Quantum Support Vector Machines

Final Report

Team Members:

Majd Haddad, Hana Shaikh, Francis Otoo, Meba Tadesse, Michael Chernak

CMSC457 Instructor:

Dr. Runzhou Tao, Professor in the Department of Computer Science at University of Maryland, College Park, Ph.D. in Computer Science, Columbia University

May 16, 2025

Table of Contents

Table of Contents 1
Introduction 2
Classical Preprocessing 4
Data Encoding and Feature Maps 5
Quantum Execution and Measurement 6
Kernel Post-Processing and Regularization 8
Model Training 11
Prediction and Deployment 12
Works Cited 15

Introduction

Support Vector Machines (SVMs) are a powerful class of supervised machine learning algorithms widely used for classification, regression, and outlier detection tasks. At their core, SVMs seek to find an optimal hyperplane that separates data points belonging to different classes with the maximum possible margin. This "maximum-margin" hyperplane not only classifies the training data but also provides robustness against noise and improves generalization to unseen data. For example, imagine a dataset with red and black points; an SVM identifies a decision boundary—represented as a line or hyperplane—that cleanly segregates these two classes.

While classical SVMs excel at separating linearly separable data, many real-world datasets are not so straightforward. When the classes cannot be separated by a single linear boundary in the original feature space, kernel methods are employed to implicitly project the data into a higher-dimensional space where linear separation becomes possible. For instance, consider two colored tennis balls placed on a 2D plane, whose colors cannot be separated by a simple line. By projecting these data points into a higher-dimensional space, such as lifting them upward into a third dimension, the previously inseparable classes become distinguishable. This lifting is achieved mathematically through kernel functions, allowing nonlinear support vector machines to efficiently classify complex data without explicitly computing coordinates in the higher-dimensional space.

The mathematical foundation behind SVMs involves selecting critical data points, known as support vectors, and solving an optimization problem that defines the hyperplane maximizing the margin between classes. For linear SVMs, this involves finding a vector w and bias b that satisfy constraints ensuring correct classification with maximum margin. Kernel SVMs extend

this by using nonlinear transformations to map data points into feature spaces where a linear decision boundary can be found, enabling classification of data that is not linearly separable in the original space.

Today, SVMs are employed in diverse applications such as facial recognition, text categorization, speech recognition, and bioinformatics, proving their versatility and effectiveness. With the advent of quantum computing, researchers have sought to enhance classical SVMs by leveraging quantum feature spaces and quantum circuit execution, giving rise to Quantum Support Vector Machines (QSVMs). QSVMs aim to exploit the exponentially large Hilbert spaces accessible via quantum computers to improve classification power and efficiency.

This paper explores recent developments in QSVMs, focusing on how classical preprocessing, quantum data encoding, circuit execution, measurement strategies, and kernel post-processing converge in hybrid quantum-classical frameworks. We compare various state-of-the-art methodologies that execute quantum circuits on noisy quantum hardware and simulators, evaluate measurement techniques such as swap-tests and expectation value estimations, and analyze the challenges of kernel regularization and model training under noise. Through this, we aim to illuminate the practical considerations and future potential of quantum-enhanced SVMs in machine learning.

Midterm Report

Classical Preprocessing

Before encoding classical data into quantum states, many QSVM implementations apply classical preprocessing to reduce feature dimensionality, eliminate redundancies, and improve data quality. This step is especially critical in hybrid quantum-classical models, where minimizing quantum resource demands is essential due to the limitations of current quantum hardware. Preprocessing ensures that quantum circuits remain shallow and that the number of required qubits stays within the bounds of Noisy Intermediate-Scale Quantum (NISQ) devices.

A widely used technique for this purpose is Principal Component Analysis (PCA), which projects high-dimensional data into a lower-dimensional space that preserves the most important variance across features. Masum et al. (2023) combine PCA with wavelet transforms to denoise and compress their input features prior to quantum encoding. By reducing dimensionality upfront, they significantly reduce the number of qubits needed for feature mapping while preserving task-relevant information. This not only improves scalability but also enhances the learnability of the quantum model, especially in noisy settings. On the other hand, Terashi et al. (2021) adopt a lighter preprocessing approach, manually selecting a subset of detector variables so that their data can fit into an 8-qubit quantum circuit. This shows that even modest classical preprocessing—without formal transformations—can significantly improve the feasibility of QSVMs when working with constrained hardware.

Some studies, such as Havlíček et al. (2019) and Park et al. (2020), skip preprocessing altogether, as their datasets are synthetic, structured, or already low-dimensional. Their work focuses on establishing the viability of quantum kernel methods, and while useful as benchmarks, they underscore the practical limitations of skipping preprocessing: real-world data is rarely that well-behaved. These studies highlight the importance of careful dataset design or

selection when preprocessing is not an option, which further emphasizes the role of classical techniques in supporting scalable quantum workflows.

Other preprocessing techniques designed for quantum situations have been investigated in recent research in addition to PCA and wavelets. In order to prepare data in ways that are more compatible with quantum architectures, methods such as autoencoder-based dimensionality reduction and Quantum Random Access Coding (QRAC) are being researched. These techniques, while still in the experimental stage, show promise for quantum-aware classical preprocessing, which may ultimately lessen the classical-quantum mismatch and increase the effectiveness of data loading.

Although preprocessing introduces classical dependencies, it is essential for bridging the gap between the rigorous limitations of NISQ quantum devices and complicated, high-dimensional real-world datasets. It reduces circuit depth, minimizes quantum noise, and minimizes the number of shots needed for execution. Preprocessing is not simply a convenience for the majority of hybrid and even fully quantum models, it is a necessary step to guarantee model viability, accuracy, and efficiency. For significant QSVM applications, classical preprocessing will continue to be a crucial enabler until quantum devices make significant progress.

Data Encoding and Feature Maps

In the first step, the QSVM turns classical data points into quantum states. This step defines how the data is "seen" by the quantum model. Havlíček et al. (2019) use the ZZFeatureMap, where each data point controls rotations on qubits and entangles them using ZZ gates. This encoding makes a simple dataset linearly separable in the quantum feature space.

Terashi et al. (2021) face a more complex task: classifying particle collision events with many features. They use a simpler angle-based encoding, turning each feature directly into rotation angles on different qubits. Park et al. (2020) use the same ZZFeatureMap as Havlíček, but first reduce the feature set. This reduction helps keep the quantum circuit small enough to run on real hardware. Masum et al. (2023) go even further and perform classical preprocessing first (PCA and wavelets), greatly reducing the number of features before encoding them into a shallow quantum circuit.

Overall, each paper shows a different trade-off. More complex encodings like Havlíček's can capture richer patterns but need deeper circuits, increasing noise problems. Simpler angle embeddings, like those used by Terashi, fit more data onto fewer qubits but might oversimplify the quantum kernel. Classical feature reduction (Park, Masum) makes quantum encoding easier, but some useful information might be lost. Choosing the right encoding remains a central challenge for effective QSVM implementations.

Quantum Execution and Measurement

Recent research in quantum machine learning demonstrates a variety of approaches for executing quantum circuits and performing measurements, particularly on NISQ hardware. A prominent example is the work by Havlíček and Park, who implement quantum-enhanced supervised learning using *swap-tests* or overlap circuits on IBM’s superconducting quantum processors, including a 20-qubit device. Their experiments utilize parameterized quantum circuits to map classical data into quantum feature spaces and measure quantum state fidelities through repeated sampling. They employ advanced error mitigation techniques, such as

zero-noise extrapolation, to counteract device noise, enabling high classification accuracy even on current noisy hardware. This approach demonstrates the feasibility of executing meaningful quantum machine learning algorithms on near-term quantum devices which helps us understand the practical limitations and potentials of quantum advantage in real-world noisy environments.

In a similar aspect of hardware benching and simulation, Terashi and collaborators run the same overlap circuits on two IBM quantum chips which are Johannesburg and Boeblingen, where these hardware runs with simulations on both noisy and ideal quantum simulators. Their systematic comparison across devices and simulators provides valuable insights into the effects of noise and hardware imperfections on quantum circuit execution. By being able to benchmark real device outputs against simulator baselines, they were able to identify the practical limitations and error characteristics affecting quantum machine learning tasks. This dual-device and multi-environment evaluation helps inform strategies for noise mitigation and algorithmic robustness in real quantum experiments.

On a hybrid computing perspective, Masum et al. propose a hybrid quantum-classical machine learning framework specifically targeted at natural language processing tasks, such as sentiment analysis. Their methodology leverages trainable parameterized quantum circuits (PQCs) operated on noisy state-vector simulators, rather than direct quantum hardware. They measure expectation values of quantum observables and use the parameter-shift rule to compute analytic gradients, enabling classical optimization algorithms to iteratively update quantum circuit parameters. This hybrid execution and measurement scheme effectively combines classical pre-processing and training with quantum circuit evaluations, illustrating how noisy simulators can be harnessed for developing and testing trainable quantum models under realistic noise assumptions.

Contrasting with these quantum computing-focused studies, the work by Park et al. centers on classical kernel ridge regression techniques enhanced by feature space partitioning to address scalability in large datasets. Their approach does not involve quantum circuit execution or quantum measurements, instead providing a purely classical machine learning benchmark. These differences of recent machine learning research emphasizes the unique challenges faced by quantum methods in terms of hardware implementation and noise.

Together, these studies showcase a broad landscape of quantum execution and measurement approaches, ranging from experimental runs on noisy quantum processors and hardware-simulator comparisons to hybrid quantum-classical frameworks that are optimized using simulators. Each methodology addresses the inherent challenges of NISQ-era quantum computing from different angles, advancing the understanding of how to effectively use quantum resources for machine learning and related tasks.

Kernel Post-Processing and Regularization

In classical machine learning generalizability, the ability of the model to be effective when tested on never seen before data, defines the effectiveness of models. While the expressive power of quantum kernels presents significant advantages in certain learning tasks, it also introduces a heightened risk of overfitting and sensitivity to noise, particularly under NISQ conditions. The same applies to QSVMs and one step that is used to reach this ideal is kernel post-processing and regularization.

Kernel post-processing refers to the transformation or refinement of the quantum kernel matrix after it has been estimated on a quantum device. This step aims to improve

generalizability, reduce the effects of noise and measurement errors, and strengthen the mathematical properties required for learning algorithms, like positive semi-definiteness. Using it in conjunction with classical optimization or in hybrid frameworks is the typical standard.

Huang et al. in their 2021 paper “Power of data in quantum machine learning” utilize post-processing the quantum kernel matrix to center and normalize the kernel before using a classical SVM solver. Centering is a method used to eliminate bias introduced by unequal distributions in feature space and is standard practice to use even in classical kernel methods as it reduces the potential skew in the decision boundary by ensuring that the SVM operates on features that are symmetrically distributed around the origin. Next, the kernel is normalized, where all the diagonal entries are set to 1. This enhances numerical stability and mitigates magnitude discrepancies due to varying quantum state norms as this makes all data points lie on the unit hypersphere in feature space.

Montalbano and Banchi integrate adversarial training techniques to enhance the robustness and generalization capabilities of QSVMs thus producing a novel approach to quantum kernel learning. Aiming to mitigate vulnerabilities inherent in quantum classifiers they optimize quantum kernels through alignment measures and adversarial strategies. To do so they aim to optimize the quantum kernel to align closely with an ideal target kernel using the kernel-target alignment measure which calculates the similarity between the quantum kernel matrix K and a target kernel matrix K_{target} (15). The target kernel is constructed based on the labels of the training data, typically defined as $K_{target} = yy^t$ where y is the label vector. The metric is calculated using the below equation where $\langle \cdot,\cdot \rangle$ denotes the Frobenius inner product. This measure is maximized as it ensures that the quantum kernel captures the underlying structure of the data as represented by the labels, thereby enhancing the classifier's performance.

$$A\left(\mathrm{K}, \mathrm{K}^{t}\right)=\frac{\left\langle \mathrm{K}, \mathrm{yy}^{t}\right\rangle}{\sqrt{\langle \mathrm{K}, \mathrm{K}\rangle\left\langle \mathrm{yy}^{t}, \mathrm{yy}^{t}\right\rangle}}=\frac{\mathrm{y}^{t} \mathrm{K}\, \mathrm{y}}{M\|\mathrm{K}\|}$$

Regularization in QSVMs is used to control overfitting, a common problem in machine learning where the model is overtrained on training data and will no longer generalize well to any other data, by including model complexity penalties in the training objective. This is especially used whenever using complex or expressive quantum kernels. Current research varies in its use of regularization with some finding marginal test accuracy by implementing it and other models bypassing it entirely.

One method of regularization used by Park et al. in their paper from 2020 is Tikhonov regularization also known as ridge regression or ℓ_2 regularization. This procedure is used to stabilize ill-posed problems by adding a penalty term to the loss function. Through the lens of QSVMs this means that it penalizes large weights to reduce model complexity and improve generalization. ℓ_2 regularization "helps encourage smoother blending of the complex functionals" (5) which is helpful when dealing with complex kernels. ℓ_2 is another method that can be used and focuses on "sparse selection as opposed to smoothing" but Park found that ℓ_2 regularization is more effective for QSVMs.

Inspired by Principal Component Analysis used in classical machine learning, Kernel Matrix Truncation using low-rank approximations is a technique that serves to extract meaningful structure from high-dimensional quantum feature maps without overfitting to irrelevant variations in the context of quantum kernels. Huang et al. implemented this technique after demonstrating that quantum kernels derived from finite-depth quantum circuits tend to produce kernel matrices with low effective rank, particularly when the data distribution has

inherent structure. Using Singular Value Decomposition they project the kernel matrix onto its leading singular components which reduces the dimensionality of the kernel-induced feature space and suppresses noisy or redundant directions. The low-rank projection becomes a regularizer and forces the extraction of the most salient features from the data.

Model Training

QSVM training typically follows a hybrid structure in which the kernel matrix is computed on quantum hardware, while the model is trained classically. Havlíček et al. (2019) demonstrate this pipeline by computing quantum kernel values between data points using swap-test circuits and then passing the resulting matrix into a classical SVM solver. This preserves the convex optimization guarantees of classical SVMs while allowing for richer, quantum-enhanced feature spaces.

Once the kernel matrix is obtained, a classical optimizer solves for the optimal hyperplane in feature space using well-established techniques like quadratic programming. This division of labor makes QSVMs compatible with standard tools such as scikit-learn's SVC module and ensures that training remains efficient, even if the underlying feature space is exponentially large.

Masum et al. (2023) take a fully trainable approach by using a parameterized quantum circuit (PQC) and optimizing its parameters end-to-end with the "Adam" optimizer. This method avoids constructing a kernel matrix altogether by using the parameter-shift rule to compute gradients directly on the quantum circuit. This allows the model to learn task-specific parameters during training.

While the variational method offers flexibility, it also increases circuit depth and requires more measurements per iteration, potentially making it less efficient for larger datasets. Compared to this, kernel-based QSVMs are typically more stable and leverage mature classical optimization methods. These two strategies represent complementary directions in quantum machine learning, and they both have tradeoffs in training complexity, stability, and hardware demands.

Prediction and Deployment

The final step in the QSVM pipeline involves using the trained model to predict class labels for new, unseen data. This phase is where the model's ability to generalize is tested and its practical value is demonstrated. No matter how powerful the earlier steps, like data encoding or quantum kernel construction, QSVMs must ultimately produce reliable outputs when deployed on new inputs. If inference is too resource-intensive or sensitive to noise, the benefits of using a quantum approach may not justify the cost.

In kernel-based QSVM implementations such as those by Havlíček et al. (2019), Park et al. (2020), and Terashi et al. (2021), predictions are made by computing a quantum kernel vector between the new input and all training samples. Each entry in this vector represents the overlap, or fidelity, between the encoded quantum state of the test point and each support vector, typically evaluated using a swap test or similar circuit. Once these kernel values are obtained through quantum execution, they are passed into a classical SVM decision function, usually a weighted sum followed by a sign function, to output the predicted class. While this hybrid design allows the model to harness the expressive power of quantum feature spaces, it also highlights a

bottleneck: predicting a single data point may require running many quantum circuits, especially as the number of support vectors grows.

Park et al. (2020) applied Tikhonov regularization to the kernel matrix prior to prediction in order to enhance test-time performance. By fixing distortions brought on by hardware noise, this method improves robustness and produces predictions that are more stable. When working with real-world quantum devices or simulations, when absolute realism cannot be expected, this is especially helpful.

Predictions are handled differently by alternative architectures, such as the hybrid QSVM investigated by Masum et al. (2023). They employ a parameterized quantum circuit (PQC) that was trained end-to-end in previous steps rather than creating a kernel vector. An expectation value that directly correlates to a class label is returned by the PQC after performing a forward pass on the test input at inference time. This approach moves the computing burden to the training phase and still depends on repeated measurements for accuracy, even if it eliminates the requirement to analyze numerous quantum overlaps.

Finally, by comparing the performance of QSVM inference on simulators, Kubota et al. (2023) offer more information about the deployment step. Their findings highlight the substantial effects of encoding complexity, circuit depth, and measurement recurrence on prediction delay. Because of resource constraints, even models with strong training accuracy may have trouble with real-time classification tasks. (Kubota et al. paper)

Despite these difficulties, the most important and obvious component of the QSVM process is still prediction. From experimental models to potentially scalable, practical applications, the ability to take fresh inputs, process them via a quantum-enhanced system, and yield reliable outputs signifies the shift. Realizing the full potential of quantum machine learning

in domains like natural language processing, finance, and medicine may be possible as technology advances and the prediction step becomes quicker and more effective.

Works Cited

Havlíček, V., Córcoles, A. D., Temme, K., et al. (2019). *Supervised learning with quantum-enhanced feature spaces.* Nature, 567, 209–212. https://doi.org/10.1038/s41586-019-0980-2

Huang, H.-Y., Broughton, M., Mohseni, M., Babbush, R., Boixo, S., Neven, H., & McClean, J. R. (2021, February 10). *Power of data in Quantum Machine Learning*. arXiv.org. https://arxiv.org/abs/2011.01938

Lui, Ruoshi, et al. (2023, March 20). *Zero-1-to-3: Zero-Shot One Image to 3D Object.* arXiv.Org arxiv.org/abs/2303.11328

Masum, Abu Kaisar Mohammad et al. (2023, October 8) *Hybrid Quantum-Classical Machine Learning for Sentiment Analysis.* arXiv.Org. arxiv.org/abs/2310.10672

Montalbano, G., & Banchi, L. (2025, February 5). *Quantum adversarial learning for kernel methods - quantum machine intelligence*. SpringerLink. https://link.springer.com/article/10.1007/s42484-025-00238-8

Park, J.-E., Quanz, B., Wood, S., Higgins, H., & Harishankar, R. (2020, December 14). *Practical application improvement to quantum SVM: Theory to practice*. arXiv.org. https://arxiv.org/abs/2012.07725

Terashi, K., et al. (2021). *Event Classification with Quantum Machine Learning in High-Energy Physics.*Computing and Software for Big Science, 5(1), 1–11. https://arxiv.org/abs/2002.09935