Multi-class Support Vector Machine on Quantum Annealers

BADEMA, Junya ARAI and Keitaro HORIKAWA
NTT Software Innovation Center
Email: tokuma.tomoe.py@hco.ntt.co.jp

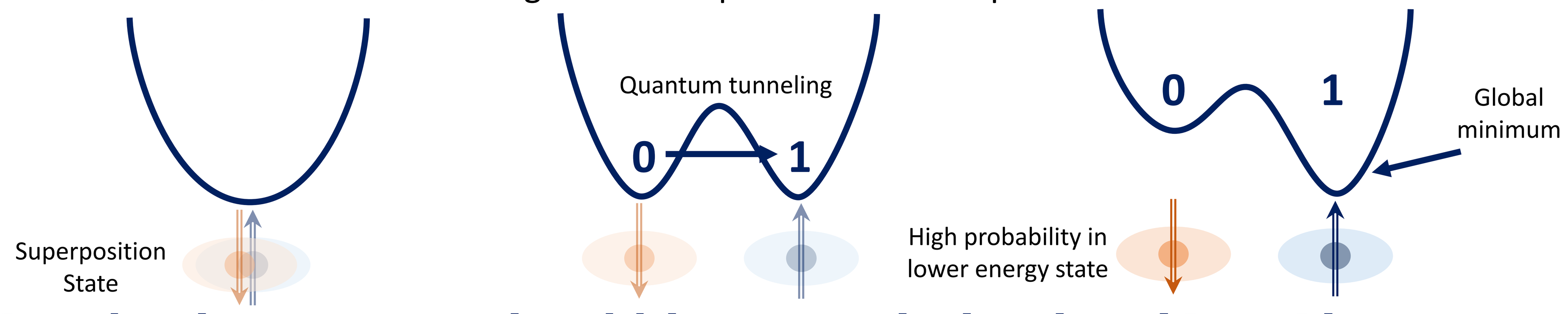
(i) Introduction

- ✓ With the end of Moore's Law, new generation computing such as **quantum computing** is beginning to prosper.
- ✓ Both **universal gate quantum computing and quantum annealing (QA)** may be able to improve computing performance in the future. However, QA is easier to realize compared with universal gate quantum computing. There are already some commercialized quantum annealers such **as D-Wave 2000Q [1]** and quantum-inspired annealers such as **Fujitsu Digital Annealer (DA) [2]**.
- ✓ **QA is specialized for combinatorial optimization problems**. To expand its application, we proposed **a multi-class SVM algorithm on Quantum Annealers**.
- ✓ Our main idea is classifying multi-class data using multiple binary classifiers. When the results of binary classifiers are inconsistent, we **compare the energy obtained by quantum annealing to prioritize the classifiers**.
- ✓ We evaluated our method using synthetic data and benchmark dataset (IRIS). The experimental results showed that our method can **classify multi-class data at a precision comparable to classical implementations**.

(ii) Preliminaries

Quantum annealing

- Quantum annealing is a metaheuristic for finding the global minimum by a process using quantum fluctuations. Quantum-annealing-based computers are called quantum annealers.

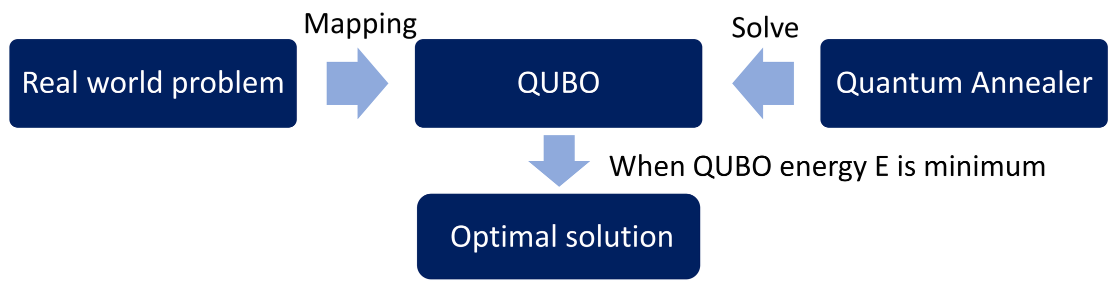


Quadratic unconstrained binary optimization (QUBO)

- To solve problems on quantum annealers, we need to formulate a problem as QUBO.
- A QUBO is a minimization problem of energy E defined as follows:

QUBO

$$E = \sum_{i \le j} a_i Q_{ij} a_j, \quad a_i \in \{0,1\}$$



Previous work

- Support vector machine (SVM) is a machine learning model for binary classification. SVM is trained to **find a hyperplane that separates two classes with the maximum margin**.
- We use a kernel trick when the data is linearly inseparable.

Classical SVM (cSVM) — **Encode[3]** → **Quantum SVM (qSVM)**

Real variables α_n

$$\alpha_n = \sum_{k=0}^{K-1} B^k a_{Kn+k}\,,\, where\ a_{Kn+k} \in \{0,1\}$$

Binary variables a_{Kn+k}

$$\max_{\alpha} \quad W(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{N} t_i t_j \alpha_i \alpha_j k(\mathrm{x}_i, \mathrm{x}_j)$$

subject to $\boxed{0 \leqq \alpha_i \leqq C, i = 1, \cdots, N}$

and $\boxed{\sum_{i=1}^{N} \alpha_i t_i = 0}$

Automatically satisfied in encoding formation by setting $C = \sum_{k=0}^{K-1} B^k$

$$E = \frac{1}{2}\sum_{n,m=0}^{N-1}\sum_{k,j=0}^{K-1} a_{Kn+k} a_{Km+j} B^{k+j} t_n t_m k(\mathrm{x}_n, \mathrm{x}_m) - \sum_{n=0}^{N-1}\sum_{k=0}^{K-1} B^k a_{Kn+k} + \boxed{\xi\Big(\sum_{n=0}^{N-1}\sum_{k=0}^{K-1} B^k a_{Kn+k} t_n\Big)^2}$$

qSVM QUBO $= \sum_{n,m=0}^{N-1}\sum_{k,j=0}^{K-1} a_{Kn+k} \tilde{Q}_{Kn+j,Km+j} a_{Km+j}$

Lagrangian relaxation: move the constraint into the objective function with penalty coefficient ξ.

(iii) Proposed Algorithm

Main idea

- We combined multiple binary classifiers to construct a multi-class classifier.
- If the binary classifiers output inconsistent results, our proposed algorithm **uses energy E as a priority of classifiers** to determine the result.
  - Smaller energy = Larger margin between two classes = Better generalization = **Higher reliability**
- We implemented this idea in two approaches: **one-versus-rest** and **one-versus-one**.

One-versus-rest

- **"Winner-takes-all"**
- Need to train N_{class} classifiers

Classifier	Output
#1 (C1 vs rest)	Not C1
#2 (C2 vs rest)	C2
#3 (C3 vs rest)	Not C3
Result	Class 2

Example of inconsistent results

Classifier	Output	Energy
#1 (C1 vs rest)	C1	-200
#2 (C2 vs rest)	C2	-240
#3 (C3 vs rest)	Not C3	
Result	Class 2	

- **When the results are inconsistent, #2 classifier has the lowest energy (-240). Choose the output of #2 as the prediction result.**

$$t_{predict} = t_{\mathrm{argmin} E_{C_{n_c}}}$$

Where $n_c \in \{1, \cdots, N_c\}$, $E_{C_{n_c}}$ is the energy of classifier $\#n_c$.

One-versus-one

- **"Max-wins"**
- Need to train $N_{class}(N_{class}-1)/2$ classifiers

Classifier	Output
#1 (C1 vs C2)	C2
#2 (C1 vs C3)	C1
#3 (C3 vs C2)	C2
Result	Class 2

Example of inconsistent results

Classifier	Output	Energy
#1 (C1 vs C2)	C1	-190
#2 (C1 vs C3)	C3	-100
#3 (C3 vs C2)	C2	-260
Result	Class 2	

- **When the results are inconsistent, #3 classifier has the lowest average energy (-260). Choose the output of #3 as the prediction result.**

$$t_{predict} = t_{\mathrm{argminavg} E_{C_{n_c}}}$$

Where $n_c \in \{1, \cdots, N_c(N_c-1)/2\}$, $E_{C_{n_c}}$ is the energy of classifier $\#n_c$.

(iv) Evaluation

Experiment

- We evaluated our multi-class qSVM (one-versus-rest) using Digital Annealer (DA) and simulated annealing (SA) [4]. The parameters is in the tables right-side.
- For comparison, we implemented cSVM using sklearn.
- We used rbf kernel ($\boldsymbol{rbf(X_n, X_m) = e^{-\gamma\|X_n - X_m\|^2}}$) in the experiments.

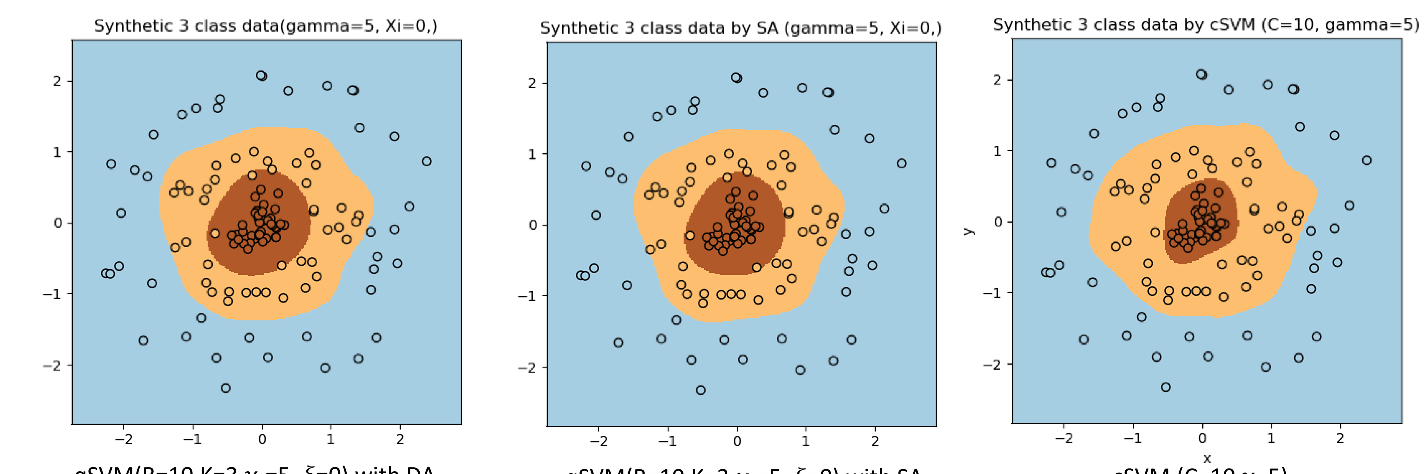
DA parameter	
Solver	FujitsuDA2PTSolver
Number iterations	2000000
Number replicas	26

SA parameter	
start temperature	10
End temperature	0.2
Cooling rate	0.97
Start repeat number	1000
Repeat number rate	1.0

Experiment 1 : Synthetic data

- Training data generated by $\mathbf{X_n} = \boldsymbol{r^n}\begin{pmatrix}\cos\phi_n \\ \sin\phi_n\end{pmatrix} + \begin{pmatrix}s_n^x \\ s_n^y\end{pmatrix}$.
- While qSVM showed slightly lower accuracy, we can see the same trends among all three methods.

	Accuracy
qSVM with DA	0.975
qSVM with SA	0.975
cSVM	1.000



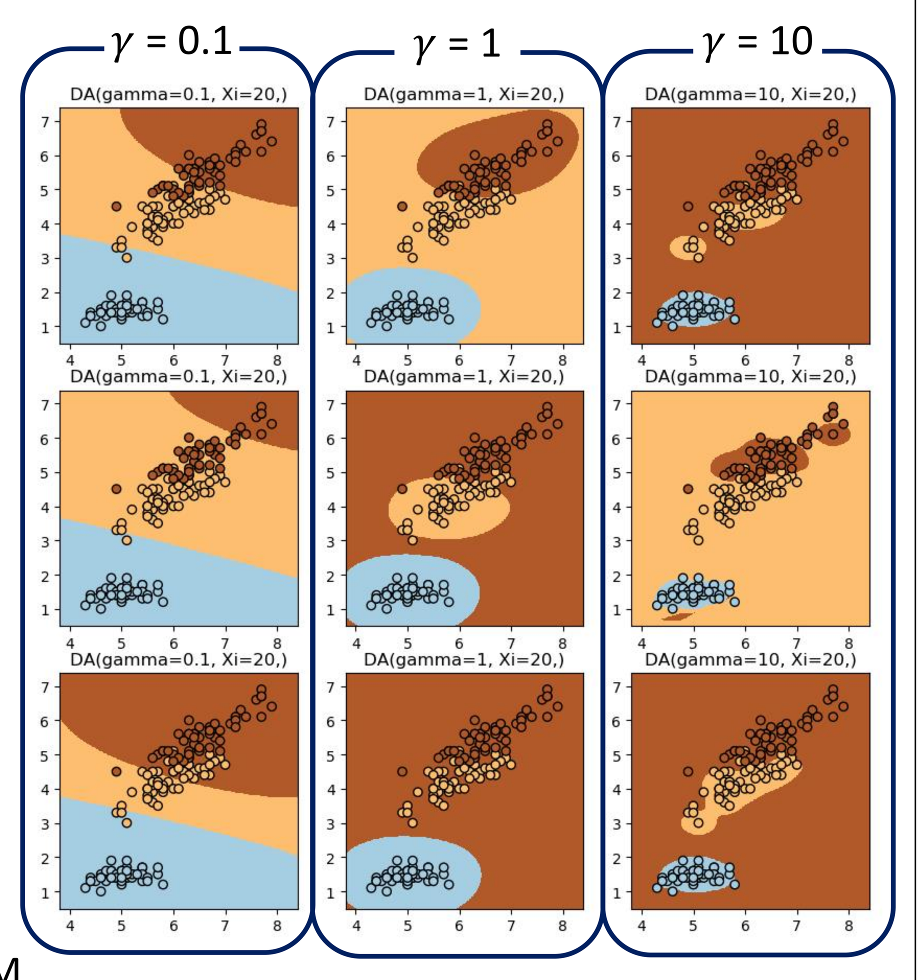
qSVM(B=10,K=3,γ =5, ξ=0) with DA　　qSVM(B=10,K=3,γ =5, ξ=0) with SA　　cSVM (C=10,γ=5)

Experiment 2 : IRIS

- We chose two variates as training data.
- Due to the randomness of DA, we performed the experiments on the same condition for three times.
- Three different colors represent three classes, and their boundaries are the decision boundaries.

Accuracy of qSVM with DA

	γ = 0.1	γ = 1	γ = 10
1	**0.853**	**0.947**	0.813
2	0.713	0.900	0.847
3	0.807	0.667	**0.900**



qSVM (B=10, K=3, ξ=20,γ=0.1, 1, 10) with DA

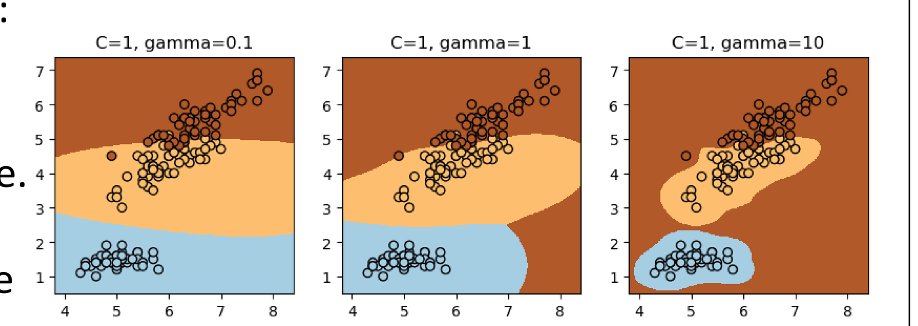
Discussion

- When γ = 0.1, the decision boundaries are broad and underfit the data.
- When γ = 1, qSVM separate data well with high generalization performance.
- When γ = 10, the decision boundaries fit the data better and tend to overfit the data.

Comparison with cSVM

- qSVM and cSVM have similar classification results and same effect of increasing or decreasing γ.
- The difference between qSVM and cSVM may be caused by the following reasons:

1. Binary encoding of real numbers.
2. Unguaranteed best results, as DA can not find the optimal solution every time.
3. Incorrespondence of C in qSVM and cSVM, as qSVM cannot adjust the range of C.



cSVM (C=1, γ=0.1, 1, 10) using sklearn

(v) References

[1] P. I. Bunyk, et al. "Architectural considerations in the design of a superconducting quantum annealing processor." IEEE Transactions on Applied Superconductivity 24.4 (2014): 1-10.
[2] M. Sao, et al. "Application of Digital Annealer for Faster Combinatorial Optimization." FUJITSU SCIENTIFIC & TECHNICAL JOURNAL 55.2 (2019): 45-51.
[3] D. Willsch, et al. "Support vector machines on the D-Wave quantum annealer." Computer Physics Communications 248 (2020): 107006.
[4] P. JM. Van Laarhoven and E. HL. Aarts. "Simulated annealing." Simulated annealing: Theory and applications. Springer, Dordrecht, 1987. 7-15.

ISC High Performance 2020, June 21-25, Frankfurt, Germany