Subgradient method-based SVMs for cancer classification with RNA sequencing data

CS 314: Distributed Data Management

Eric Zander

April, 2023

Abstract

Transcriptomics constitutes a valuable area of study for personalized medicine. As more and higher resolution RNA sequencing (RNA-seq) data becomes more accessible, so too can the value of deriving actionable insight via analysis. However, the high dimensionality of such data renders the identification and appraisal of clinically significant biomarkers difficult. Moreover, increases in the accessibility and resolution indicates potential growth in the volume and velocity with which RNA-seq data is generated. These attributes of transcriptome data in the present and going forward suggest the value of scalable and online machine learning techniques capable of reliably handling high dimensional data. Deep learning models optimized through stochastic and mini-batch gradient descent are promising in this regard, but they are also notoriously complex and challenging along the dimension of explainability. Support vector machines (SVM) stand out as powerful alternatives due to innate ability to adapt to high dimensionality attributable to the reliance on easily regularized decision boundaries for supervised learning tasks. Also, while quadratic programming approaches are common for SVM optimization, stochastic and mini-batch subgradient methods can also be employed in optimization. In this project, the groundwork for a scalable multi-class support vector machine amenable to online learning is offered in the form of a custom implementation in Python. This is tested on cancer classification problems with data from the Cancer Genome Atlas (TCGA) and Gene Expression Omnibus (GEO) with comparison to Scikit-learn's LinearSVC and SVC.

1 Introduction

Researchers have identified transcriptomics, or the study of RNA transcripts, as a promising avenue for development in medicine and elsewhere. One example of this concerns the classification of cancer or subtypes of cancer; research indicates that an understanding of the expression of the genetic instructions encoded in measurements of RNA biomarkers could support diagnosis and personalized treatment decisions.[1]

Notably, methods for gathering and processing RNA sequencing (RNA-seq) data are improving in resolution and scope. In addition to becoming more accessible, platforms have advanced to support sequencing at the single-cell level. This points to a potential growth in data volume and velocity that would correspond with an increasing value of scalable machine learning applications to transcriptome-based approaches to areas such as personalized medicine.

Given the high dimensionality of RNA-seq data and the potential value of scalable and online learning in particular, some machine learning (ML) models stand out as candidates for tasks such as classification. Certainly included is the ever-flexible deep learning framework. However, support vector machines (SVM) also stand out as potential alternatives to neural networks due to built-in regularization capable of mitigating overfitting associated with the curse of dimensionality. Notably, sequential minimal optimization (SMO), a popular quadratic programming approach for optimizing SVMs, is not particularly conducive to online learning and could limit scalability. However, SVMs can also be optimized via a subgradient method based on the dual formulation of the optimization problem. In other words, one can combine the applicability to online learning of a neural network with the powerful regularization of a SVM. This could provide an edge in supervised learning with RNA-seq data.

To explore subgradient methods for optimizing SVMs to classifying RNA-seq data, a custom SVM was implemented in Python. This implementation supports binary and multi-class classification, and was applied to RNA-seq data from the Cancer Genome Atlas (TCGA) project and Gene Expression Omnibus (GEO) database to support comparisons with Scikit-learn's [2] implementations of SVM classifiers (SVC).

2 Literature Review

Several ML approaches to cancer classification with RNA-seq data have been tested. These include nearest neighbor methods [3], linear discriminant analysis (LDA) [4], deep learning approaches [1][5], and more. Notably, the authors of [1] and [6] highlight SVMs for their performance in validation and general success with mid-size problems.

Gradient methods for optimizing according to the dual formulations of the SVM problem are described in [7] and [8], but primarily focus on the linear cases. Additionally, the referenced literature concerning applications of SVMs to RNA-seq classification primarily concern non-gradient-based optimization methods. In addition to more readily supporting online learning, using a subgradient method for optimization could support parallelization; as described in [7] and [9], multiple iterations of stochastic gradient descent may be performed in parallel and aggregated with relatively little intercommunication between nodes.

Dimensionality reduction also stands to offer much to the processing of wide data. The authors of [10] explore the performance of linear methods such as Principal Component Analysis (PCA), non-linear methods such as t-Distributed Stochastic Neighbor Embedding (t-SNE), deep learning-based methods like Variational Autoencoders (VAE), and more. The VAE and Universal Manifold Approximation and Projection (UMAP) techniques appear particularly promising. Feature selection is also an potentially valuable avenue to making such classification problems more manageable, but one of the theoretical benefits of a scalable and regularized ML approach is that the additional bias associated with selection may be more effectively mitigated.

2.1 Datasets

2.1.1 Old Faithful Geyser Data

For testing the implementation of a binary SVM classifier optimized through subgradient methods, a simple dataset describing eruptions from the Old Faithful geyser in Yellowstone National Park was classified with various settings. Two continuous features describing time between and duration of eruptions corresponded to a binary label describing the eruptions' types. This dataset was originally found in [11] and accessed through the Seaborn [12] Python library.

2.1.2 Palmer Archipelago Data

Early testing of a multi-class variant of the discussed SVM implementation was done with a dataset describing 3 species of penguins found on islands in the Palmer Archipelago, Antartica [13]. The set contains 344 samples and 4 continuous features describing bill length/depth, flipper length, and body mass. Island name and sex are also included, but were not used for implementation testing. Like the geyser dataset, this was accessed through Seaborn.

2.1.3 TCGA Pan-Cancer RNA-seq Data

Data from the Cancer Genome Atlas (TCGA) pan-cancer analysis project [14] accessed through the UCI machine learning repository [15] was leveraged in testing the performance of SVMs in classification with gene expression levels. This dataset consists of 801 samples with 20,531 gene expression levels each. Each has a label corresponding to one of 5 cancer types: lung adenocarcinoma (LUAD), breast carcinoma (BRCA), kidney renal clear-cell carcinoma (KIRC), colon adenocarcinoma (COAD), and prostatic adenocarcinoma (PRAD). The high dimensionality paired with relatively few samples are emblematic of the challenges innate to this type of dataset.

2.1.4 GEO Cancer RNA-seq Data

To further test the applicability and scalability of SVMs in classifying RNA-seq data, a relatively large dataset consisting of 35,240 gene expression levels for 15,554 samples was tested. Each sample corresponds with a label corresponding to one of 8 cancer types: colorectal, prostate, breast, lung, gastric, pancreatic, ovarian, and kidney. The data from many studies is made available on the Gene Expression Omnibus (GEO) [16] platform, but this set was collected through the ARCHS4 platform [17].

Human gene expression files were downloaded through ARCHS4 on March 15, 2023. All samples with metadata pertaining to cancer were selected before taking only samples with labels describing one of the more common types as mentioned previously. This resulted in 35,240 samples describing 8 types of cancer. In addition to offering more features and samples, this data serves to provide an additional challenge due to a relative lack of uniform processing compared to the TCGA data available through the UCI repository.

2.2 Methods

2.2.1 SVM Optimization

Optimization of the dual formulation of the SVM problem involves maximizing the margin subject to additional constraints encoded in a Langrangian. This formulation supports the kernel trick by which the decision boundary optimized to distinguish class may be computed using alternate representations of data in higher dimensions. Commonly employed kernels include the polynomial or Gaussian radial basis function (RBF) kernels, and these are implemented for this project.

The Lagrangian representing the dual formulation of the SVM for use in gradient descent can be written as:

$$L(\alpha) = -\frac{1}{2}\sum_{i} \alpha_i \alpha_j y_i y_j K(x_i, x_j) + \sum_{i} \alpha_i$$

with Lagrange multipliers α, labels y, and inputs x transformed by the kernel function $K(x_i, x_j)$. To maximize the margin, one can minimize L by finding optimal values for α subject to the constraint $0 <= \alpha_i <= C$ where C is a parameter for regularization. To find the optimal Lagrange multipliers α, one can use the subgradient method with the following partial derivative:

$$\frac{\delta L}{\delta \alpha_i} = 1 - \frac{1}{2}\sum_{j} \alpha_j y_i y_j K(x_i, x_j)$$

With optimal values of α where non-zero values are associated with support vectors, the bias term b and decision function $f(x)$ are:

$$b = y_i - \sum_j \alpha_j y_j K(x_i, x_j)$$

$$f(x) = sign(\sum_i \alpha_i y_i K(x, x_i) + b)$$

Note that the sign can be dropped from the decision function to get distances from the decision boundary that can be compared for multiple binary classifiers in multi-class classification. For this project, one-vs-rest (OvR) multi-class classification was implemented by training a binary classifier for each class and choosing the label associated with the largest distance.

The functions for the kernels K supported by this implementation for input matrices X are as follows where γ is a tunable parameter and d is the degree of the polynomial:

$$K(X_i, X_j)_{linear} = X_i^T X_j$$

$$K(X_i, X_j)_{polynomial} = (\gamma X_i^T X_j)^d$$

$$K(X_i, X_j)_{rbf} = e^{-\gamma ||X_i - X_j||^2}$$

Throughout training, shrinking is implemented to stabilize learning and encourage a convergence with fewer support vectors. α_i is initialized as non-zero and is set permanently to zero if falls low enough based on the premise that if a Lagrange multiplier is unlikely to be associated with a valuable support vector later if it gets zero'd out during gradient descent.[8].

This approach is encoded in a custom model implemented as the foundation for a parallelizable SVM capable of online learning. The implementation uses Python and NumPy[18], and supports stochastic and mini-batch learning where convergence is determined by a lack in change between Lagrange multipliers per epoch. For these experiments, min-max scaling was applied to all datasets to render features more comparable.

2.2.2 Dimensionality Reduction

Several methods were implemented and tested to reduce the dimensionality of data prior to fitting.

The first was PCA. This is a powerful method for feature reduction that commonly leverages singular value decomposition (SVD) with the dataset's covariance matrix to obtain eigenvectors that capture, to declining degrees, the variance of the dataset. Data may then be expressed in a limited number of these principal components to effectively perform lossy compression. However, the goal of scalability limits PCA's usefulness due to the requirement of covariance matrix computation.

Another approach involves random projection. This is a relatively simple technique that entails creating a matrix of random values to consistently project data into a form with fewer dimensions while preserving aspects of the original data. It is computationally efficient and may be parallelized, but can be less effective than other forms of dimensionality reduction. A custom implementation of Gaussian random projection was created for this project, but Scikit-learn's implementation of Sparse random projection (SparseRandomProjection) was also tested.

Finally, some experimentation was done with custom implementations of random Fourier features (RFF) and CUR in the context of data dimensionality reduction. However, RFF would be more appropriate for kernel approximation and the potential benefits of CUR in the realm of scalability were not realized here.

2.2.3 Kernel Approximation

Scalable learning with nonlinear kernels can be challenging. High dimensional data can correspond with a need for a sufficient number of samples to accurately capture relationships, but the requirement of the computation of pairwise distances for the RBF kernel in particular limits the number of samples that may be feasibly used at once. Consequently, Nyström approximation was implemented as one approach to help ameliorate issues related to kernel scalability.

This technique involves sampling the data and creating a low-rank approximation of the original matrix in the form of a smaller submatrix that can then be used in developing the kernel. For this implementation, uniform sampling of a parameterized number of rows was used due to both simplicity and success as described in [19].

3 Experimental Results

3.1 Implementation Testing

The implementation of the binary and multi-class forms of the classifier were built while testing on the geyser and penguin datasets. SVMs with polynomial and RBF kernels were capable of perfect classification of the geyser dataset, and all kernels were capable of obtaining the same for the penguin dataset. See Figure 1 for an example illustrating the behavior of different kernels.

3.2 TCGA Data Classification

5-fold cross validation with each kernel reveals some of the strengths and weaknesses of a stochastic or mini-batch gradient-based optimization method. As seen in Figure 2, the custom implementation affords faster execution types at the price of a slight decline in accuracy for the linear and polynomial kernels. However, convergence proves significantly slower and inferior for the RBF kernel.

Easy convergence affords excellent results in great time, but the gradient descent method can be noisier and encounter issues with finding local minima. These issues are exacerbated by the use of nonlinear kernels: results shown for classification with the RBF kernel required far more hyperparameter tuning and employment of the dimensionality reduction methods discussed before

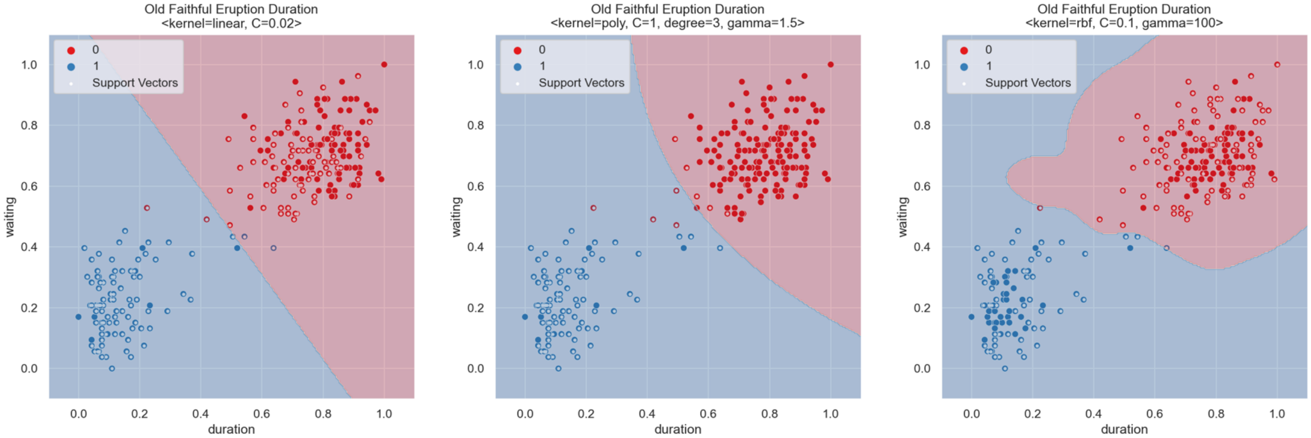

Figure 1: Plots of SVM decision boundaries for the geyser dataset. High values of gamma were selected to exaggerate the differences in kernel behavior for illustrative purposes.

only to achieve markedly worse scores compared to the linear and polynomial kernels. Additional metrics for 5-fold cross validation with the linear classifier are shown in Figure 3.

3.3 GEO Data Classification

The jump up in the scale and relative drop in uniformity of the data corresponded with an exaggeration of differences between the Scikit-learn and custom implementations. Both the implemented gradient-based and more standard SVMs afforded slow and resource-intensive convergence with the polynomial and RBF kernels; the linear kernel proved more reliable.

Several dimensionality reduction methods were tested here to render the dataset more manageable, but Scikit-learn's sparse random projection provided the best balance of computational efficiency and scores out of those which were tested. However, the linear kernel was reasonably tested in lieu of any dimensionality reduction with the following results.

The best validation accuracy of the custom implementation following hyperparameter selection was 0.9550 while the score for the application of Scikit-learn's LinearSVC was 0.9781.

Execution time was largely comparable, but the current custom implementation performs Nyström approximation for each binary classifier; alternative approaches to kernel computation or approximation could be of extreme benefit for in the multiclass setting in particular.

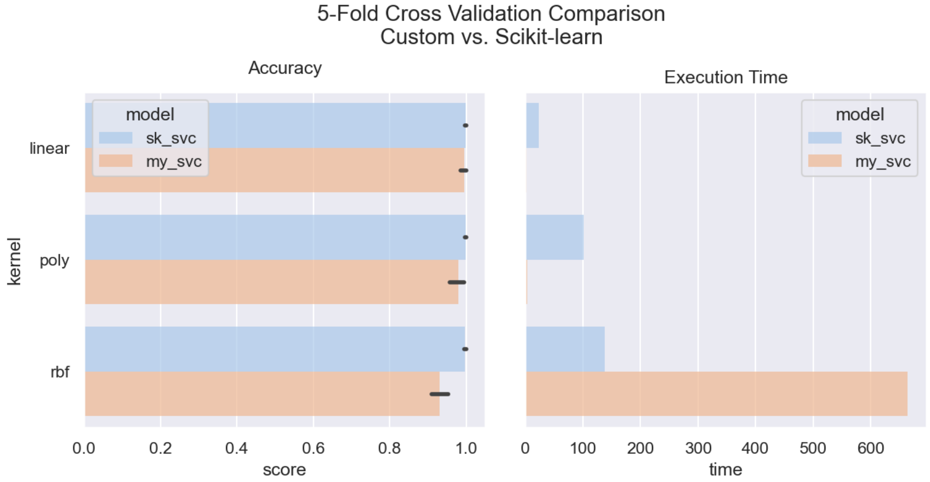

Figure 2: Comparison of accuracy and execution time in 5-fold cross validation between the custom SVM implemented for this project and Scikit-learn's LinearSVC/non-linear SVC.

4 Conclusion

The results with both the custom implementation and Scikit-learn's support vector classifiers indicate the value of regularized decision boundary-based classification of high dimensional RNA sequencing data. Easily achieved accuracies over 95% for multi-class problems involving tens of thousands of expression markers are promising. However, further comparison to the performances of other classifiers is necessary to make proper conclusions as to widescale applicability.

More significantly to the goal of the project, a fairly performant SVM trained with a subgradient method was successfully tested on RNA-seq data.

It is possible that additional and superior methods of scalable kernel approximation and data preprocessing could help in successfully applying the otherwise powerful RBF kernel in particular. There are also numerous other methods, deep learning-based and otherwise, that could be employed for dimensionality reduction. VAEs and UMAP stand out as scalable and performant alternatives to those tried here.[10]

Additionally, the low number of samples in some RNA-seq dataset may justify the employment of transductive SVMs.[20] The decision boundary-based approach is amenable to transduction due to the ability to massage hyperplanes with new testing data, and for small unbalanced datasets concerning topics such as rare diseases this may prove quite powerful.

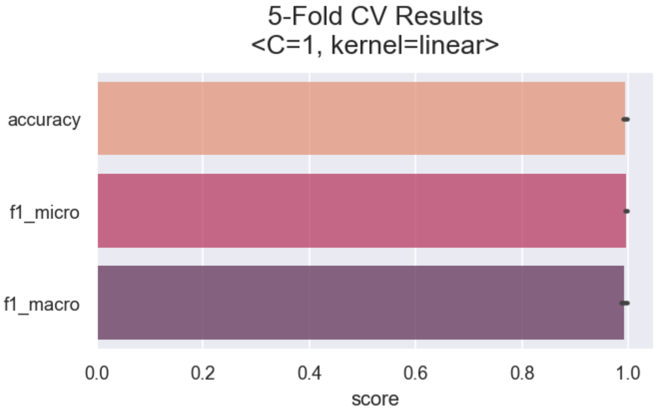

Figure 3: Mean accuracy and both micro- and macro-weighted F1-score for classification with the linear kernel and minimal hyperparameter tuning. These are 0.9938, 0.9928, and 0.9963 respectively.

References

[1] F. Carrillo-Perez *et al.*, "Non-small-cell lung cancer classification via RNA-Seq and histology imaging probability fusion," *BMC Bioinformatics*, vol. 22, no. 1, p. 454, Sep. 2021.

[2] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[3] K. F. Mahin *et al.*, "PanClassif: Improving pan cancer classification of single cell RNA-seq gene expression data using machine learning," en, *Genomics*, vol. 114, no. 2, p. 110 264, Jan. 2022.

[4] D. Goksuluk *et al.*, "MLSeq: Machine learning interface for RNA-sequencing data," en, *Comput Methods Programs Biomed*, vol. 175, pp. 223–231, Apr. 2019.

[5] L. Rukhsar *et al.*, "Analyzing rna-seq gene expression data using deep learning approaches for cancer classification," *Applied Sciences*, vol. 12, no. 4, 2022, ISSN: 2076-3417. DOI: 10.3390/app12041850. [Online]. Available: https://www.mdpi.com/2076-3417/12/4/1850.

[6] J. Alquicira-Hernandez *et al.*, "Scpred: Accurate supervised method for cell-type classification from single-cell RNA-seq data," *Genome Biology*, vol. 20, no. 1, p. 264, Dec. 2019.

[7] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of Massive Datasets*, 2nd. USA: Cambridge University Press, 2014, ISBN: 1107077230.

[8] C.-J. Hsieh *et al.*, "A dual coordinate descent method for large-scale linear svm," 2008. DOI: 10.1145/1390156.1390208.

[9] R. K. L. Kennedy *et al.*, "A parallel and distributed stochastic gradient descent implementation using commodity clusters," *Journal of Big Data*, vol. 6, no. 1, p. 16, Feb. 2019.

[10] R. Xiang *et al.*, "A comparison for dimensionality reduction methods of single-cell rna-seq data," *Frontiers in Genetics*, vol. 12, 2021, ISSN: 1664-8021. DOI: `10.3389/fgene.2021.646936`. [Online]. Available: `https://www.frontiersin.org/articles/10.3389/fgene.2021.646936`.

[11] W. K. Härdle, *Smoothing Techniques: With Implementation in S*. Springer, 1991.

[12] M. L. Waskom, "Seaborn: Statistical data visualization," *Journal of Open Source Software*, vol. 6, no. 60, p. 3021, 2021. DOI: `10.21105/joss.03021`. [Online]. Available: `https://doi.org/10.21105/joss.03021`.

[13] K. B. Gorman, T. D. Williams, and W. R. Fraser, "Ecological sexual dimorphism and environmental variability within a community of antarctic penguins (genus pygoscelis)," *PLoS ONE*, vol. 9(3), no. e90081, 2014. [Online]. Available: `https://doi.org/10.1371/journal.pone.0090081`.

[14] J. N. Weinstein *et al.*, "The cancer genome atlas Pan-Cancer analysis project," en, *Nat. Genet.*, vol. 45, no. 10, pp. 1113–1120, Oct. 2013.

[15] D. Dua and C. Graff, *UCI machine learning repository*, 2017. [Online]. Available: `http://archive.ics.uci.edu/ml`.

[16] T. Barrett *et al.*, "NCBI GEO: Archive for functional genomics data sets—update," en, *Nucleic Acids Res.*, vol. 41, no. D1, pp. D991–D995, Nov. 2012.

[17] A. Lachmann *et al.*, "Massive mining of publicly available RNA-seq data from human and mouse," en, *Nat. Commun.*, vol. 9, no. 1, Apr. 2018.

[18] C. R. Harris *et al.*, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. DOI: `10.1038/s41586-020-2649-2`. [Online]. Available: `https://doi.org/10.1038/s41586-020-2649-2`.

[19] S. Kumar, M. Mohri, and A. Talwalkar, "Sampling techniques for the nystrom method," in *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, vol. 5, PMLR, 2009, pp. 304–311. [Online]. Available: `https://proceedings.mlr.press/v5/kumar09a.html`.

[20] T. Joachims, "Transductive inference for text classification using support vector machines," in *Proceedings of the Sixteenth International Conference on Machine Learning*, ser. ICML '99, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999, pp. 200–209, ISBN: 1558606122.