

# Использование квантовых вычислений в методе опорных векторов для задач классификации

## Введение

Квантовые компьютеры способны открыть новые возможности в машинном обучении, которые были недоступны при классических подходах, как со стороны улучшения точности моделей, так и со стороны сокращения их вычислительной сложности. Ускорение алгоритмов обучения с учителем может быть потенциально полезно для решения задач “больших данных”. Один из таких алгоритмов является метод опорных векторов (support vector machine, SVM), который решает задачи регрессии и классификации при помощи построения оптимальных разделяющих гиперплоскостей в оригинальном пространстве признаков данных или более многомерном пространстве. SVM имеет следующий ряд преимуществ как алгоритм обучения с учителем: эффективность в высоко-размерных пространствах, даже если число измерений больше числа наблюдений в данных; экономия памяти, так как он хранит только подмножество точек наблюдений в решающей функции (т.н. опорные векторы или support vectors), и гибкость из-за возможности использовать различные ядерные функции.

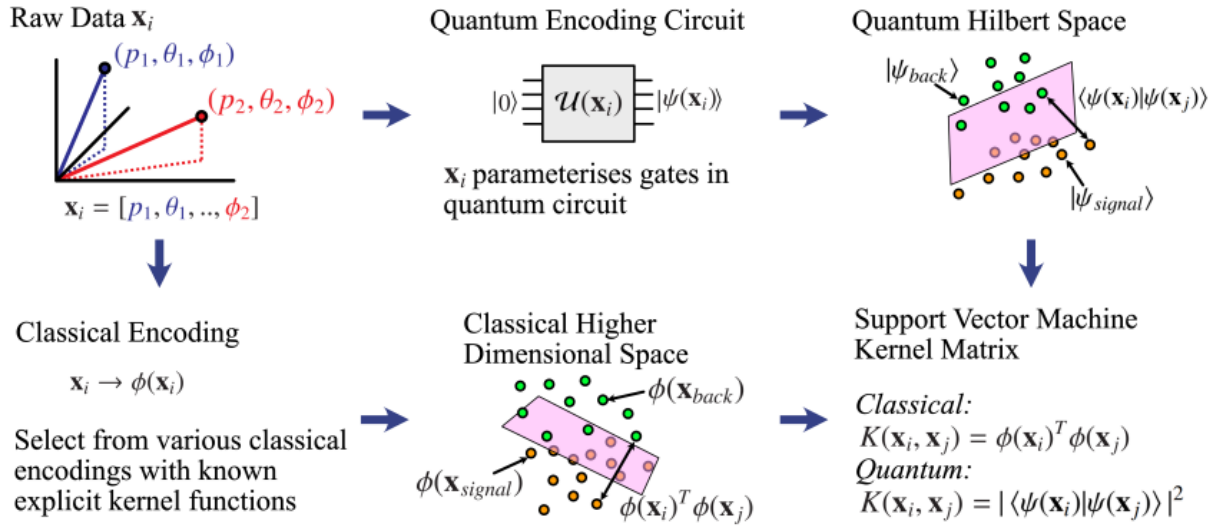
Именно ядерный метод или “ядерный трюк” делает SVM полезным для решения множества задач и возможным для улучшения с помощью квантовых вычислений. Ядерный метод можно рассматривать как обучение на примерах - предсказание для непомеченного ввода, т.е. не входящего в тренировочное множество, изучается при помощи функции сходства (называемой ядром) между непомеченным входом и каждым из тренировочных входов (или избранным из них). Например, ядерный бинарный классификатор вычисляет взвешенную сумму похожести по формуле:

$$\hat{y} = \text{sgn} \sum_{i=1}^n w_i y_i k(\mathbf{x}_i, \mathbf{x}')$$

где  $w_i$  – веса тренировочных примеров, определенные алгоритмом обучения,  $k$  – ядерная функция.

Самыми распространенными классическими ядерными функциями являются линейный, полиномиальный, радиально-базисный и сигмоидальный.

Применение квантовых вычислений к SVM вынуждает перенести данные из классического формата в квантовый: эта задача решается при помощи построения карты квантовых признаков (quantum feature map)  $\phi(\mathbf{x})$ , которая транслирует вектор  $\mathbf{x}$  в квантовое состояние  $|\Phi(\mathbf{x})\rangle\langle\Phi(\mathbf{x})|$ . Это достигается за счет применения унитарного оператора  $U_{\Phi(\mathbf{x})}$  к изначальному состоянию  $|0\rangle_n$ . Разница между процессами классического и квантового SVM показана на рис. 1.



**Рис. 1.** Концепции классического и квантового SVM.

Карта квантовых признаков  $\phi(\mathbf{x})$  естественным образом дает начало квантовому ядру  $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_j)^\dagger \phi(\mathbf{x}_i)$ , которое можно рассматривать как меру схожести векторов  $\mathbf{x}_i$  и  $\mathbf{x}_j$ . Квантовое ядро может быть рассмотрено как матрица  $K_{ij} = \|\langle\phi^\dagger(\mathbf{x}_j)|\phi(\mathbf{x}_i)\rangle\|^2$ . Предполагая, что карта признаков это параметризованная квантовая цепь, которая может быть описана как унитарная трансформация  $\mathcal{U}\phi(\mathbf{x})$  для  $n$  кубитов, можно подсчитать каждый элемент ядерной матрицы:

$$\|\langle\phi^\dagger(\mathbf{x}_j)|\phi(\mathbf{x}_i)\rangle\|^2 = \|\langle 0^{\otimes n} | \mathcal{U}^\dagger \phi(\mathbf{x}_j) \mathcal{U} \phi(\mathbf{x}_i) | 0^{\otimes n} \rangle\|^2$$

Таким образом, мы получаем оценку квантовой ядерной матрицы, которую можно использовать в ядерных алгоритмах машинного обучения, в частности – SVM.

Необходимо указать, что алгоритмы с квантовыми ядрами имеют потенциальное преимущество перед классическими только если это ядро трудно вычислить или оценить классическими способами [1]. При этом существование задач, для которых квантовые ядра имеют преимущество, уже доказано [2].

## Quantum Feature Map

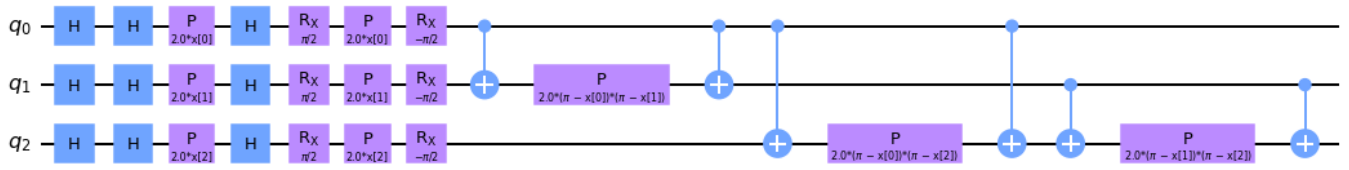
В данной работе рассматриваются вариации карты квантовых признаков Паули (Pauli Feature Map), в частности ZZ Feature Map и Z Feature Map.

Pauli Feature Map:

$$\mathcal{U}_{\Phi(\mathbf{x})} = \prod_d \mathcal{U}_{\Phi(\mathbf{x})} H^{\otimes n}, \quad \mathcal{U}_{\Phi(\mathbf{x})} = \exp\left(i \sum_{S \subseteq [n]} \phi_S(\mathbf{x}) \prod_{k \in S} P_k\right),$$

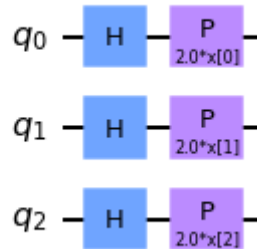
$$S \in \{(nk) \text{ combinations}, k=1, \dots, n\}$$

Если  $P_0=X, P_1=Y, P_2=ZZ$ :



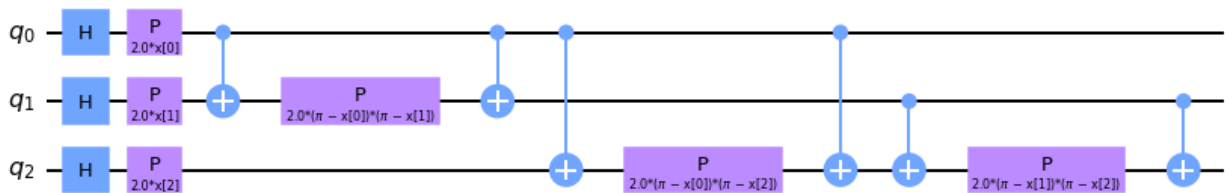
Z Feature Map ( $k=1, P_0=Z$ ):

$$\mathcal{U}_{\Phi(\mathbf{x})} = \left( \exp \left( i \sum_j \phi_{\{j\}}(\mathbf{x}) Z_j \right) H^{\otimes n} \right)^d$$



ZZ Feature Map ( $k=2, P_0=Z, P_1=ZZ$ ):

$$\mathcal{U}_{\Phi(\mathbf{x})} = \left( \exp \left( i \sum_{jk} \phi_{\{j,k\}}(\mathbf{x}) Z_j \otimes Z_k \right) \exp \left( i \sum_j \phi_{\{j\}}(\mathbf{x}) Z_j \right) H^{\otimes n} \right)^d$$



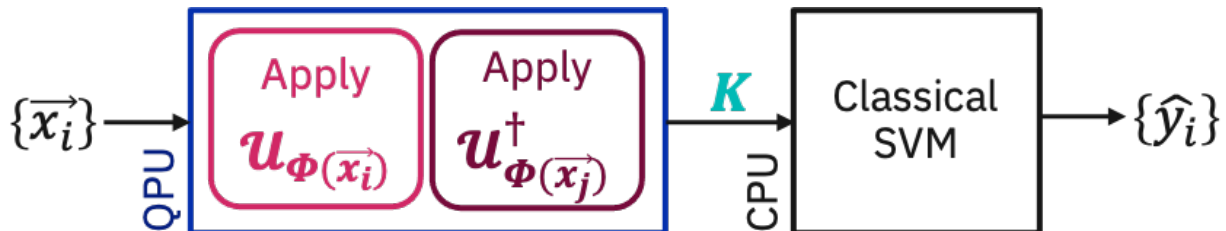
## Quantum Support Vector Classification

Алгоритм классификации на базе метода опорных векторов (support vector classifier, SVC) с квантовым ядром состоит из следующих шагов [1]:

1. Построение тренировочной и проверочной матриц квантового ядра:

- а) Для каждой пары векторов в тренировочных данных  $\mathbf{x}_i, \mathbf{x}_j$  применить карту признаков и получить вероятности:  $K_{ij} = ||\langle 0 | \mathbf{U}^\dagger \Phi(\mathbf{x}_j) \mathbf{U} \Phi(\mathbf{x}_i) | 0 \rangle||^2$
- б) Выполнить предыдущий подшаг для вектора из тренировочных данных и вектора из проверочных данных

2. Использовать тренировочную и проверочную матрицы квантового ядра в классическом алгоритме классификации на базе метода опорных векторов



## Эксперименты

### 1. Сравнение классических и квантовых ядер

Было проведено сравнение точности SVC с использованием классических (Linear - линейный, RBF - радиально-базисный, Polynomial – полиномиальный) и квантовых (Z, ZZ, Pauli X-Y-ZZ) ядер на следующих датасетах с задачей классификации:

**Таблица 1. Характеристики датасетов.**

| Dataset name            | Original features | Classes | Samples |
|-------------------------|-------------------|---------|---------|
| Tic-Tac-Toe             | 9                 | 2       | 958     |
| Banknote Authentication | 4                 | 2       | 1372    |
| Digits                  | 64                | 10      | 1797    |
| Iris                    | 4                 | 3       | 150     |
| Wine                    | 13                | 3       | 178     |
| synth_bin_cls           | 4                 | 2       | 500     |
| synth_multi_cls         | 4                 | 5       | 500     |
| synth_blobs_2           | 4                 | 2       | 500     |
| synth_blobs_5           | 4                 | 5       | 500     |

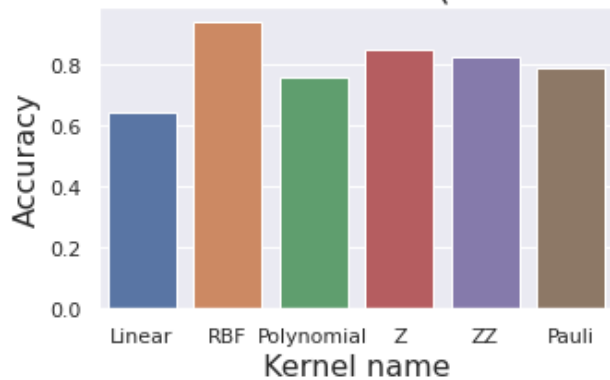
Четыре последних датасета являются синтетическими и были сгенерированы с помощью функций `datasets.make_blobs` и `datasets.make_classification` из библиотеки `scikit-learn`.

У датасетов с количеством признаков больше 4-ех была снижена размерность при помощи алгоритма Principal Component Analysis (PCA).

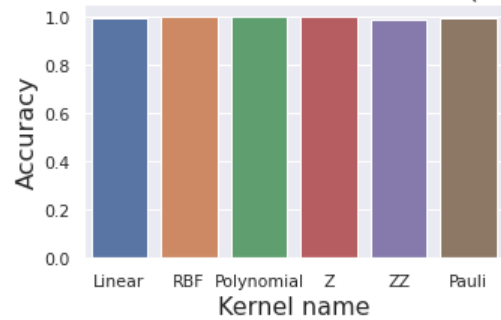
Параметр регуляризации SVC “C” был равен 10 во всех экспериментах.

Датасеты были разбиты на тренировочную и проверочную выборки с 75 и 25% векторов от общего числа соответственно.

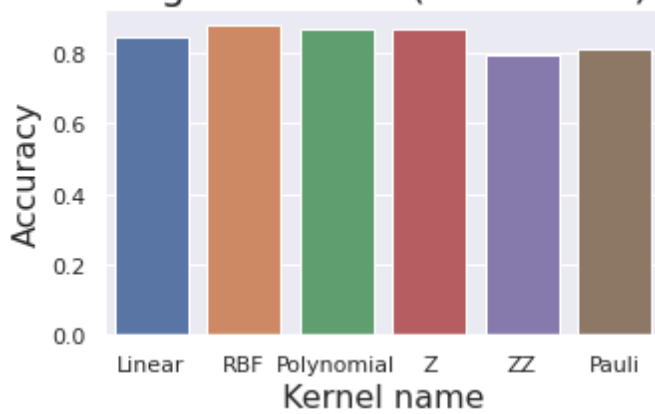
Tic-Tac-Toe dataset (2 classes)



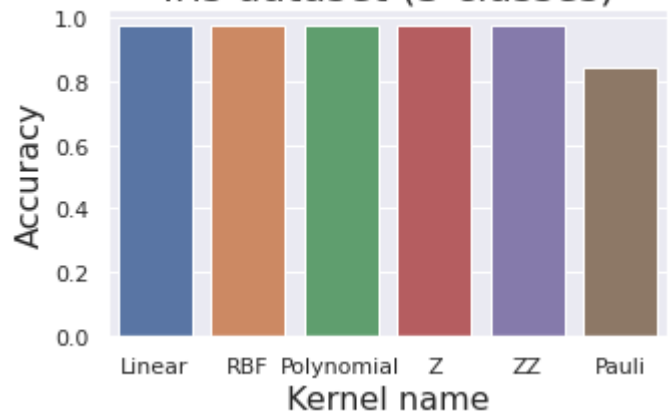
Banknote Authentication dataset (2 classes)



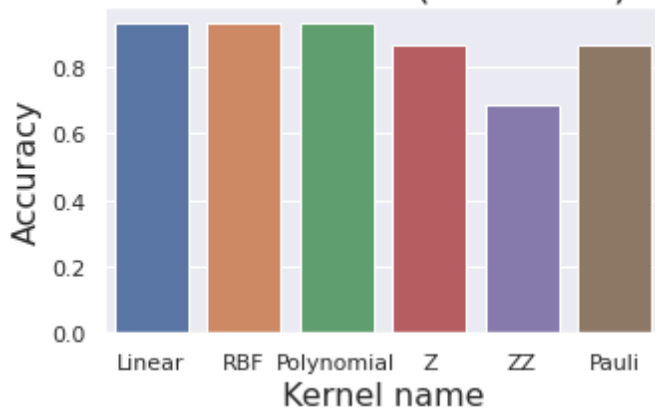
Digits dataset (10 classes)

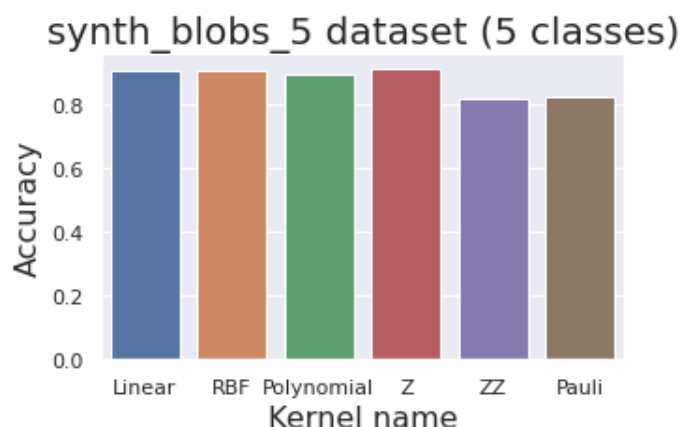
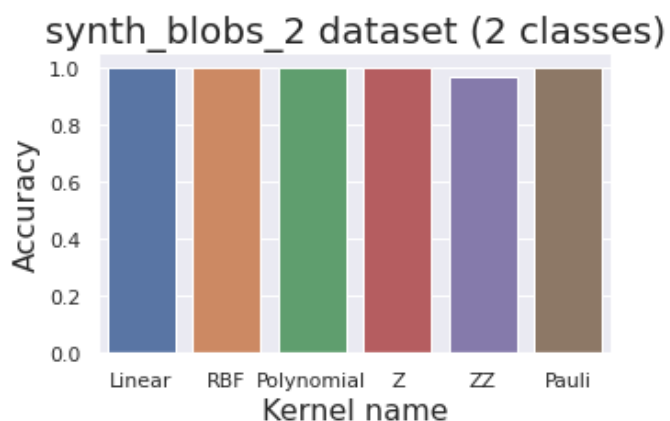
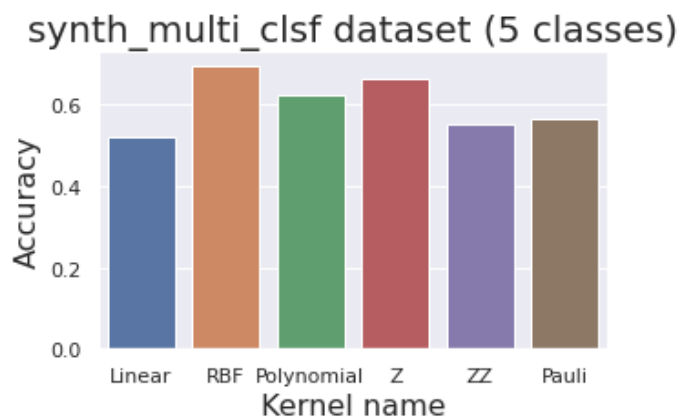
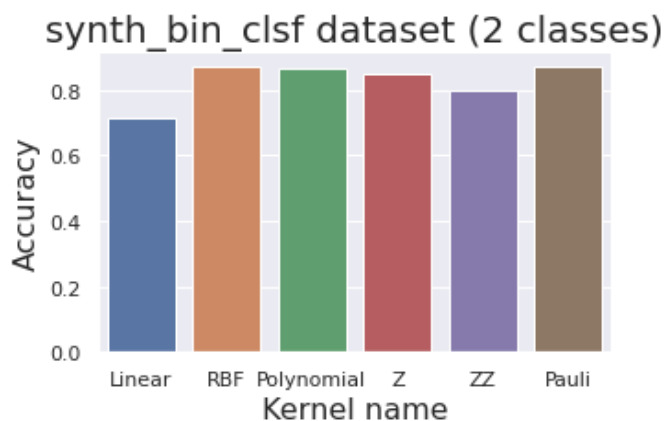


Iris dataset (3 classes)



Wine dataset (3 classes)

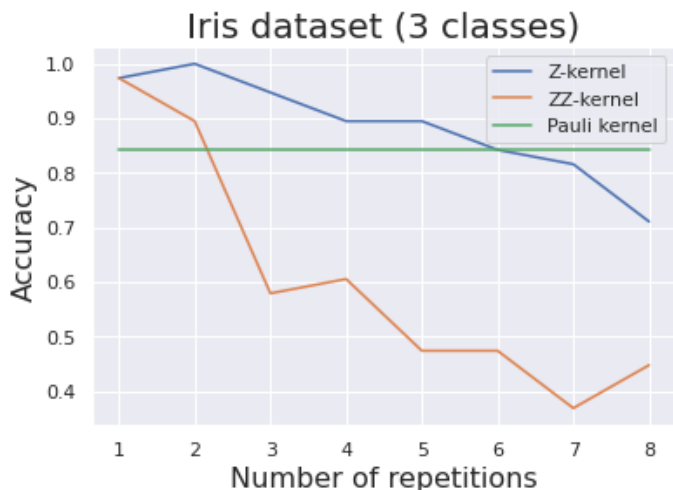




Заметно, что в большинстве случаев как минимум одно квантовое ядро работает с такой же или чуть худшей точностью, за исключением synth\_blobs\_5 датасета, где у Z-ядра есть небольшое преимущество, возможно, из-за скученности пяти кластеров в пространстве.

## 2. Повторение операторов

Был проведен эксперимент с повторением операторов в цепи квантового ядра для датасетов Iris и Wine:



## Reference

1. Supervised learning with quantum-enhanced feature spaces. Havlicek et al. Nature 567, 209-212 (2019).
2. A rigorous and robust quantum speed-up in supervised machine learning. Yunchao Liu, Srinivasan Arunachalam, Kristan Temme. ArXiv:2010.02174 (2020)  
<https://arxiv.org/abs/2010.02174>