Quantum Support Vector Machine for Type 1 Diabetes Diagnosis  
Kylee Shiekh  
PHGN 519  
3 October 2022

Abstract and Motivation

Type 1 Diabetes is a chronic condition in which the pancreas produces little to no insulin, thus making the regulation of sugar difficult without assistance. Symptoms commonly associated with Type 1 are excessive thirst or urination, hunger, and fatigue. These symptoms can be indicative of a variety of diseases, but together raise alarms about Type 1 Diabetes. This project was inspired by a peer in my undergraduate program who had Type 1 Diabetes, and together we created classical machine learning algorithms to try to diagnose diabetes with these symptoms. Using a quantum support vector machine, this project will attempt to improve those results, and focus on the frequency of false positives and false negatives rather than the validation score.

This work was partially inspired by Joseph Geraci, a panelist at IEEE Quantum Week. His company works in clinical trials for ALS and uses quantum machine learning to distinguish symptoms and group patients together. I am fascinated with health care, and this is a perfect intersection between my skills in machine learning, developing my quantum knowledge, and entering a field I’m interested in.

Introduction

In 2021, myself, Nathan Millwater, and Victoria Messmore developed multiple models for predicting Type 1 diabetes based on symptom data and picked the best performing one. The results were not replicable using a survey of Mines students. In both data sets the model predicted True Positives and True Negatives appropriately but struggled to identify patients who had some symptoms but did not have diabetes.

I will be working with two data sets. The first data set is from the original project, obtained from Kaggle [1], which included symptoms and demographic data: Age, Gender, Sudden weight loss, Weakness, Itching, Irritability, Delayed Healing, and more. There are 520 samples (patients) represented in the set. The second data set looks at qualitative medical tests: Glucose level, Blood pressure, skin thickness, insulin level, BMI, and Diabetes Pedigree Function [2].

Support vector machines are supervised learning models used for classification and regression analysis. The model creates a hyperplane, a line which will separate the groups we’re classifying when displayed on a feature vs feature graph. This line is not necessarily linear but can be squiggly to maximize the largest separation between groups. The shape of this line depends on the kernel function selected to suit the problem, which can have a significant impact on accuracy.

Often our hyperplane is not easily linear, but to still be able to work with a linear SVM we move our data into a higher space. We map the original features to a higher transformer space, where we can obtain a set of weights corresponding to the boundary hyperplane. Then we transform the hyperplane back into the original 2D space to obtain a non linear boundary. [3] The kernel function performs this mapping more efficiently than using a feature map.

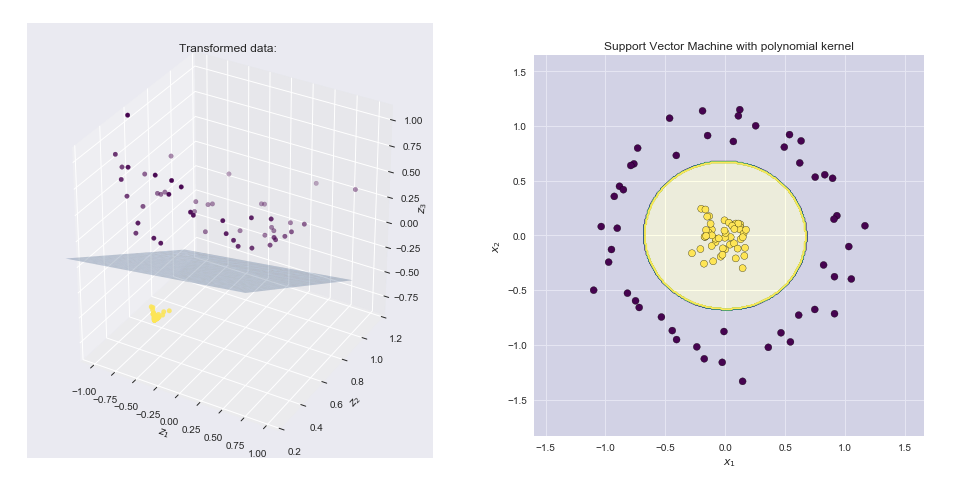


Figure : Left plot shows the points plotted in the transformed space with the SVM linear boundary hyperplane. Right shows the results of the original 2D space.[3]

To conduct supervised classification on a quantum computer we must do the following [4]: translate the classical data point X into a quantum datapoint |Φ(𝑥⃗)⟩. This can be achieved by a circuit V(Φ(𝑥⃗)). Where Φ(…) could be any classical function applied on the classical data 𝑥⃗. Then we need a parameterized quantum circuit W(Θ) with parameters Θ that processes the data in a way that in the end we can apply a measurement that returns a classical value -1 or 1 for each classical input 𝑥⃗ that identifies the label of the classical data.

The quantum kernel has the same idea as the classical case. We take the inner product of the feature maps 𝐾(𝑥⃗,𝑧⃗)=|⟨Φ(𝑥⃗)|Φ(𝑧⃗)⟩|², but now with the quantum feature maps [4]. If we choose a quantum feature map that is not easy to simulate with a classical computer we might be able to achieve an advantage. There is no proof yet that the QSVM brings a quantum advantage, but there is for sure no advantage if we use feature maps that are easy to simulate classically [5].

The Model

The SVM involves specifying a kernel, which is way of computing the dot product of two vectors. I determined the kernel for the classical models by testing a range of parameters to find the one that maximized the accuracy of the models. I adjusted the parameters C and gamma. C is the regularization parameter, which controls the fit of the model. Gamma is the kernel coefficient for polynomial kernels, and this is our primary control over the kernel function.

In a quantum SVM, the kernel is found from the ZZ feature map. The ZZ feature map is the second-order Pauli-Z evolution circuit below. The QuantumKernel class in qiskit calculates a kernel function from a feature map.

A screenshot of a computer

Description automatically generated with low confidence

Figure : For 3 qubits and 1 repetition and linear entanglement the 2nd order Pauli-Z evolution circuit is represented by [6]

The kernel is recalculated for the testing and training data sets and the following matrices are calculated. These matrices are essentially heat maps to show the hyperplane lines. These are generated by Qiskit using a different data set than my own. The diabetes data is currently unsuited to Qiskit’s quantum SVM.

Chart, scatter chart

Description automatically generated

Figure : Plots of the training and testing kernel matrices from Qiskit SVC code

One main limitation of this project is that the quantum SVM was not replicable for the diabetes data. When reduced to a much smaller data set, 25 patients rather than 500, we are able to obtain the following kernel matrices but unable to test the classification score or fit the QSVM. Quantum machine learning is thought to be applicable to health care because of the need to use large data sets, but that hasn’t been shown in this project.

Chart, scatter chart

Description automatically generated

Figure : Plots of the training and testing kernel matrices from diabetes SVC code

The Results

Using a classical SVM, neither data set achieved an average validation score of more than 66%. The validation score indicates the models ability to generalize, and these models did not fit their test set well. This could be indicative of overfitting. When viewing a graph of the predicted labels (orange) vs actual labels (blue) of data, where predicted labels are orange and actual labels are blue dots, we can see that the model is poor at predicting positive patients.

Chart

Description automatically generated with medium confidenceA picture containing text

Description automatically generated

This poor performance is why I chose to explore a quantum advantage. The quantum-SVM model was tested on an ad-hoc data set and achieved an average accuracy of 100%. Ad-hoc data is generated such that labels are between -1 and 1, which our data suits, where training and testing data are separated [7]. While it seems unrealistic to get an accuracy of 100%, this result is also verified by Havlicek et al [7]. This however requires at least 150 trial steps, displayed in the figure below

Chart, line chart

Description automatically generated

Conclusions and Future Work

While a classical SVM was easy to make and test, the QSVM did not live up to this. There is a confirmed advantage, but this is from 6 year old research and progress moves fast in this field! For this comparison to be possible in the future, I must first guarantee that the classical SVM had the best possible parameters and performance. Next, I will need to explore the ZZFeatureMap and its associated circuit further. Then, I will need to explore the most advanced SVM and QSVM to see if theoretical advantages are still present.

Bibliography

[1] “Early Stage Diabetes Risk Prediction Dataset.” https://www.kaggle.com/datasets/ishandutta/early-stage-diabetes-risk-prediction-dataset (accessed Sep. 24, 2022).

[2] “Diabetes Dataset.” https://www.kaggle.com/datasets/mathchi/diabetes-data-set (accessed Sep. 24, 2022).

[3] X. B. Sicotte, “Answer to ‘How to intuitively explain what a kernel is?,’” *Cross Validated*, Jul. 08, 2018. https://stats.stackexchange.com/a/355046 (accessed Oct. 01, 2022).

[4] P. Huembeli, “Introduction into Quantum Support Vector Machines,” *Medium*, Oct. 23, 2019. https://medium.com/@patrick.huembeli/introduction-into-quantum-support-vector-machines-727f3ccfa2b4 (accessed Sep. 24, 2022).

[5] V. Havlicek *et al.*, “Supervised learning with quantum enhanced feature spaces,” *Nature*, vol. 567, no. 7747, pp. 209–212, Mar. 2019, doi: 10.1038/s41586-019-0980-2.

[6] “ZZFeatureMap — Qiskit 0.38.0 documentation.” https://qiskit.org/documentation/stubs/qiskit.circuit.library.ZZFeatureMap.html (accessed Oct. 01, 2022).

[7] V. Havlicek *et al.*, “Supervised learning with quantum enhanced feature spaces,” *Nature*, vol. 567, no. 7747, pp. 209–212, Mar. 2019, doi: 10.1038/s41586-019-0980-2.

Appendix

Colab link, original work

<https://colab.research.google.com/drive/1m-fRDBoexh44MwVnE-TJXOMP4Dduc0PL?usp=sharing>

Qiskit QSVM colab link

<https://colab.research.google.com/drive/1bOJ1p5XC-MDDB1cifzjlnPTPna8O87ci?usp=sharing>