

The concrete relaxation autoencoder (CoRAE) is a variation of the original autoencoder (AE) (Hinton and Salakhutdinov, 2006) which is used for dimension reduction. It is a neural network that consists of two parts: (a) an encoder that selects latent features and (b) a decoder that uses selected features to reconstruct an output that matches the input. Instead of using a sequence of fully connected layers in the encoder, we propose a concrete relaxation based feature selection layer where the user can define the number of nodes (features), k as shown in Figure 1. This layer selects a probabilistic linear arrangement of input features while training, which converges to a discrete set of k features by the end of training and during the testing.

The original features are selected based on the temperature of this layer which is tuned using an *annealing schedule*. More specifically, the concrete selector layer identifies k important features as the temperature decreases to zero. For reconstructing the input, a simple decoder similar to the ones associated with a standard AE is used. This simple neural network can be updated based on the characteristics of the data and its complexity.

The layer that selects the features and highlighted in Figure 1 is called concrete variable selctor layer adopted from the concrete distribution (Maddison *et al.*, 2016) and categorical representation (Jang *et al.*, 2016). Since backpropagation does not allow computation of the parameters' gradient through stochastic nodes of a standard autoencoder, Gumbel $softmax$ distribution g (Gumbel, 1954) is the right choice to pick samples f from categorical distribution with class probabilities α_k.

$$f = \textbf{one-hot}\left(\operatorname*{argmax}_{k}\left[g_k + log\,\alpha_k\right]\right) \tag{1}$$

Because argmax is not differentiable, a simple $softmax$ function can be used as a continuous approximation of argmax. The aim of using Concrete random variables is to relax the state of a discrete variable and the relaxation degree is controlled by a temperature parameter $\tau \in (0, \infty)$. To sample a concrete random variable in z dimension with parameter $\alpha \in \mathbb{R}^z > 0$ and τ, one must sample a z-dimensional vector of $i.i.d.$ (independent and identically distributed) samples from a Gumbel distribution, g. Then each element of the sample f from the Concrete

Table 1. Sample distribution for 33 cancers along with 75-25 split for training and testing.

Sl	Cancer site name	Acronym	#Sample	#Train	#Test	Sl	Cancer site name	Acronym	#Sample	#Train	#Test
1	Adrenocortical Cancer	ACC	77	57	20	18	Lung Squamous Cell Carcinoma	LUSC	498	373	125
2	Bladder Cancer	BLCA	407	305	102	19	Mesothelioma	MESO	86	64	22
3	Breast Cancer	BRCA	1089	816	273	20	Ovarian Cancer	OV	375	281	94
4	Cervical Cancer	CESC	304	228	76	21	Pancreatic Cancer	PAAD	177	132	45
5	Bile Duct Cancer	CHOL	36	27	9	22	Pheochromocytoma & Paraganglioma	PCPG	177	132	45
6	Colon Cancer	COAD	301	225	76	23	Prostate Cancer	PRAD	493	369	124
7	Large B-cell Lymphoma	DLBC	47	35	12	24	Rectal Cancer	READ	95	71	24
8	Esophageal Cancer	ESCA	161	120	41	25	Sarcoma	SARC	258	193	65
9	Glioblastoma	GBM	158	118	40	26	Melanoma	SKCM	465	348	117
10	Head and Neck Cancer	HNSC	499	374	125	27	Stomach Cancer	STAD	378	283	95
11	Kidney Chromophobe	KICH	66	49	17	28	Testicular Cancer	TGCT	132	99	33
12	Kidney Clear Cell Carcinoma	KIRC	527	395	132	29	Thyroid Cancer	THCA	501	375	126
13	Kidney Papillary Cell Carcinoma	KIRP	287	215	72	30	Thymoma	THYM	118	88	30
14	Acute Myeloid Leukemia	LAML	147	110	37	31	Endometrioid Cancer	UCEC	184	138	46
15	Lower Grade Glioma	LGG	507	380	127	32	Uterine Carcinosarcoma	UCS	56	42	14
16	Liver Cancer	LIHC	369	276	93	33	Ocular melanomas	UVM	79	59	20
17	Lung Adenocarcinoma	LUAD	512	384	128		Total		9566	7161	2405

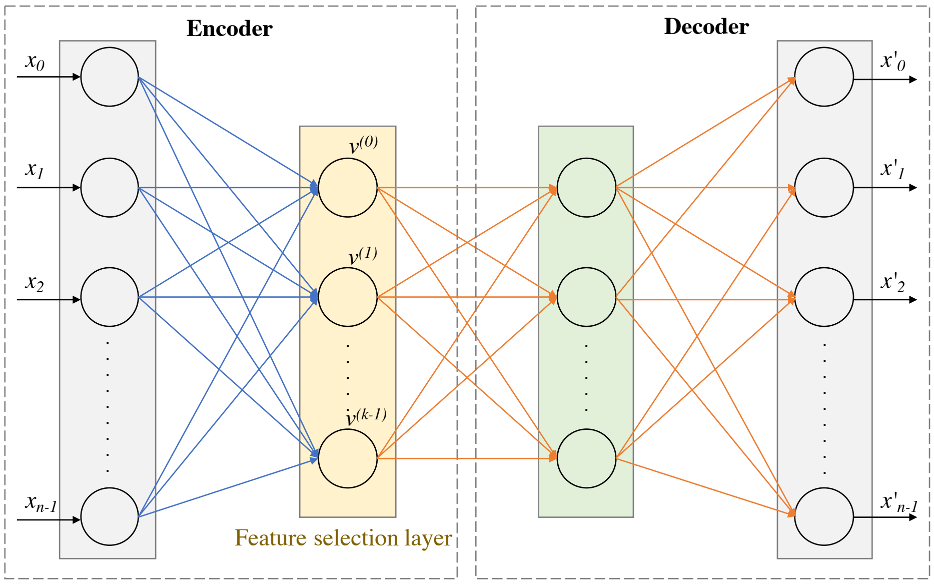

Fig. 1. Architecture of Concrete Relaxation Autoencoder. Proposed feature selection architecture consists of an encoder and a decoder. The layer after input layer in encoder is called concrete feature selection layer shown in yellow. This layer has k number of node where each node is for each feature to be selected. During the training stage, the i^{th} node $v^{(i)}$ takes the value $x^T f^{(i)}$. During testing stage, these weights are fixed and the element with the highest value is selected by the corresponding i^{th} hidden node. The architecture of the decoder remains the same during train and test stages.

distribution can be defined as:

$$f_k = \frac{exp((log\alpha_k + g_k)/\tau)}{\sum_{i=1}^{z} exp((log\alpha_i + g_i)/T)} \; for \; k = 1, ..., z \qquad (2)$$

where f_k refers to the k_{th} element in a particular sample vector. With the limit $\tau \rightarrow 0$, the concrete variable uniformly progresses the discrete distribution, producing one-hot vector with $f_k = 1$ with a probabilistic chance of $\alpha_k / \sum_p \alpha_p$. The advantage of using a concrete random discrete variable is that it is differentiable $w.r.t$ α using the reparameterization technique as mentioned by (Kingma and Welling, 2013).

More concisely, the way original feature is selected using the concrete random variable as follows: a z-dimensional concrete random variable $f^{(i)}$ is sampled for each node of the selector layer with k nodes where i refers to the index of the node, $i \in \{1...k\}$. The output of the i^{th} node is $\mathbf{x}.f^{(i)}$. Although it is a combination of the input feature's weight, every node of the selector layer produces exactly one of the original input features in the limit $\tau \rightarrow 0$. After training the network, a discrete argmax layer is replaced with the concrete selector layer by which $x_{\text{argmax}_j \; \alpha_j^{(i)}}$ is produced as an output of i^{th} node during the testing phase. The value of α_i initially starts with a small positive random number so that it can explore various combinations of input features. As the model is being trained, the value of α_i, in other words, the probability of class i becomes more stable. As a result, the model reduces its stochasticity rather increases the confidence in drawing a particular subset of features.

The temperature of the random variable in the selector layer has a significant impact in forming the output of each node. Initially, when τ is high, search space is large since it considers a linear combination of all features. In contrast, the selector layer will not be able to search all possible combinations of features in low τ and thus, model converges to a bad local minimum. Effect of changing the temperature in feature selection is shown in Figure 2. Instead of using a fixed temperature, a simple annealing scheduling scheme is used for every concrete variable. It starts with an user-defined high temperature (τ_s) and steadily lowers the temperature until it touches the ending bound (τ_e) by every epoch as follows:

$$\tau_{(e)} = \tau_s(\tau_N/\tau_s)^{e/N} \quad (3)$$

where $T_{(e)}$ is the temperature at epoch e, N refers to the total number of epochs. The proposed annealing schedule is good enough to explore the feature combinations during the training phase and finally lowered temperature enables the model to strict to the best set of features which is shown in Figure 3. The pseudocode of training a CoRAE is shown in Algorithm 1.

Algorithm 1: Concrete relaxation autoencoder

```
Input: $k$-number of feature to be selected, $X \in \mathbb{R}^{n \times z}$-Input,
$d$-decoder, $\theta$-number of learning rate, $N$-number of epoch,
$\tau_s$-starting temperature, and $\tau_f$-stopping temperature
Result: $k$ number of features
for $i \in \{1\ to\ z\}$ do
    Assign a positive value to each $\alpha^{(i)}$
for $e \in \{1\ to\ N\}$ do
    Let $\tau_e = \tau_s(\tau_N/\tau_s)^e/N$
    for $k \in \{1\ to\ z\}$ do
        sample $f^{(k)} \sim Concrete\ (\alpha^{(k)}, \tau)$
        let $X^{(k)} = X.f^{(k)}$, where $f^{(k)}$ = argmax($\alpha^{(k)}$)
```

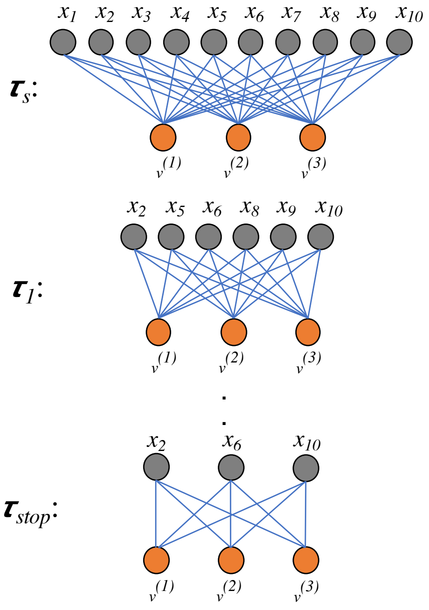

Fig. 2. Effect of temperature in reducing search space. For example, at starting temperature τ_s, the number of input features is 10 and the number of features to be selected k is 3, at the next epoch when the temperature is τ_1, the number of possible features reduces to 6. After some epochs, when the temperature reaches to its lower bound τ_{stop}, the number of features further reduces to 3 which is equal to k

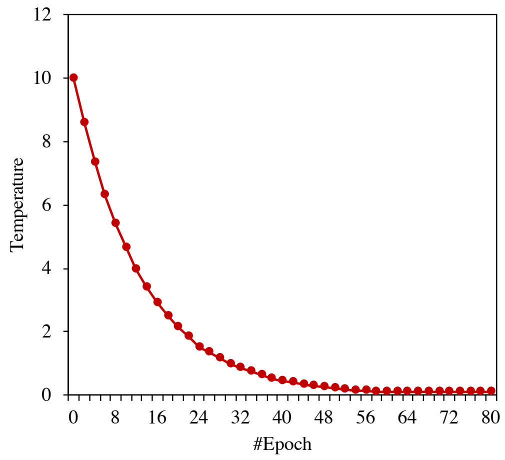

Fig. 3. Annealing schedules for the CoRAE. Effect of different annealing schedules on a concrete autoencoder trained on the mRNA dataset with $k = 100$ selected features. If the temperature is exponentially decayed (the annealing schedule), the feature selected layer (model) converges to informative features.

2.3 Gene Selection, Classification, Reconstruction, and Evaluation

The encoder of the Concrete relaxation autoencoder (CoRAE) architecture is constructed with a hidden layer of k nodes where k is the number of genes selected. The decoder, on the other hand, is consisting of one hidden layer with $3k/2$ nodes. The number of nodes in this layer is tuned in a range of $[4k/7, 2k/5, 3k/2]$. Adam optimizer with a learning rate of 0.001 is used for all the experiments. The starting temperature of the CoRAE was set to 10 and it ends at 0.01. To avoid overfitting, the dataset is split into the train and test set according to $75/25$ ratio. The training set is used to estimate the learning parameters and the test set is used for performance evaluation. To control the performance, the model is trained for the same number of epoch 100.

Performance of CoRAE has been compared with state-of-the-art feature selection techniques such as LASSO and SVM-RFE on both mRNA and lncRNA expression datasets. In LASSO, a regularization parameter α decides the number of most important features. More precisely, the higher the α, the more feature's coefficient shrinks to zero, and fewer features will be selected. Recursive feature elimination is a recursive method in which less important features are eliminated in every iteration. In the recursive feature elimination technique, SVM is used as an estimator. A linear kernel with a regularization parameter $C = 0.05$ is used. C controls the tradeoff between the error and norm of the learning weights. GridSearch algorithm is used to estimate the best set of parameters for SVM. In every iteration of RFE, the number of dropped features is set to 100.

We extract a subset of features by varying k from 10 to 500. For the comparison to be fair and along the same grounds with CoRAE, the same number of genes are selected using LASSO and SVM-RFE. The dataset with reduced number of features (expression of selected genes) is passed to the SVM for classifying 33 cancer types. Similarly, to reconstruct all the input features, we trained a linear regressor with no regularization and measure the reconstruction mean square error. LASSO and SVM-RFE are developed using the scikit-learn framework (Pedregosa *et al.*, 2011) whereas CoRAE is built using Google-developed Tensorflow (Abadi *et al.*, 2015) based deep learning framework Keras (Chollet *et al.*, 2015). Experiments are parallelized on NVIDIA Quadro K620 GPU with 384

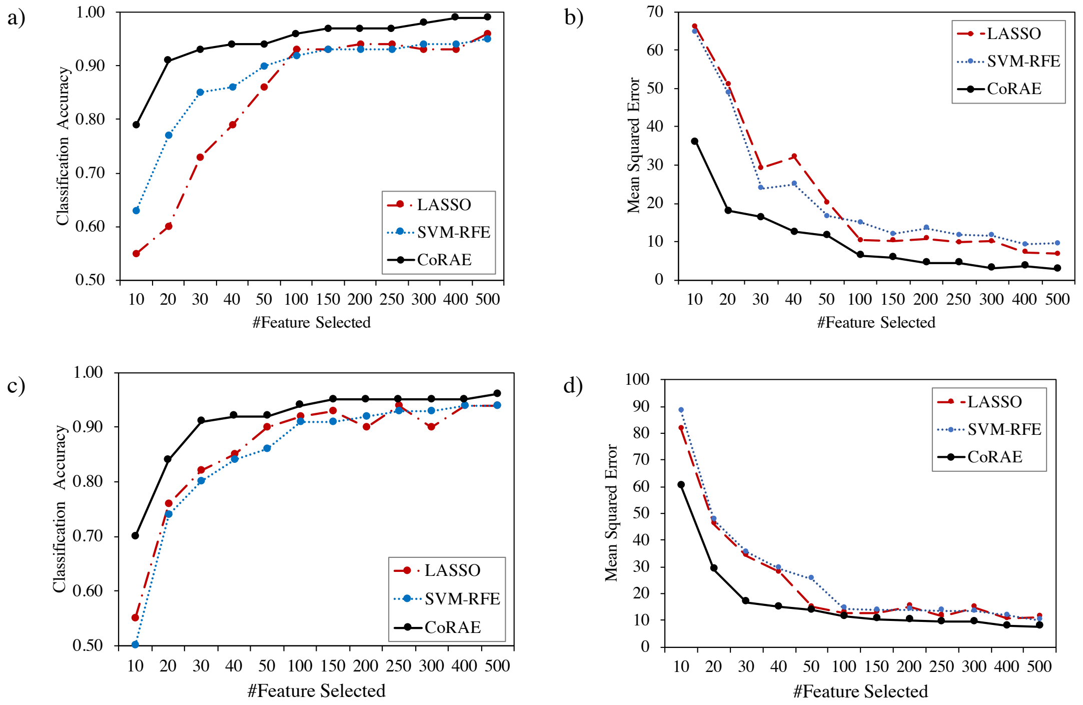

Fig. 4. Classification performance using selected RNA features. Comparison of CoRAE with other feature selection methods. Throughout the all values of k tested on both mRNA(a) and lncRNA(c) CoRAE have highest classification accuracy. Similarly, it shows lowest reconstruction mean squared error on both mRNA(b) and lncRNA(d)

cores and 2GB memory devices. Five different evaluation metrics have been used to record the classification and reconstruction performance such as accuracy, precision, recall, f1 score, and mean squared error (MSE).

Accuracy is the number of correct predictions made by the model over all kinds of predictions made. True positives(TP) and True Negatives(TN) are the correct predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

Precision is the number of correct positive results divided by the number of positive results predicted by the classifier. It indicates the predicted positive portion of the samples.

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

Recall is the number of correct positive results divided by the number of all relevant samples.

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

F1 score is a measure of a test's accuracy. It considers both the precision and the recall of the test to compute the score.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{7}$$

Mean squared error (MSE) is the average of $(\frac{1}{n}\sum_{i=1}^{n})$ of the square of the errors $(Y_i - Y_i')$ where Y_i is a true label and Y_i' is a predicted label. All performance metrics are measured on the predicted labels and true labels of independent test samples.

Top genes can be selected based on two criteria (a) classification accuracy needs to be higher, and (b) the number of genes should be as less as possible so that biologists can conduct a wet lab experiment easily. The capabilities of selected genes in pan-cancer classification is visually validated using unsupervised visualization technique t-SNE (Maaten and Hinton, 2008).

3 Results

A series of experiments were conducted to compare the performance of CoRAE with other state-of-the-art feature selection methods such as LASSO and SVM-RFE. Each of these three methods was used to select features in the range of 10 to 500. These features are then used to train a linear classifier SVM to classify 33 cancer types from expression profiles of coding and non-coding RNA separately. Table 2 and Figure 4 show classification performance of using selected features. It can be observed that the scale used in the x-axis to depict the number of features does not increase with a consistent step size. The initial stages of the experiments were performed with a smaller subset of the selected features as we wanted to understand the performance of the models being compared. The optimal classification performance was observed about 100 features were used. Beyond this, the trend continues as shown in Figure 4. We continued to monitor the performance of the models until 500 features to check for the best possible subset of features.

For all selected k number of features, as depicted in Figure 4, CoRAE has the highest accuracy and lowest error for both mRNA and lncRNA expression. It can be observed that even with a smaller number of significant features (say 10), the accuracy of CoRAE is close to 80% for mRNA whereas LASSO (55% accuracy) and SVM-RFE (63% accuracy) shows poor results for the same number of features. The trend remains

Table 2. Classification and reconstruction performances for different number of selected mRNAs and lncRNAs. Here, k is refere to the number of feature to be selected. Perforamnce evaluations for other k are shown in supplementary-1 Table 2

		mRNA					lncRNA				
#Feature selected, k	Method Name	Accuracy	Precision	Recall	F1	MSE	Accuracy	Precision	Recall	F1	MSE
10	LASSO	0.55	0.51	0.52	0.51	66.02	0.55	0.51	0.50	0.52	81.56
	SVM-RFE	0.63	0.6	0.53	0.53	64.69	0.50	0.40	0.36	0.35	88.19
	CoRAE	0.79	0.73	0.67	0.68	36.07	0.70	0.60	0.58	0.57	60.11
100	LASSO	0.93	0.89	0.89	0.89	10.4	0.92	0.90	0.90	0.89	12.83
	SVM-RFE	0.92	0.88	0.88	0.88	15.04	0.91	0.87	0.87	0.87	14.36
	CoRAE	0.96	0.95	0.95	0.95	6.40	0.94	0.93	0.91	0.92	11.37
500	LASSO	0.96	0.94	0.94	0.94	6.79	0.94	0.92	0.92	0.92	11.19
	SVM-RFE	0.95	0.93	0.93	0.93	9.47	0.94	0.92	0.91	0.91	10.01
	CoRAE	**0.99**	**0.98**	**0.98**	**0.98**	**2.86**	0.96	0.94	0.93	0.94	7.70

the same with the increase of number of features. With only 50 features, accuracy of CoRAE is more than 90%.

The CoRAE method is resilient to errors that occur during reconstruction using a small feature set. In comparison, this error is more pronounced in the other competing methods. In case of mRNA, CoRAE starts with an MSE of 38 and quickly reduces to a value of less than 10 within the use of the top 100 features as shown in Figure 4. The behavior in classification is highly comparable in both coding and non-coding genes. However, coding gene expressions perform slightly better than non-coding gene expression as shown in Figure 5.

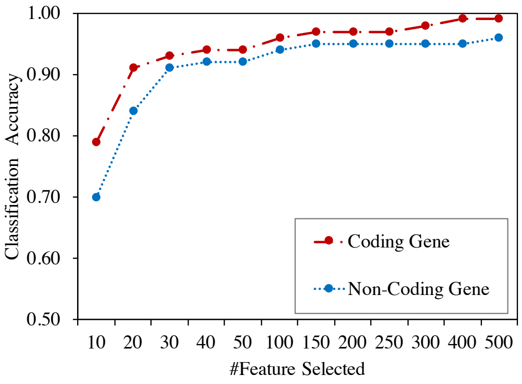

Fig. 5. Classification accuracy comparison between coding and non-coding genes expression. Across all the k, coding gene expressions show slightly better classification accuracy over non-coding gene expressions. Here, these features has been seelcted using proposed method only.

3.1 Selected Features Interpretation

With the use of CoRAE, we are able to identify important features while allowing the user to examine the relevance of each feature by observing the corresponding estimated concrete parameter $\alpha^{(i)}$. In CoRAE, feature selection is based on the value of vector α which gives the user the value of the importance score which in turn gives the power to identify features and based on their correlation with the other selected features. Figure 6, highlights how the top 100 mRNA or lncRNA, selected using the CoRAE, is capable of distinguishing 33 cancer types.

4 Discussion

In this paper, a new differentiable feature selection method called concrete relaxation autoencoder (CoRAE) is developed. It uses re-parameterization and concrete random variable technique to allow gradients to pass through a layer that stochastically selects discrete original input features of higher significance. The randomness of the proposed method enables it to effectively search and converge to a user-defined number of original features that maximize the objective function and minimize the loss as discussed in section 2.2. The estimated parameters learned by the models can be further examined by the biologists and other stakeholders to interpret biological relevance as discussed in section 3.1. The above-mentioned characteristics of CoRAE provide it with a distinction from numerous other competing approaches which are based on regularization.

Since CoRAE is built on top of a standard autoencoder architecture, it is easily scalable to a higher number of samples or dimensions. It is observed that the features selected by the CoRAE outperformed the ones from the competing methods. This paper accounts for a generalized approach of CoRAE. However, there are other avenues that can be explored using the proposed method. As an example, CoRAE can be used to extract important genes during the molecular subtype classification of a single cancer dataset, unlike the existing approaches which are based on the multiple cancer classification. The proposed method also provides the privilege to users to integrate multi-omics data such as gene, protein, RNAseq expression, DNA methylation, copy number and so on.

CoRAE is easy to use and requires modifying a few lines for implementing it in the popular machine learning frameworks. Moreover, the runtime and space complexity is similar to that of the standard autoencoder. In addition, it allows parallelization and enhances hardware acceleration which are obvious demands for deep learning techniques. Starting and ending temperature are the only added hyperparameters used for annealing schedule. The default values used in the experiment are carefully identified and is found to work adequately for the various datasets.

5 Conclusion

In this paper, we propose a novel feature subset selection method that increases the power of a standard deep autoencoder for differentiable feature selection. The proposed CoRAE is a new variant of the standard autoencoder which uses a concrete relaxation discrete random variable selection layer for encoding which allows selection of a user-defined number of original features by modifying the codebase of a standard

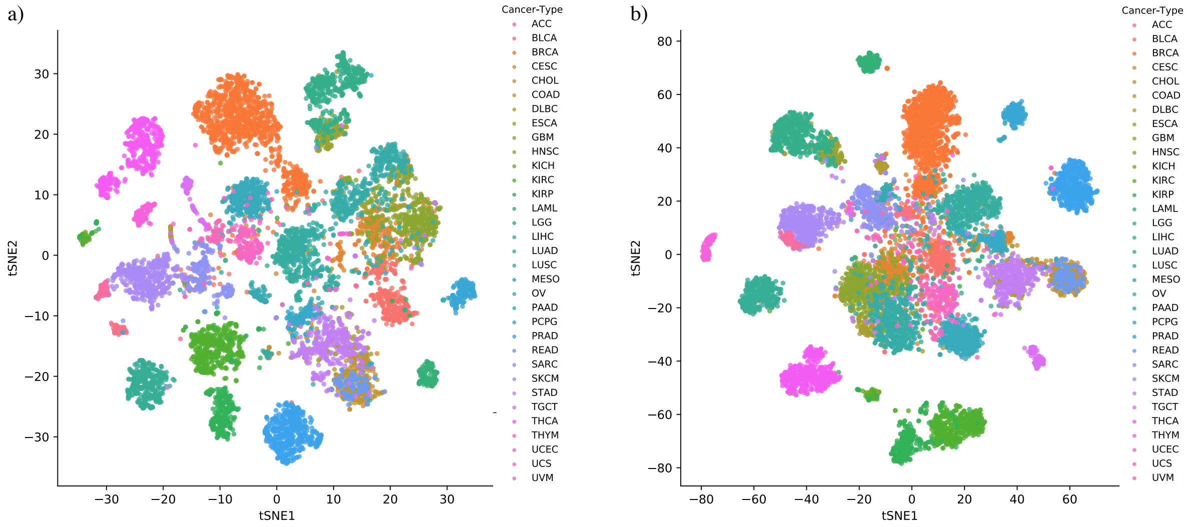

Fig. 6. Visualization of 33 different cancer types based on top-100 CoRAE features. Here, we show the t-SNE representation of 33 cancer samples using selected features. Each dot represents a cancer sample and each color represents a cancer type (a) t-SNE using top 100 mRNA, (b) t-SNE using top 100 lncRNA

autoencoder. We evaluate the performance of the proposed approach on coding and non-coding gene expression datasets and compare the results with state-of-the-art methods like LASSO and SVM-RFE. Our experiments show that on publicly available gene expression cancer datasets, CoRAE efficiently maximizes the classification accuracy and minimizes the reconstruction error using a selected subset of genes. For both mRNA and lncRNA gene expression datasets, CoRAE outperformed several sophisticated feature selection techniques. Use of a single hidden layer in the decoder, minimizes the reconstruction error and allows for selection of features from large datasets.

As a part of the future work, we will conduct more biological validations such as survival analysis of 33 cancer patients using selected features to measure the prognostic capabilities. Similarly, pathway analysis of selected coding and non-coding genes will be analyzed in the future.

Acknowledgements

We are grateful to Sanjeev Kaushik Ramani for helpful contribution in writing. This research is partially funded by the National Science Foundation CAREER award #1651917 (transferred to #1901628) to AMM.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Abid, A., Balin, M. F., and Zou, J. (2019). Concrete autoencoders for differentiable feature selection and reconstruction. *arXiv preprint arXiv:1901.09346*.

Chollet, F. *et al.* (2015). Keras. `https://github.com/fchollet/keras`.

Ding, C. and Peng, H. (2005). Minimum redundancy feature selection from microarray gene expression data. *Journal of bioinformatics and computational biology*, **3**(02), 185–205.

Ding, H. and Li, D. (2015). Identification of mitochondrial proteins of malaria parasite using analysis of variance. *Amino acids*, **47**(2), 329–333.

Fang, J. (2019). Tightly integrated genomic and epigenomic data mining using tensor decomposition. *Bioinformatics*, **35**(1), 112–118.

Gumbel, E. J. (1954). Statistical theory of extreme values and some practical applications. *NBS Applied Mathematics Series*, **33**.

Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine learning*, **46**(1-3), 389–422.

Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, **313**(5786), 504–507.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, **24**(6), 417.

Jang, E., Gu, S., and Poole, B. (2016). Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Jeong, I.-S., Kim, H.-K., Kim, T.-H., Lee, D. H., Kim, K. J., and Kang, S.-H. (2018). A feature selection approach based on simulated annealing for detecting various denial of service attacks. *Software Networking*, **2018**(1), 173–190.

Kabir, M. M., Shahjahan, M., and Murase, K. (2012). A new hybrid ant colony optimization algorithm for feature selection. *Expert Systems with Applications*, **39**(3), 3747–3763.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Le, T. T., Urbanowicz, R. J., Moore, J. H., and McKinney, B. A. (2019). Statistical inference relief (stir) feature selection. *Bioinformatics*, **35**(8), 1358–1365.

Liang, S., Ma, A., Yang, S., Wang, Y., and Ma, Q. (2018). A review of matched-pairs feature selection methods for gene expression data analysis. *Computational and structural biotechnology journal*, **16**, 88–97.

Liu, H. and Motoda, H. (2012). *Feature selection for knowledge discovery and data mining*, volume 454. Springer Science & Business Media.

Lu, X., Gu, H., Wang, Y., Wang, J., and Qin, P. (2019). Autoencoder based feature selection method for classification of anticancer drug response. *Frontiers in genetics*, **10**, 233.

Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, **9**(Nov), 2579–2605.

Maddison, C. J., Mnih, A., and Teh, Y. W. (2016). The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*.

Mamun, A. A. and Mondal, A. M. (2019). Feature selection and classification reveal key lncrnas for multiple cancers. *2019 IEEE International Conference on Bioinformatics and Biomedicine (IEEE BIBM)*, pages 2825–2831.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, **12**, 2825–2830.

Pirgazi, J., Alimoradi, M., Abharian, T. E., and Olyaee, M. H. (2019). An efficient hybrid filter-wrapper metaheuristic-based gene selection method for high dimensional datasets. *Scientific Reports*, **9**(1), 1–15.

Pouyan, M. B. and Kostka, D. (2018). Random forest based similarity learning for single cell rna sequencing data. *Bioinformatics*, **34**(13), i79–i88.

Ram, M., Najafi, A., and Shakeri, M. T. (2017). Classification and biomarker genes selection for cancer gene expression data using random forest. *Iranian journal of pathology*, **12**(4), 339.

Rau, A., Flister, M., Rui, H., and Auer, P. L. (2019). Exploring drivers of gene expression in the cancer genome atlas. *Bioinformatics*, **35**(1), 62–68.

Speed, T. (2003). *Statistical analysis of gene expression microarray data*. Chapman and Hall/CRC.

Sun, Y., Lu, C., and Li, X. (2018). The cross-entropy based multi-filter ensemble method for gene selection. *Genes*, **9**(5), 258.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, **58**(1), 267–288.

Wang, R. (2012). Adaboost for feature selection, classification and its relation with svm, a review. *Physics Procedia*, **25**, 800–807.

Wu, Y.-L., Tang, C.-Y., Hor, M.-K., and Wu, P.-F. (2011). Feature selection using genetic algorithm and cluster validation. *Expert Systems with Applications*, **38**(3), 2727–2732.

Xue, B., Zhang, M., and Browne, W. N. (2012). Particle swarm optimization for feature selection in classification: A multi-objective approach. *IEEE transactions on cybernetics*, **43**(6), 1656–1671.