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

Subject Section

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

Abdullah Al Mamun and Ananda Mondal

School of Computing and Information Sciences, Miami, US

*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 having high-dimension with a low number of samples in omic data, classification models encountered over-fitting problem. Therefore, there is a demand 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 useful features to increase the classification performance. However, these methods are unable to provide the original features. In this paper, we are introducing 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. During the training time, a predefined temperature of the feature selection layer is steadily decreased which allows the model to learn a user-specified number of discrete features. Also, during testing time, only selected features can be used to reconstruct the input in the decoder.
Results: We evaluated the proposed feature selection method on coding and non-coding gene expression of 33 cancer samples from TCGA where it significantly outperforms state-of-the-art methods in identifying top 100 coding and non-coding genes. Later, expression of selected genes is used to train a linear classifier to dishtinghish 33 cnacer types where features selected by CoRAE shows highest performance up to 99%. The proposed method can be implemented by adding a few lines of code to the standard autoencoder.

Availability: Source code and sample dataset can be found in https://github.com/pwaabdullah/MyPhD.git
Contact: amondal@fiu.edu
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

The well-known issue of recent omics data is a feature-sample ratio which is highly imbalanced means the number of features is way more than the number of samples. Among all the available features, few might be meaningful for distinguishing the samples which belong to different classes and the rest of these are either irrelevant, redundant, or noise (Pirgazi *et al.*, 2019). During classification or clustering the high dimensional data, irrelevant features cause unnecessary computational complexities and decrease the performance. Therefore, it is essential to identify the most relevant features that would have a high contribution to the classificaiton or clustering of the data. During the feature selection process, redundant features are removed because there is a subset of features that carries approximate similar information. In a similar fashion, noise features that provide no information about labels are also be removed from the database. Thus, only relevant features will be remain that will increase the efficiency of any classification or clustering problems Liu and Motoda (2012). Any dataset with N number of features has 2^N possible subset of features. The goal of feature selection algorithms is to find the most precise subset of features. Due to having a large number of possible combinations, finding the best subset of N features is computationally challenging and costly Liang *et al.* (2018).

Filter, wrapper, embedded are the types of feature selection methods. Numerous algorithms have been proposed for each type of feature selection method. In the filtering method, a rank is assigned to each feature depending on the statistical relevance to the class type. In both univariate and multivariate filter method, feature-feature interactions are not considered in the selection process. Some example studies such as Pearson correlation coefficient(PC), t-statistics(TS) Speed (2003), F-Test Ding and Peng (2005), and ANOVA Ding and Li (2015). These methods are effective for selecting features for high-dimensional data because of its fewer computation expenses but failed to provide a good accuracy Sun *et al.* (2018). To enhance the performance, the wrapper method is proposed with a learning algorithm and a classifier to find a suitable subset of features. First, it generates a random solution, then it maximize an objective function using a black-box optimization method Rau *et al.* (2019) such as 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). Since these methods evaluate every candidate subset of feature iteratively, they can find a strong relationship between features but it increases computational expenses. Similarly, embedded method do so efficiently as it is a part of its learning phase. Thus, it reduces the computational costs. Some well-known example studies are LASSO Tibshirani (1996), recursive feature ellimation with state vector machien estimator (SVM-RFE) Abdullah (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, feature selection methods are useful to get insight about large and complex dataset which can simplify the learning process of any machine learning algorithm. The use of feature selection is worthy when using the whole set of features is difficult to collect or costly to execute. For example, the gene expression dataset contains more than 60 thousand features with a very low number of samples. It is normal to ask: *Is it possible to identify important genes those expressions can classify available disease or cancer type?* The domain of feature selection is way more dissimilar than standard dimension reduction techniques such as principal component analysis (PCA) Hotelling (1933), and autoencoders Hinton and Salakhutdinov (2006). They can preserve maximum variance with a fewer number of features, however, these methods do not provide the original features of the dataset. Thus, it is impossible to eliminate redundant or irrelevant features from the dataset.

In this paper, a novel feature subset selection method that increases the power deep autoencoder for differentiable feature selection is proposed. Our method CoRAE introduces a new layer in the autoencoder called concrete distribution of features which allows the model to select a user-defined number of original features. 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). We have tested our end-to-end model on coding and non-coding gene expression dataset and it outperforms state-of-the-art feature selection techniques.

2 Materials and Methods

2.1 Coding and Non-coding Gene Expression

To validated the proposed idea, TCGA RNAseq cancer (n=9566) and clinical samples for 33 cancers were downloaded from UCSC Xena database (https://xenabrowser.net). TCGA processed raw RNAseq data using Illumina HiSeq 2000 RNA sequencing platform where per-gene normalized abundance estimation were calculated with FPKM method. RNASeq normalized counts were then log transformed after adding a constant of 1. Later UCSC re-processed using GENCODE v23 transcript annotation to quantify protein coding() and non-coding transcripts() expression Harrow *et al.* (2006). Coding genes refers to mRNA whereas non-coding genes refers to long non-coding RNA (lncRNA) in this experiment. To improve the focus on individual feature selection, we separated mRNA and lncRNA expression from combined database using TANRIC Li *et al.* (2015) provided standard list of lncRNAs. Another important reason of performing experiment on individual RNA types is because their expression level is different. The number of mRNA and lncRNA are 18731, 12309 respectively. We merged all the cancer samples for individual RNA types for further experiment. Each row is mapped to a unique Ensemble ID, and each column mapped to a patient ID. Normal patients or RNA with missing data were removed from the original dataset. Each RNA expression was further processed using min-max normalization method to achieve good training performance.

2.2 Concrete Relaxation Autoencoder

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

The way in which input features are combined depends on the temperature of this layer, which we modulate using a simple annealing schedule. As the temperature of the layer approaches zero, the layer selects k individual input features. The decoder of a CoRAE, which serves as the reconstruction function, is the same as that of a original AE: a neural network whose architecture can be set by the user based on dataset size and complexity. In effect, then, the CoRAE is a method for selecting a discrete set of features that are optimized for an arbitrarily-complex reconstruction function. We describe the ingredients for our method in more detail in the next two subsections.

Layer that selects the features shown in Fig. 1 is called concrete variable selctor layer adpoted from concrete distribution (Maddison *et al.*, 2016) and categorical represntation Jang *et al.* (2016). Since, backpropagation not allow computing the parameters gradient through stochastic nodes of standard autoencoder, gumbel $softmax$ distribution Gumbel (1954) is a right choice to pick samples z from categorical distribution with class probabilities α_k.

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

Because argmax is not differentiable, simple $soft - max$ 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 ralaxation degree is controlled by a temparature parameter $\tau \in (0, \infty)$. To sample a concrete random variable in z dimensions with prarameter $\alpha \in \mathbb{R}^z_{>0}$ and τ, one must samples a z-dimensional vector of i.i.d. samples from a Gumbel distribution, g. Then each element of the sample f from the Concrete distribution can be defined as:

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

where f_k refers to the k_{th} element in a particular sample vector. With the limit $\tau \to 0$, the concrete variable uniformly progresses the discrete distribution, producing one-hot vector with $l_k = 1$ with a probabilistic

Table 1. Data description

Sl	Cancer site name	Short	#Sample	#Train	#Test	Sl	Cancer site name	Short	#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

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 reparameterization technique 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 $l^{(i)}$ 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 $\mathbf{x}.l^{(i)}$. 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 $\tau \to 0$. After train the network, a discrete argmax layer is replaced with the concrete selector layer by which $x_{\mathrm{argmax}_j \ \alpha_j^{(i)}}$ 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, $alpha_i$ becomes more convincing. As a result, the model reduces its stochasticity rather increasing 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. The search space will be a linear combination of features when τ is high. 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 minima. 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 lessening the temperature until it touches the ending bound (τ_e) by every epoch as follows:

$$\tau(e) = \tau_s(\tau_N/\tau_s)^{e/N} \tag{3}$$

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 Fig 2.

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 being the number of gene 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 tunned in a range of [4k/7, 2k/5, 3k/2]. Adam optimizer with a learning rate of 10^-3 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, test set according to $75\% - 25\%$ ratio. The training set is used to estimates 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. To compare with other state-of-the-art methods mentioned in 1, features have been also selected using 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, fewer features would 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. Linear kernel with a regularization parameter $C = 0.05$ is used. C controls the tradeoff between the error and norm of the learning weights. Gride Search algorithm is tunned to estimates 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. Towards fair comparison with CoRAE, the same number of genes has been selected by LASSO and SVM-RFE. We then pass the reduced (expression of selected genes) dataset to an SVM to classify 33 cancer types on both mRNA and lncRNA expression. Similarly, to impute the original features, we trained a linear regressor with no regularization and measure the reconstruction mean square error. 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 prediction. LASSO and SVM-RFE are developed using scikit learn framwork Pedregosa *et al.* (2011) whereas CoRAE is developed using Google developed Tensorflow Abadi *et al.* (2015) based deep learning framwork Keras Chollet *et al.* (2015). Experiments are parallelized on NVIDIA Quadro K620 GPU with 384 cores and 2GB memory devices.

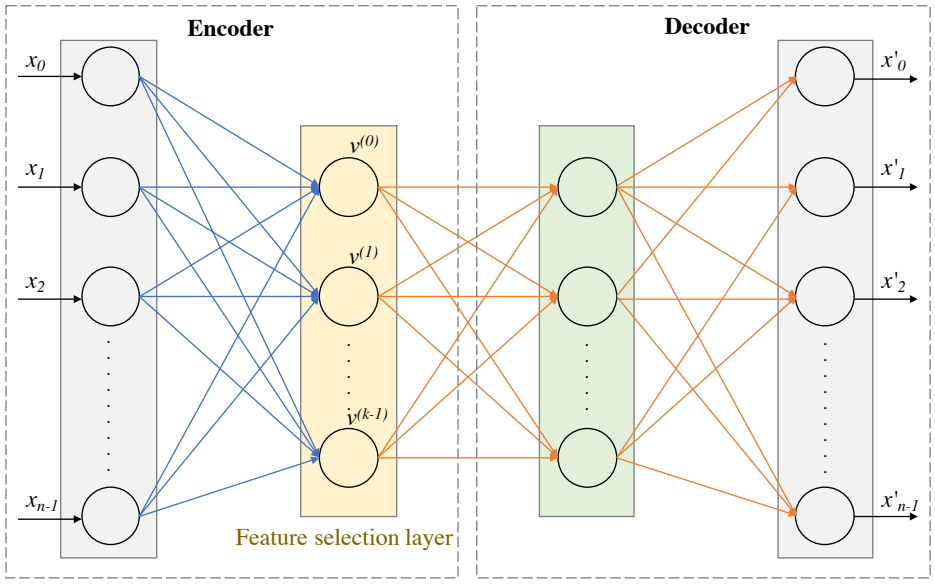

Fig. 1. Architecture of Concrete Relaxation Autoencoder. Proposed feature selection architecture is consists of a encoder and a decoder. The layer after input layer in encoder is called concrete feature selection layer shown in yellow. This layer has k number of node where each node is for each feature to be selected. During the training stage, the i^{th} node $u^{(i)}$ takes the value $x^T m^{(i)}$. 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 stage.

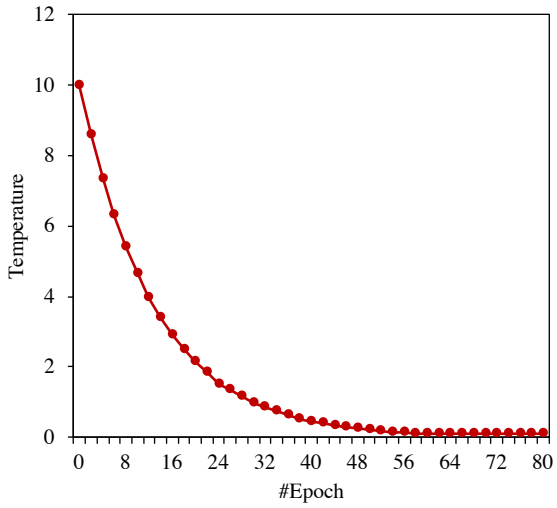

Fig. 2. Annealing schedules for the CoRAE. Here, we show the effect of different annealing schedules on a concrete autoencoder trained on the mRNA dataset with $k = 100$ selected features. At each epoch, we plot the temperature in brown. If the temperature is exponentially decayed (the annealing schedule we use), the samples converge to informative features.

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

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 avarage 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 matrices 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 experiements is conducted to compare the performance of CoRAE with other state-of-the-art feature selection methods such as LASSO and SVM-RFE. We have selected a range of features from 10 to 500 using all three methods then train a linear classifier (SVM) using selected coding and non-coding gene expression of 33 cancer patients. 3 shows the classification performance for different number of features. Across the all k, CoRAE has highest accuracy and lowest error for both mRNA and lncRNA expression. Even if the number of feature is low e.g. 10, the accuracy is almost 80% whereas LASSO and SVM-RFE shows poor restuls for lowest number of feature. For more than 50 features, all the methods shows above 90% accuracy. Also, CoRAE

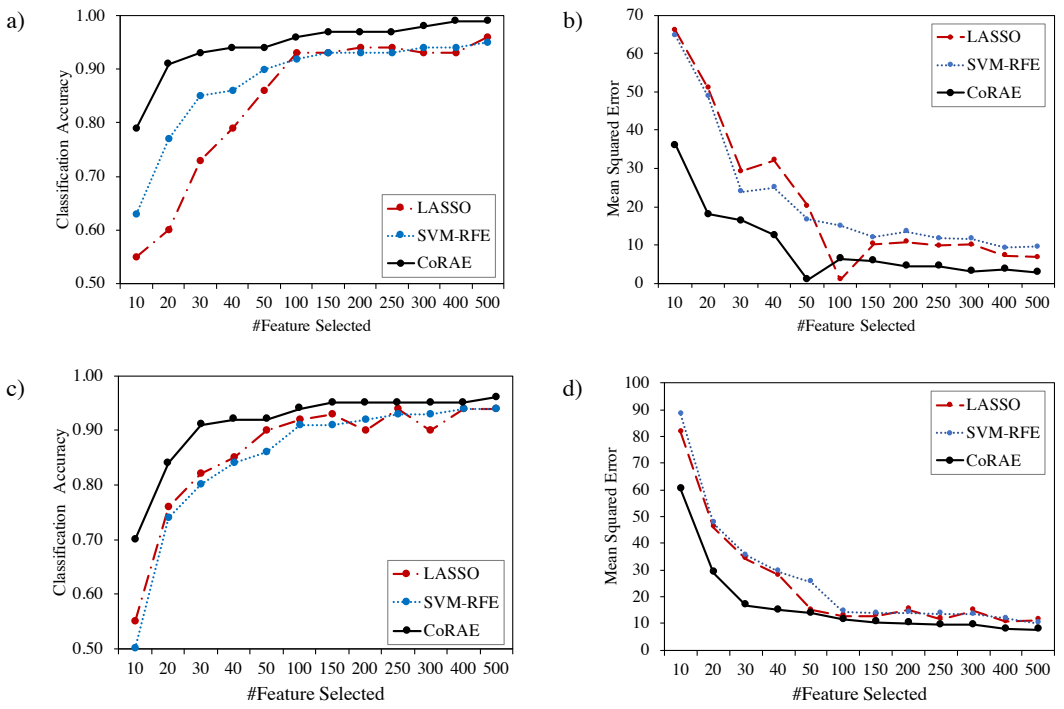

Fig. 3. Classification performance using selected RNA features. Comparison of CoRAE with other feature selection methods. Througout the all values of k tested on both mRNA(a) and lncRNA(c) CoRAE have highest classifcation accuracy. Similarly, it shows lowest reconstruction mean squared error on both mRNA(b) and lncRNA(d)

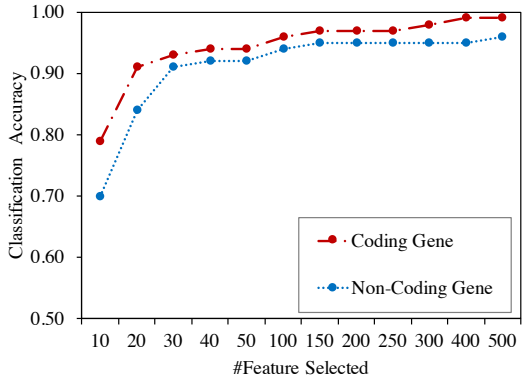

Fig. 4. Classification accuracy comparison between coding and non-coding genes expression. Across all the k, mRNA expression shows slightly better classification accuracy over lncRNA expression. Here, these features has been seelcted using proposed method only.

shows less error with less number of features. It starts from 38 and quickly reduced to less 10 within 50 features. The behaviour in classification is almost symetrical in both coding and non-coding genes. However, mRNA expression performs slightly better than lncRNA which is shown in 4.

3.1 Interpreting Related Features

CoRAE not only able to identify important features but also allows the user to examine relevance by observing the estimated concrete parameter $\alpha^{(i)}$ for each feature. Since CoRAE selects a feature based on the value of vector $\alpha^{(i)}$, the user can check the importance of each feature and find the correlation with others. In Fig. 5, it is visually revealed that the top 100 mRNA or lncRNA is capable of distinguishing 33 cancer types. Also, it is noticeable that the feature selected by CoRAE is carrying more information than among all other features. Thus, influential features are selected in proposed method.

4 Discussion

In this research, a new differentiable feature selection method via backpropagation is proposed. In brief, the concrete relaxation autoencoder used reparameterization and concrete random variable technique to allow gradients to pass through a layer that stochastically selects discrete original input features. This randomness of the proposed method enables it to effectively search and converge to a user-defined number of original features which maximizing the objective function and minimizing the loss as discussed in section 2.2. The estimated parameters learned by the models can be further examined by the biologists to interpret biological relevance as discussed in section 3.1. This made CoRAE distinctive from numerous competing approaches based on regularization.

It is shown via several experiments on publicly available gene expression cancer datasets that CoRAE efficiently maximizes the classification accuracy and minimizes the reconstruction error using a selected subset of genes. For both datasets mRNA and lncRNA gene expression, CoRAE outperformed several sophisticated feature selection techniques. This phenomena still remain and minimizes the reconstruction error when only a single hidden layer is used in the decoder. It indicates the power of CoRAE in selecting features from a large dataset.

Since CoRAE is built on standard autoencoder architecture, it is easily scalable to the higher number of samples or dimensions as discussed

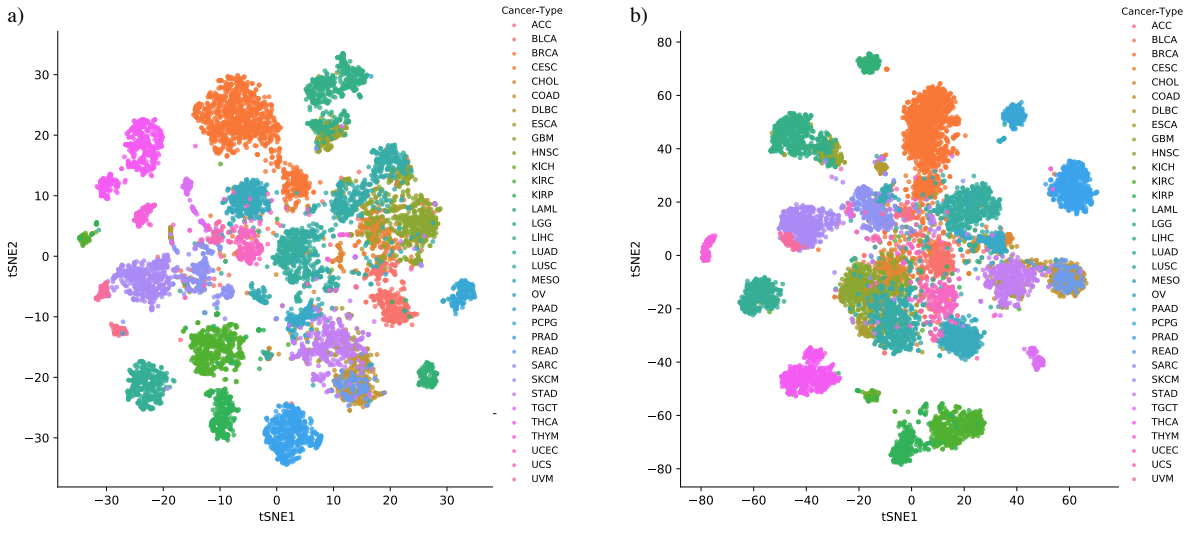

Fig. 5. Visualization of 33 different cancer types using top-100 CoRAE features. Here, we show the t-SNE representation of 33 cancer samples using selected features. Each dot represents a cancer sample and each color represents a cancer type. a) t-SNE using top 100 mRNA, b) t-SNE using top 100 lncRNA

in section **??** where the features selected by the CoRAE outperformed the competing methods. Moreover, as CoRAE proposed in its generic form, it can be surely prolonged in several fashions. For example, unlike multiple cancer classification, important genes can be extracted during the molecular subtype classification of a single cancer dataset. Also, It allows 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 it requires only a few lines of modification in implementing it in the popular machine learning algorithm. Moreover, the runtime and space complexity is similar to that of the standard autoencoder. In addition, it enhances parallelization and hardware acceleration which is an obvious demand for deep learning. Starting and ending temperature are the only added hyperparameters used for annealing schedule. The default value used in the experiment is found well enough for the various datasets.

5 Conclusion

future work: we will conduct more biological validation in our extended work 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 future work as well.

Acknowledgements

Funding

This research is partially funded by NSF CAREER award #1651917 (transferred to #1901628) to AMM.

References

Abadi, M. *et al.* (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Abdullah, Al Mamun, A. M. (2019). Feature selection and classification reveal key lncrnas for multiple cancers.

Abid, A. *et al.* (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. *et al.* (2002). Gene selection for cancer classification using support vector machines. *Machine learning*, **46**(1-3), 389–422.

Harrow, J. *et al.* (2006). Gencode: producing a reference annotation for encode. *Genome biology*, **7**(1), S4.

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. *et al.* (2016). Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Jeong, I.-S. *et al.* (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. *et al.* (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. *et al.* (2019). Statistical inference relief (stir) feature selection. *Bioinformatics*, **35**(8), 1358–1365.

Li, J. *et al.* (2015). Tanric: an interactive open platform to explore the function of lncrnas in cancer. *Cancer research*, **75**(18), 3728–3737.

Liang, S. *et al.* (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. *et al.* (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. *et al.* (2016). The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*.

Pedregosa, F. *et al.* (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, **12**, 2825–2830.

Pirgazi, J. *et al.* (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. *et al.* (2017). Classification and biomarker genes selection for cancer gene expression data using random forest. *Iranian journal of pathology*, **12**(4), 339.

Rau, A. *et al.* (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. *et al.* (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. *et al.* (2011). Feature selection using genetic algorithm and cluster validation. *Expert Systems with Applications*, **38**(3), 2727–2732.

Xue, B. *et al.* (2012). Particle swarm optimization for feature selection in classification: A multi-objective approach. *IEEE transactions on cybernetics*, **43**(6), 1656–1671.