SANTA CLARA UNIVERSITY

Quantum Machine Learning Seminars

Week 6: Quantum Clustering & Classification

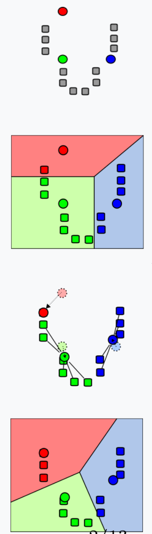
Dr. Hassan Alshal

Outline

1. Quantum Clustering
2. Quantum Classification

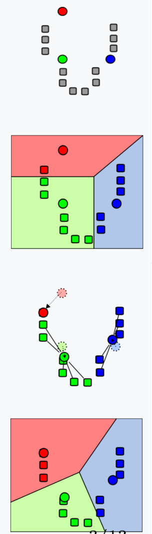
Quantum K-means

- A dataset V has N instances in a d-dimensional feature space $\mathcal{F}$. A partition into K clusters is $P = \{P_1, \ldots, P_K\}$ such that the clusters are disjoint and cover V:
  $P_i \cap P_j = \emptyset \quad (i \neq j), \qquad \bigcup_{j=1}^{K} P_j = V.$
- The centroid of P_j is the mean of its elements:
  $c_j := \frac{1}{N_j} \sum_{\mathbf{x} \in P_j} \mathbf{x}, \qquad N_j = |P_j|.$
- K-means iteratively minimizes the dispersion using a reference distance $d(\cdot, \cdot)$ as: $f(P) = \sum_{j=1}^{K} \sum_{\mathbf{x} \in P_j} d(\mathbf{x}, c_j)$.
- For an initial partition, compute centroids, reassign each point to its nearest centroid, and iterate until convergence $\mathcal{O}(NKd)$. Iterations no. $< N$. It should reach a local minimum of f.



Quantum K-means

- For quantum K-means, assume $\mathcal{F} = \mathbb{R}^d$ with Euclidean distance and use amplitude encoding in an n-qubit register.
- Store components $\{x_{ij}\}_j$ and the norm $\|\mathbf{x}_i\|$ in QRAM so the normalized state can be retrieved in: $|\mathbf{x}_i\rangle = \frac{1}{\|\mathbf{x}_i\|} \sum_j x_{ij} |j\rangle$.
- If centroids are computed classically and then encoded as quantum states, the routine `Qdist` is used to compute Euclidean distances between each datapoint and each centroid. Since one K-means iteration requires NK distances, the per-iteration complexity becomes $O(NK \log d)$.



Qdist routine

- Use amplitude encoding for real vectors $\mathbf{x} \in \mathbb{R}^d$, encode it into a $\log d$-qubit as $|\mathbf{x}\rangle = \frac{1}{|\mathbf{x}|}\sum_{j=1}^{d} x_j |j\rangle$.
- For two real vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, the overlap of their amplitude-encoded states gives: $\langle \mathbf{x}|\mathbf{y}\rangle = \cos(\mathbf{x}, \mathbf{y}) = \dfrac{\mathbf{x}\cdot\mathbf{y}}{|\mathbf{x}||\mathbf{y}|}$.
- Prepare an entangled state with the data register, $|\psi\rangle = \frac{1}{\sqrt{2}}\Big(|0\rangle \otimes |\mathbf{x}\rangle + |1\rangle \otimes |\mathbf{y}\rangle\Big) \in \mathbb{C}^2 \otimes (\mathbb{C}^2)^{\otimes n}$, then another state $|\varphi\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle - |1\rangle\big) \in \mathbb{C}^2$.
- Check the overlap with the ancilla $|\varphi\rangle$ to get $\Phi = \frac{1}{\sqrt{2}}\big(\langle\varphi|0\rangle|\mathbf{x}\rangle + \langle\varphi|1\rangle|\mathbf{y}\rangle\big) = \frac{1}{\sqrt{2}}\big(|\mathbf{x}\rangle - |\mathbf{y}\rangle\big)$.
- SWAP to estimate $|\Phi|^2$, and $|\Phi|^2 = 1 - \frac{\mathbf{x}\cdot\mathbf{y}}{|\mathbf{x}||\mathbf{y}|}$. Hence $\mathbf{x}\cdot\mathbf{y} = (1 - |\Phi|^2)|\mathbf{x}||\mathbf{y}|$, and the squared distance is $|\mathbf{x}-\mathbf{y}|^2 = |\mathbf{x}|^2 + |\mathbf{y}|^2 - 2\mathbf{x}\cdot\mathbf{y}$.

Distance-based Quantum Classification

- Distance-based classification labels a new point $x \in \mathbb{R}^d$ by comparing it to a training set $\{(x_i, y_i)\}_{i=1}^N$ with $y_i \in \{-1, 1\}$; classically, costs $\mathcal{O}(Nd)$.
- A quantum-inspired distance-weighted binary rule assigns the label via $y = \text{sgn}\left[\sum_{i=1}^{N} y_i \left(1 - \frac{1}{4}|x - x_i|^2\right)\right]$.
- The model uses all training labels in the decision, with contributions decaying with squared distance $|x - x_i|^2$.
- Data are assumed normalized and standardized (notably $x_i^T x_i = 1$ for each i), enabling simple identities relating $|x \pm x_i|^2$ and $|x - x_i|^2$.
- Amplitude-encoding as $|\psi_{x_i}\rangle = \sum_{j=1}^{d} x_{ij} |j\rangle$, same for test point $|\psi_x\rangle$.
- Using an index register plus an ancilla and a label qubit, the composite state is prepared as $|\Psi_0\rangle = \frac{1}{\sqrt{2N}} \sum_{i=1}^{N} |i\rangle \left(|0\rangle|\psi_x\rangle + |1\rangle|\psi_{x_i}\rangle\right) |y_i\rangle$.

Distance-based Quantum Classification

- Applying a Hadamard on the ancilla yields $|\Psi_1\rangle = \frac{1}{2\sqrt{N}} \sum_{i=1}^{N} |i\rangle \left[|0\rangle \left(|\psi_x\rangle + |\psi_{x_i}\rangle\right) + |1\rangle \left(|\psi_x\rangle - |\psi_{x_i}\rangle\right)\right] |y_i\rangle$.
- Measuring the ancilla and postselecting outcome 0 gives $|\Psi_2\rangle = \frac{1}{2\sqrt{N\mathbb{P}(0)}} \sum_{i=1}^{N} |i\rangle|0\rangle \left(|\psi_x\rangle + |\psi_{x_i}\rangle\right) |y_i\rangle =$ $\frac{1}{2\sqrt{NP(0)}} \sum_{i=1}^{N} \sum_{j=1}^{d} |i\rangle|0\rangle(x_j + x_{ij})|j\rangle|y_i\rangle$.
- The postselection probability is $P(0) = \frac{1}{4N} \sum_{i=1}^{N} |x + x_i|^2$, and for standardized data it is stated that $P(0) \simeq 0.5$.
- Conditioning on the label $l = \pm 1$, measuring the label register yields $P_b(l) = \frac{1}{4N\mathbb{P}(0)} \sum_{i;y_i=l} |x + x_i|^2$.

Distance-based Quantum Classification

- For unit-normalized vectors, the identity $\frac{1}{4N}\sum_{i=1}^{N}|x+x_i|^2 = 1-\sum_{i=1}^{N}\frac{1}{4N}|x-x_i|^2 = \frac{1}{N}\sum_{i=1}^{N}\left(1-\frac{1}{4}|x-x_i|^2\right)$ links these measurement probabilities to the decision rule.
- A second distance-based classifier uses cosine similarity with $y(x) = \operatorname{sgn}\left(\sum_{i=0}^{N-1} y_i \cos(x_i, x)\right)$ and $\cos(x,y) = \frac{x\cdot y}{|x|\cdot|y|}$; for normalized vectors, $|x-y| = \sqrt{2(1-x\cdot y)}$.
- Its quantum realization prepares $|X\rangle = \frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle|x_i\rangle|b_i\rangle$ with $b_i = \frac{1-y_i}{2}$, and $|\psi_x\rangle = \frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle|x\rangle|-\rangle$
- Followed by $\frac{1}{\sqrt{2}}(|X\rangle|0\rangle + |\psi_x\rangle|1\rangle)$ and a SWAP test giving $\mathbb{P}(1) = \frac{1}{4}(1-\langle X|\psi_x\rangle)$ with $\langle X|\psi_x\rangle = \frac{1}{N\sqrt{2}}\sum_{i=0}^{N-1} y_i \cos(x_i, x)$, so $y(x) = \operatorname{sgn}(1-4P(1))$.

Quantum k-Nearest Neighbors

- Classical kNN classifies a new point x by computing distances $d(x, x_i)$ to all training points, selecting the k closest, and majority-voting their labels; the per-query cost is typically $\mathcal{O}(Nd)$ for N points in d dimensions.
- The quantum kNN variant replaces distance computations with a similarity score based on $|\cos(x,y)|^2 = \left|\frac{x \cdot y}{|x|.|y|}\right|^2$, so “nearest” means “most similar.”
- Using amplitude encoding, normalized data vectors are mapped to quantum states $|x\rangle$ and $|x_i\rangle$ so the overlap $|\langle x|x_i\rangle|^2$ encodes similarity.
- The algorithm prepares a superposition over the training set together with the query, e.g. $|\Psi_0\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle|x_i\rangle|x\rangle|0\rangle$, where $|i\rangle$ is an index register and the last qubit is an ancilla.

Quantum k-Nearest Neighbors

- A SWAP-test-style procedure is applied so that measuring the ancilla yields probabilities that depend on the average similarity; specifically $\mathbb{P}(\alpha) = \frac{1}{2} + (-1)^{\alpha} \frac{1}{2N} \sum_{i=0}^{N-1} |\langle x|x_i\rangle|^2$ for $\alpha \in 0, 1$.
- After observing an ancilla outcome α, measuring the index register biases the sampling toward more similar points; the conditional probability takes the simplified form $\mathbb{P}(i|\alpha) = \frac{1+(-1)^{\alpha}|\langle x|x_i\rangle|^2}{N+(-1)^{\alpha}\sum_{j=0}^{N-1}|\langle x|x_j\rangle|^2}$.
- A practical "neighbor score" is defined by the difference $\mathbb{Q}(i) := \mathbb{P}(i|0) - \mathbb{P}(i|1) = \frac{2\left(|\langle x|x_i\rangle|^2 - C\right)}{N(1-C^2)}$, where $C = \frac{1}{N}\sum_i |\langle x|x_i\rangle|^2$; maximizing $\mathbb{Q}(i)$ is equivalent (monotonically) to maximizing $|\langle x|x_i\rangle|^2$.
- Algorithm: repeat sampling to estimate $\mathbb{Q}(i)$ well, pick the K indices with largest $\mathbb{Q}$, then predict the label by majority voting over $(x_{i_k}, y_{i_k})_{k=1}^{K}$.

Quantum Support Vector Machine

- SVM is a binary classifier that finds a maximum-margin separating hyperplane in feature space, written as $w \cdot x + b = 0$, with labels $y_i \in \pm 1$ for training points $x_i \in \mathbb{R}^d$.
- The goal is to maximize the margin by solving $\arg\min_{w,b} \frac{|w|^2}{2}$ subject to $y_i(w \cdot x_i + b) \geq 1$ for all i.
- The Lagrangian introduces multipliers $\alpha_i \geq 0$ as $\mathcal{L}(w, b, \alpha) = \frac{1}{2}|w|^2 - \sum_{i=1}^{N} \alpha_i \big(y_i(w \cdot x_i + b) - 1 \big)$.
- Stationarity gives the key structural facts $w = \sum_{i=1}^{N} \alpha_i y_i x_i$ and $\sum_{i=1}^{N} \alpha_i y_i = 0$, meaning the solution is a weighted combination of training points and only $\alpha_i > 0$ points act as support vectors.

Quantum Support Vector Machine

- Eliminating w, b yields the dual objective $\mathcal{L}_d(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$ with constraints $\alpha_i \geq 0$ and $\sum_i \alpha_i y_i = 0$.
- The classifier evaluates new inputs via a kernel expansion $y(x) = \text{sgn}\left(\sum_{i=1}^{N} \alpha_i \kappa(x_i, x) + b\right)$, where in the linear case $\boxed{\kappa(x_i, x) = x_i \cdot x}$.
- The "kernel trick" generalizes to nonlinearly separable data by embedding with $\phi : \mathbb{R}^d \to \mathbb{R}^{d'}$ and defining $\kappa(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$, so the algorithm uses inner products in feature space without explicitly constructing $\phi(x)$.
- The central quantum object is the "kernel matrix" K with entries $K_{ij} = \kappa(x_i, x_j)$; and the goal is to estimate it.

Quantum Classification

Quantum Support Vector Machine

- With amplitude encoding, vectors are represented as states $|x\rangle$ and $|x_i\rangle$ so that overlaps encode similarity; a SWAP test can estimate $|\langle x_i|x_j\rangle|^2$.
- The more "kernel-centric" quantum approach summarized (least-squares SVM) replaces inequality constraints with equalities using slack variables: $y_i(w \cdot x_i + b) = 1 - \xi_i$, and optimizes $\arg\min_{w,b,\xi} \frac{1}{2}|w|^2 + \frac{\gamma}{2}\sum_{i=1}^{N} \xi_i^2$.
- This LS-SVM reduces training to solving a linear system whose kernel dependence appears explicitly through K: $\sum_{i=1}^{N} \alpha_i = 0$ and $\sum_{i=1}^{N} \alpha_i (x_i \cdot x_j) + b + \gamma^{-1}\alpha_j = y_j$, often written compactly as $F\binom{b}{\alpha} = \binom{0}{y}$ with $F = \begin{pmatrix} 0 & \mathbf{1}^T \\ \mathbf{1} & K + \gamma^{-1} I \end{pmatrix}$.

Thank You!