Robust Classification with Non-Convex Loss and Regularization designed for Adiabatic Quantum Computation

Vasil S. Denchev[1,2] William G. Macready and Team[3]
Hartmut Neven[1]

[1]Google

[2]Purdue University

[3]D-Wave Systems

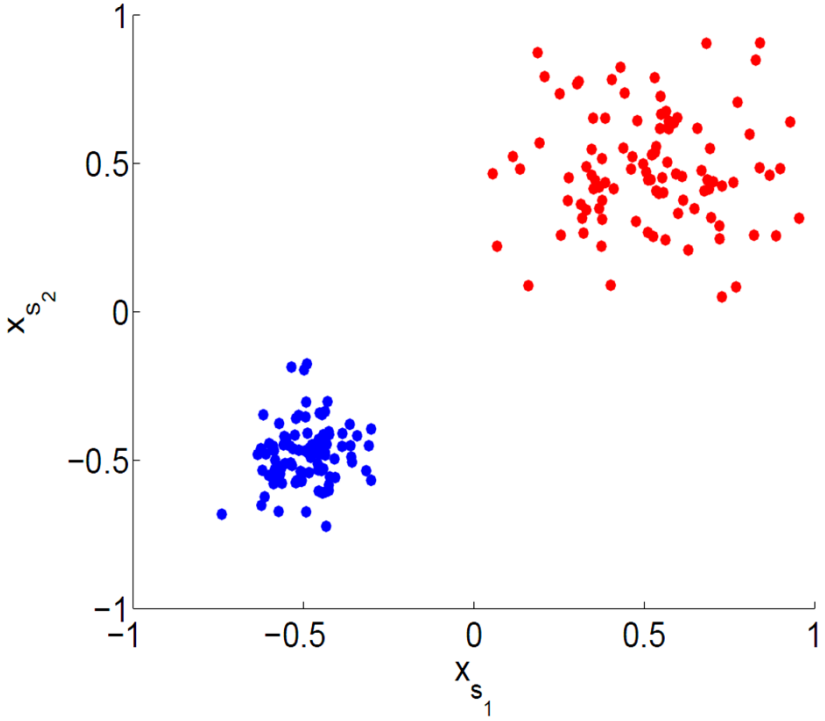
"I argue that all progress, both theoretical and practical has resulted from a single human activity: the quest for what I call good explanations."

David Deutsch, The Beginning of Infinity: Explanations That Transform the World, 2011

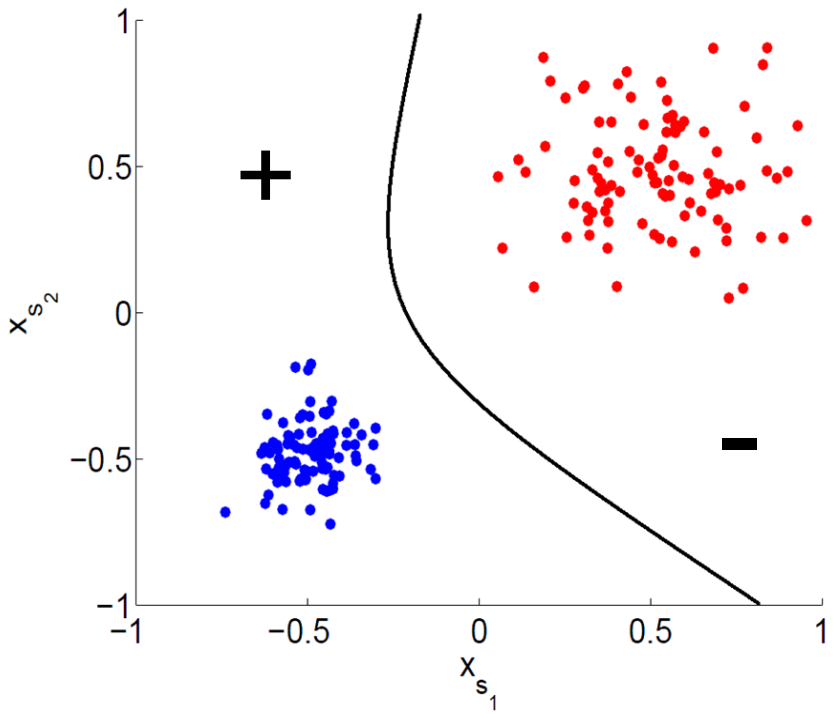
"The field of artificial (general) intelligence has made no progress because there is an unresolved philosophical problem at its heart: we do not understand how creativity works."

David Deutsch, The Beginning of Infinity: Explanations That Transform the World, 2011

Training a Binary Classifier



Training a binary classifier



Binary Classifier

$$x \mapsto y = H_{\boldsymbol{w}}(x)$$

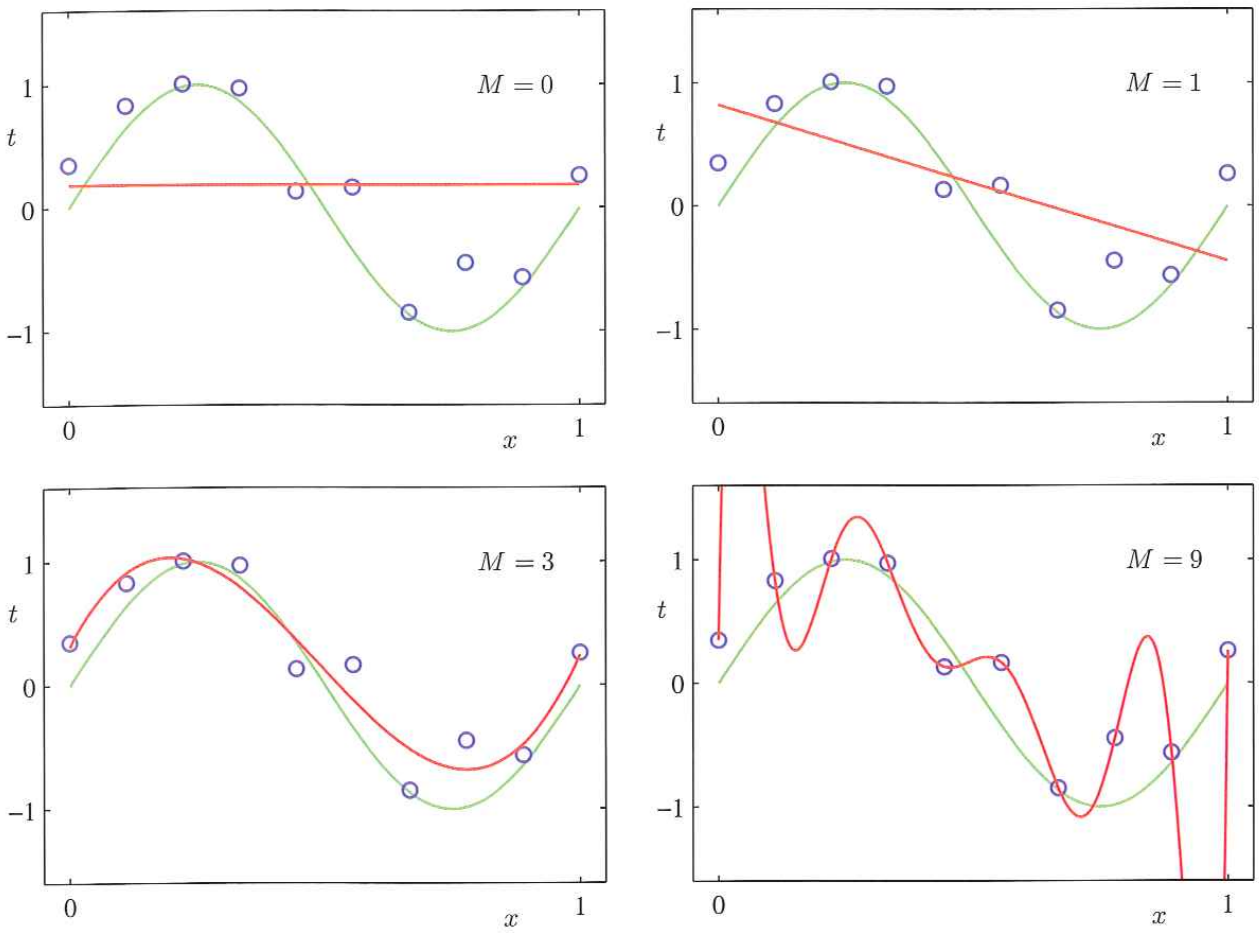
with $x \in \mathbb{R}^M$, $y \in \{-1, 1\}$ and $\boldsymbol{w} \in \mathbb{R}^M$

Many machine learning methods formulate training as optimization problem

- Goal is to form classifier: $y = H_{\boldsymbol{w}}(x)$
  that has minimal generalization error

- Given a set of S training examples $\{(x_s, y_s) | s = 1, \ldots, S\}$
- Training: $\boldsymbol{w}^{opt} = \arg\min_{\boldsymbol{w}} \left(L(\boldsymbol{w}) + R(\boldsymbol{w}) \right)$
- Loss $L(\boldsymbol{w})$ controls how well the classifier separates the classes
- Regularization $R(\boldsymbol{w})$ controls the complexity of the classifier

Regularization



From Bishop, Pattern Recognition and Machine Learning, 2006

Natural choices of loss and regularization render training formally NP-hard

For computational efficiency typically a convex objective $L(\boldsymbol{w}) + R(\boldsymbol{w})$ is constructed

But that comes at a cost

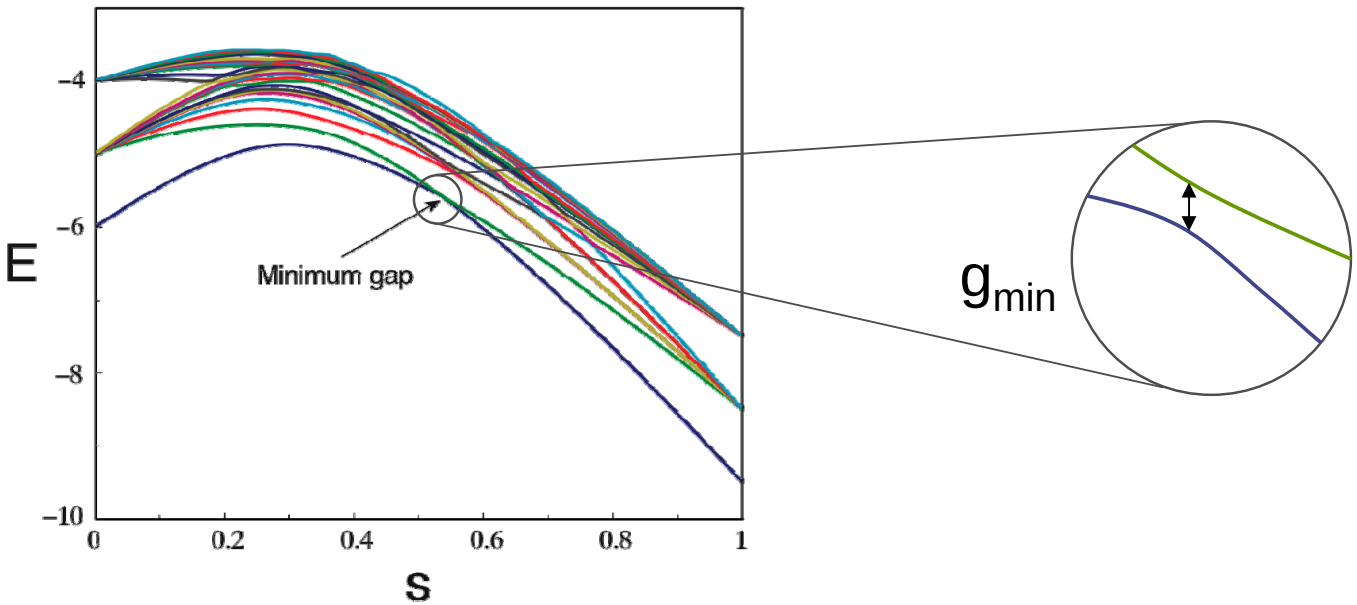
- Convex losses are not as robust to incorrectly labeled examples
- Solutions obtained with convex regularization are not as sparse
- Indication that tighter generalization bounds can be obtained with non-convex losses
- Minimum of objective function does not correspond to minimal training error

Emboldened by emerging quantum hardware ...

We studied non-convex versions of $L(\boldsymbol{w})$ and $R(\boldsymbol{w})$ rendering the problem formally NP hard

Treated training as an integer programe working with weights $\boldsymbol{w}$ of low bit depth

Adiabatic quantum optimization (AQO)



$H(t)$: slowly varying Hamiltonian for evolution from $t = 0$ to $t = T$

$H(0) = H_B$: initial Hamiltonian with known and easily preparable ground state

$H(T) = H_P$: Hamiltonian whose ground state encodes the solution to a given instance of an optimization problem

System remains in ground state during quantum evolution:
$i\frac{d}{dt}|\psi(t)\rangle = H(t)|\psi(t)\rangle$

Implementation of AQO in the D-Wave architecture

D-Wave implements the Ising model in hardware:

$$H_{Ising} = \sum_{i,j} J_{ij} \sigma_z^{(i)} \sigma_z^{(j)} + \sum_i h_i \sigma_z^{(i)}$$

$J_{i,j}$ are coupling strengths between pairs of qubits. h_i are biases of individual qubits. σ_z are Pauli matrices.

Finding the ground state of H_{Ising} amounts to solving a QUBO

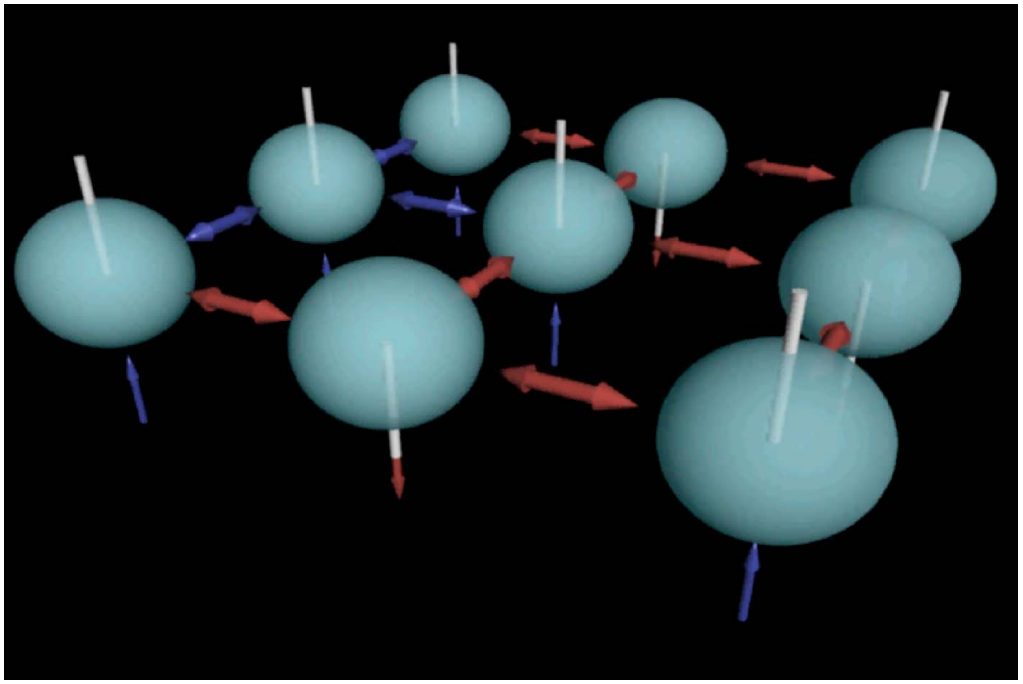
$$\boldsymbol{w}^\star = \arg\min_{\boldsymbol{w}} \left\{ \boldsymbol{w}' Q \boldsymbol{w} \right\}$$

Equivalent to Weighted MAX-2-SAT

Simplest many body system!

Illustration of computation in D-Wave processor

Qubits connected via Chimera graph



Linear binary classifier

$$y = H_{\boldsymbol{w}}(x) = \mathrm{sign}\left(\sum_{i=1}^{N} w_i h_i(x) + b\right)$$

$x \in \mathbb{R}^M$: input patterns to be classified

$y \in \{-1, 1\}$: output of the classifier

$h_i : x \mapsto \{-1, +1\}$: weak classifiers

$w_i \in [0, 1]$: set of weights to be optimized

$b \in \mathbb{R}$ is the bias

$H(x)$: strong classifier

Weights with low bit depth are sufficient, are even beneficial

Argument why weights with low bit depth are sufficient

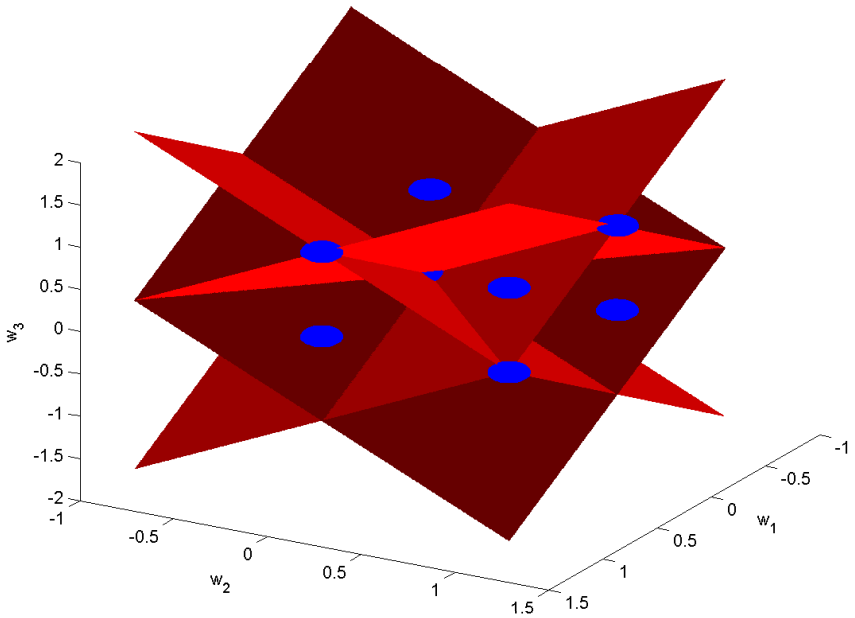
Each training example creates hyper plane in weight space

Those create solution regions

As long as each region contains a least one vertex of the hyper lattice we are good

Lower bound: $bits \geq \log_2(f) + \log_2(e) - 1$

Details in arXiv: quant-ph/0811.0416



Training with Convex Loss and Non-Convex Regularization

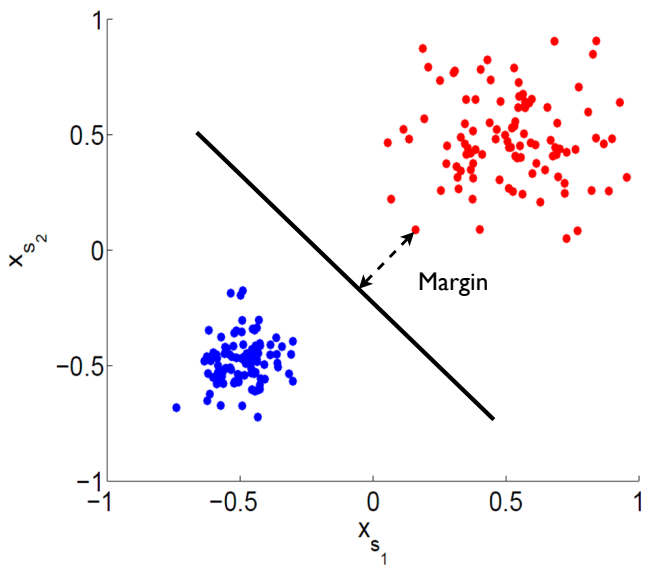
$$L(w) = \sum_{s=1}^{S} L_{square}(m_s)$$

$$L_{square}(m_s) = (m_s - 1)^2$$

$m_s = y_s \left(w^T x_s + b\right)$ is the *margin* of example s

$$R(w) = \lambda \parallel w \parallel_0$$

$O(N)$ qubits are spent representing the weights and threshold



Efficient use of qubits allows for flexible (heuristic) extensions

Assume first round of optimization yields

$$y = \text{sign} \quad \sum_{t=1}^{T} w_t h_t(x) + b$$

Assume $T < Q$

Re-run optimization by adding new weak classifiers $\{h_i(x) | i = 1, \ldots, Q - T\}$ selected via boosting

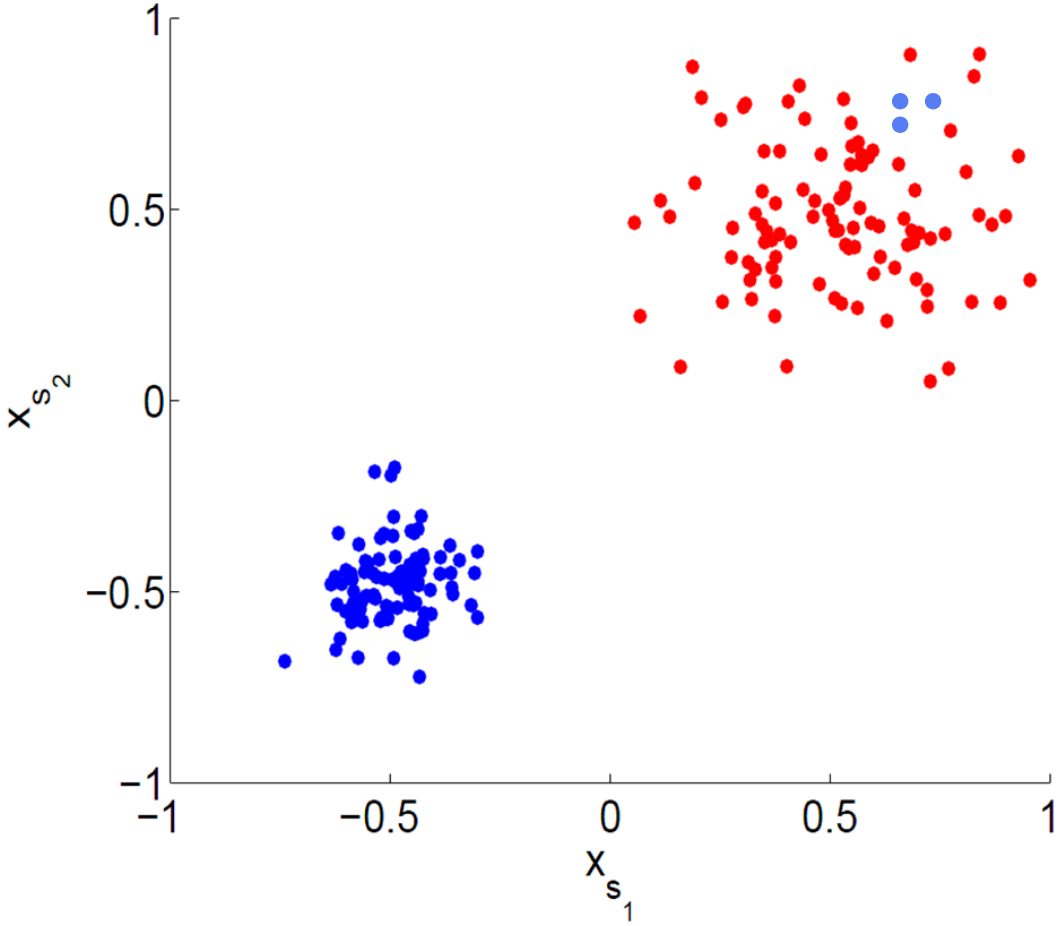
If large classifier is needed with $T > Q$

Freeze weights obtained in previous optimization rounds and concatenate

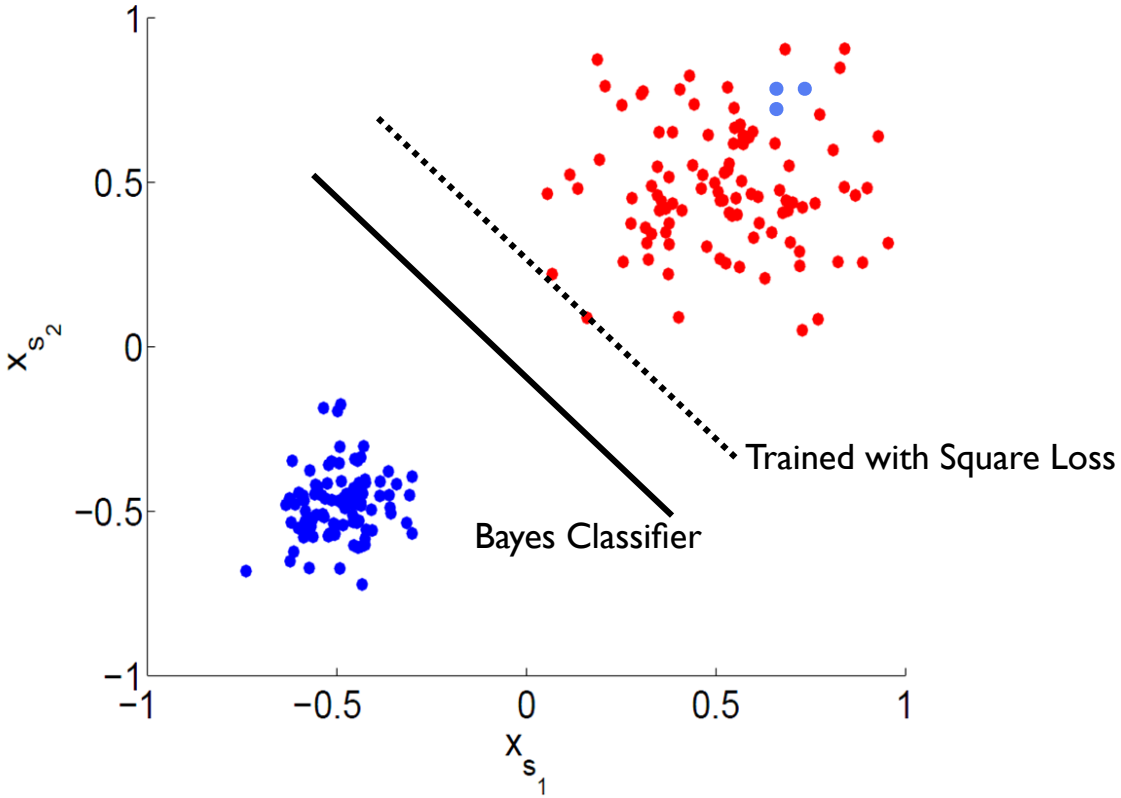
$$y = \text{sign} \left(\underbrace{\sum_{t=1}^{T_{prev}} w_t h_t(x)}_{frozen} + \sum_{t=1}^{Q} w_t h_t(x) + b \right)$$

Details in arXiv: quant-ph/0912.0779

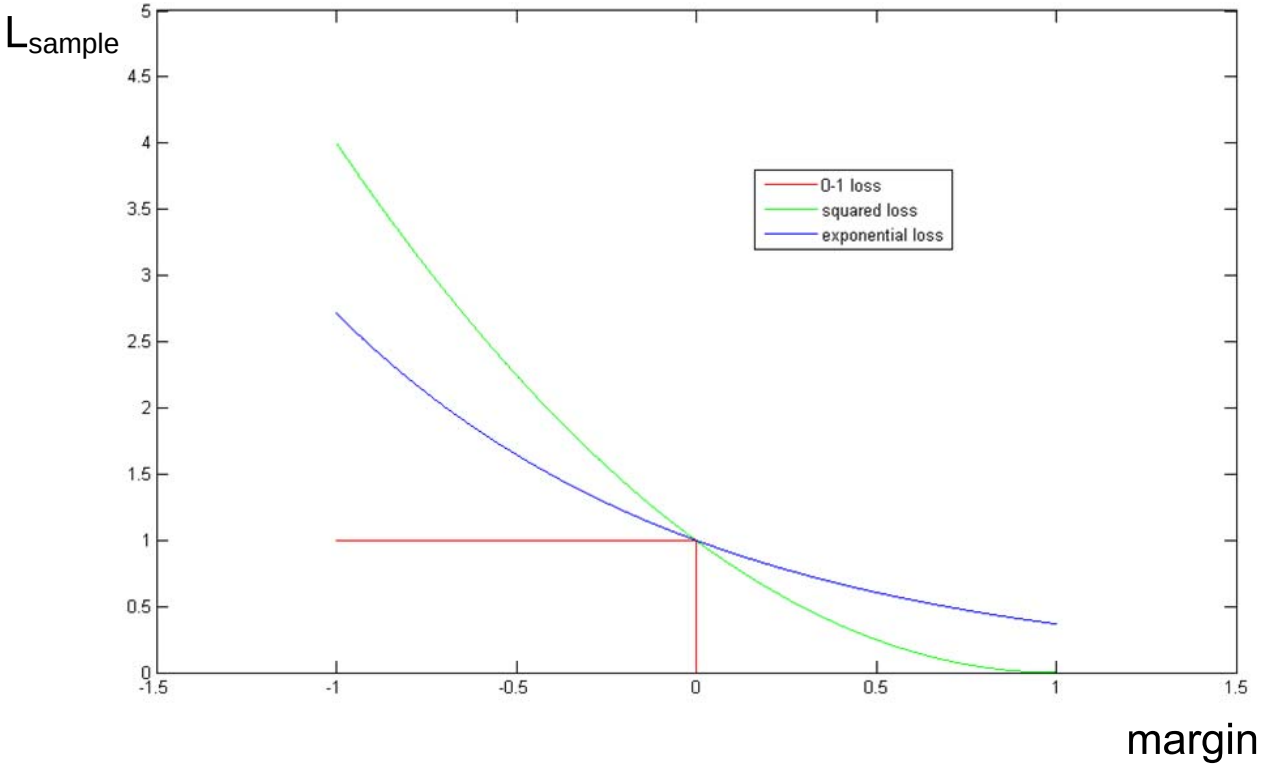
Dealing with Outliers



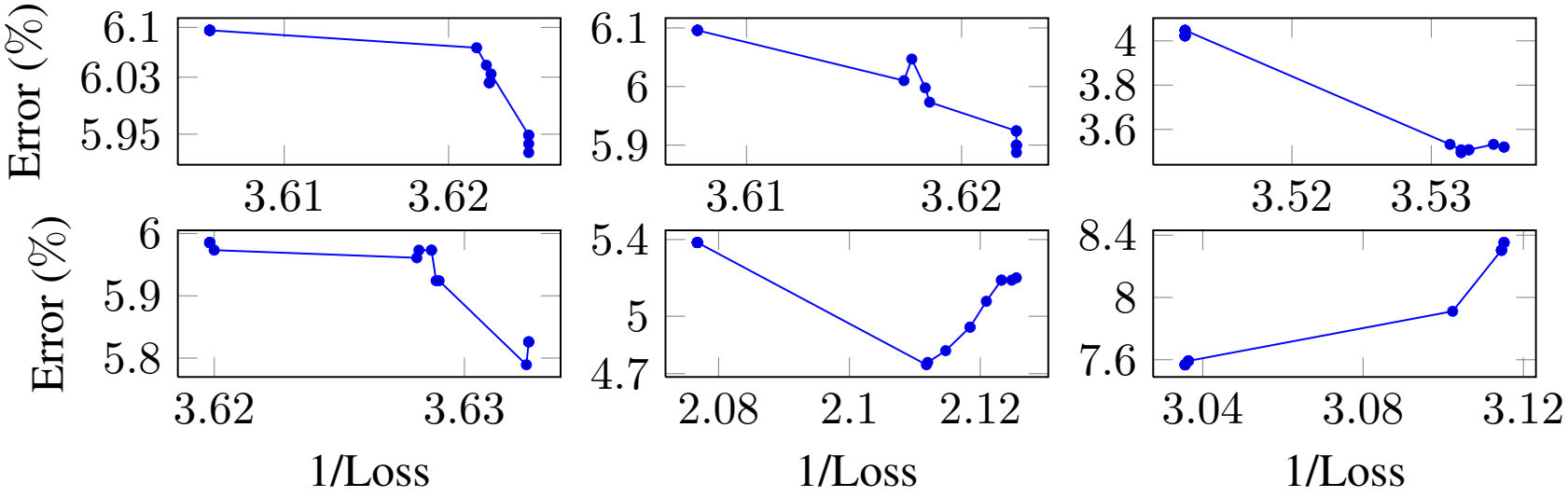
Convex losses are sensitive to label noise



Loss functions



Non-monotonous relationship between objective value and training error



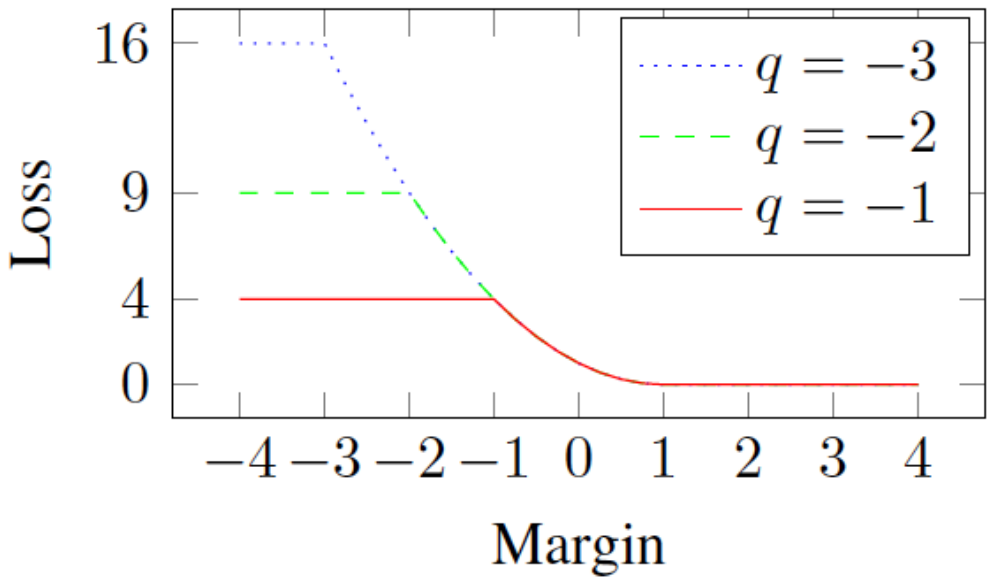
Training with Non-Convex Loss and Convex Regularization

$$L(w) = \sum_{s=1}^{S} L_q(m_s)$$

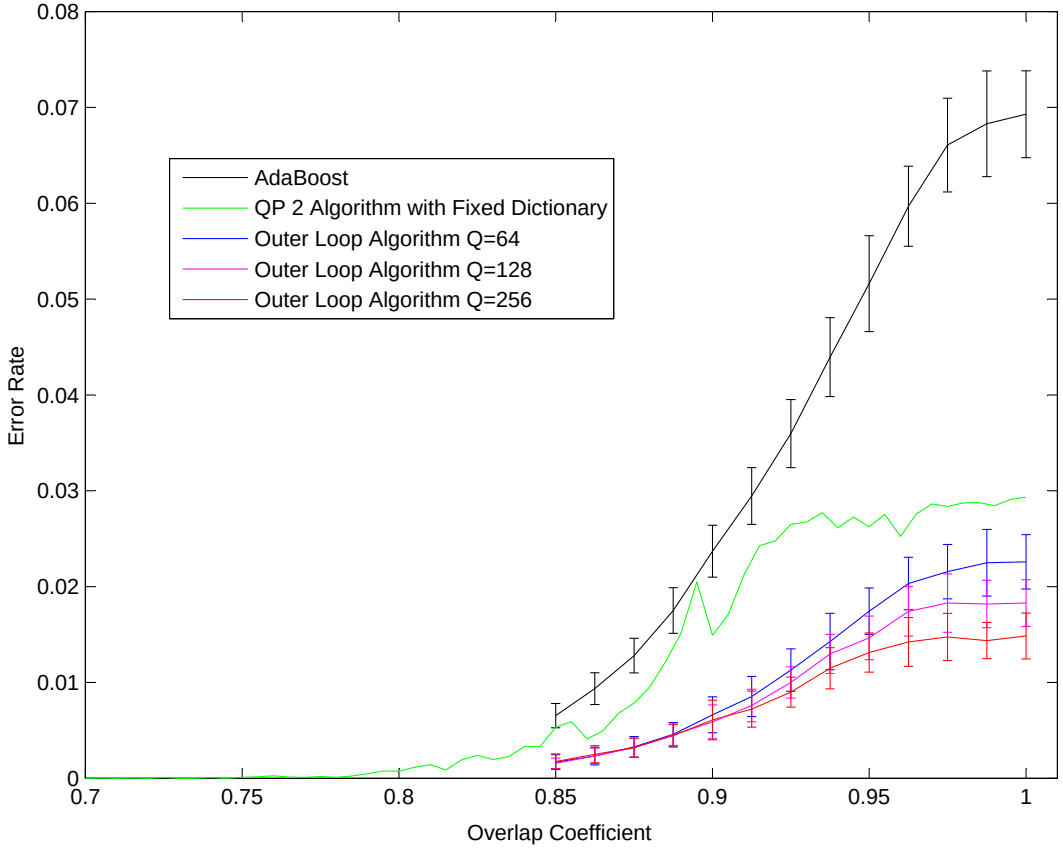
$$L_q(m_s) = \min\left((1-q)^2, (\max(0, 1-m_s))^2\right)$$

$$R(w) = \lambda \parallel w \parallel_2$$

In addition to qubits needed to represent w and b we need $O(S)$ ancillary qubits to represent non-convex loss!



Numerical experiments: non-convex regularization



	AdaBoost	**Outer Loop Q = 64**	**Outer Loop Q = 128**	**Outer Loop Q = 256**
Test Error	0.258 ± 0.006	0.254 ± 0.010	0.249 ± 0.010	0.246 ± 0.009
Weak Classifiers	257.8 ± 332.1	116.8 ± 139.0	206.1 ± 241.8	356.3 ± 420.3
Reweightings	658.9 ± 209.3	130.8 ± 65.3	145.6 ± 65.5	159.8 ± 63.7
Training Error	0.038 ± 0.054	0.185 ± 0.039	0.170 ± 0.039	0.158 ± 0.040
Outer Loops		11.9 ± 5.0	12.2 ± 4.6	12.6 ± 4.4

Improved generalization is expected from Vapnik-Chernovenkis theory

Vapnik-Chernovenkis dimension:

$$VC_H = 2(VC_{\{h_i\}} + 1)(T + 1)\log_2(e(T + 1))$$

$H(x) = \sum_{t=1}^{T} h_t(x)$
$VC_{\{h_i\}}$ is the VC dimension of dictionary

Weak $L0$-norm regularization with $\lambda < \frac{2}{N} + \frac{1}{N^2}$ produces classifier with lower VC bound while not increasing the training error

Hardware experiments: Training a detector with D-Wave's hardware

Car detector

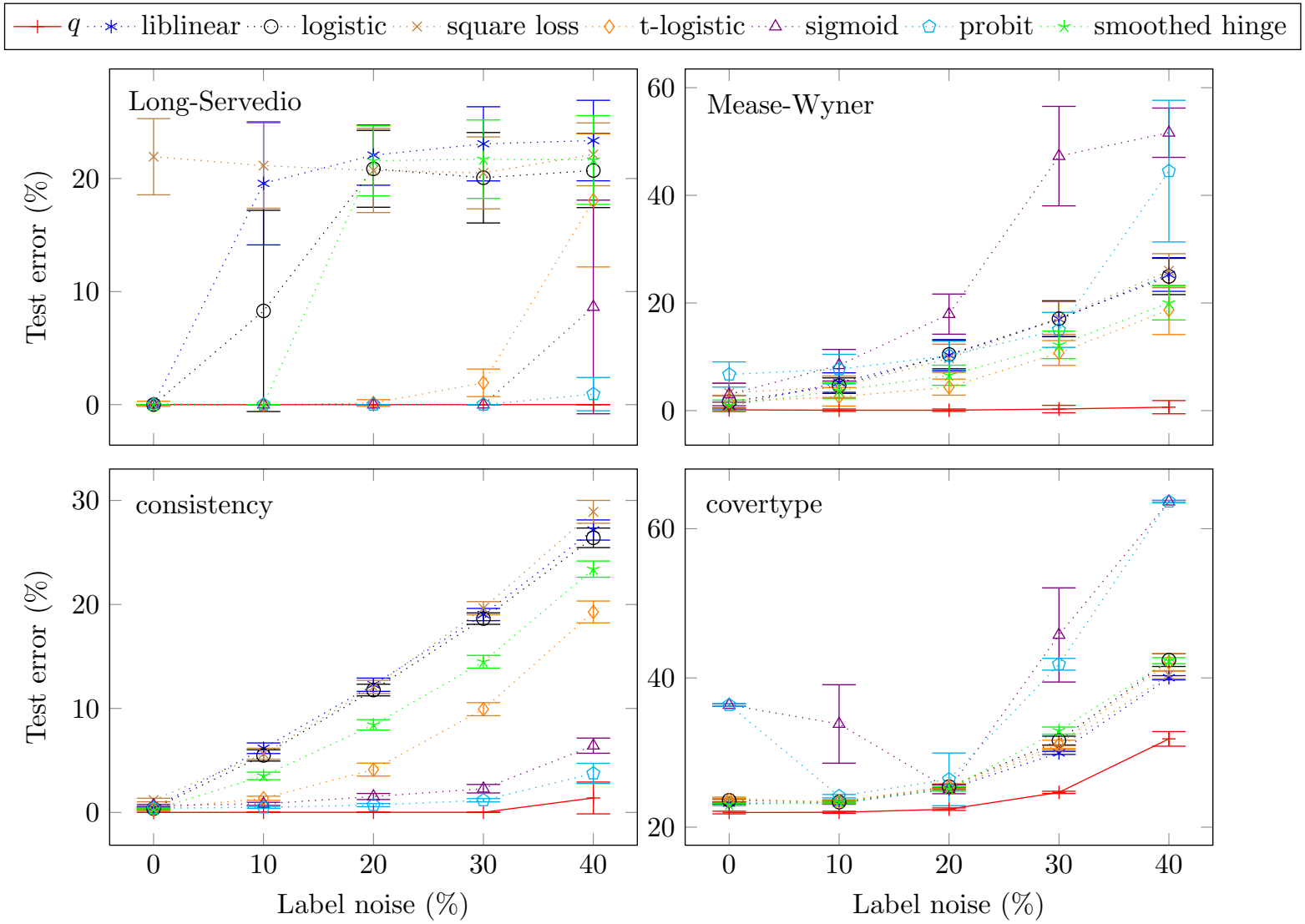
Detect cars in images

Training data: 20,000 images with city street scenes

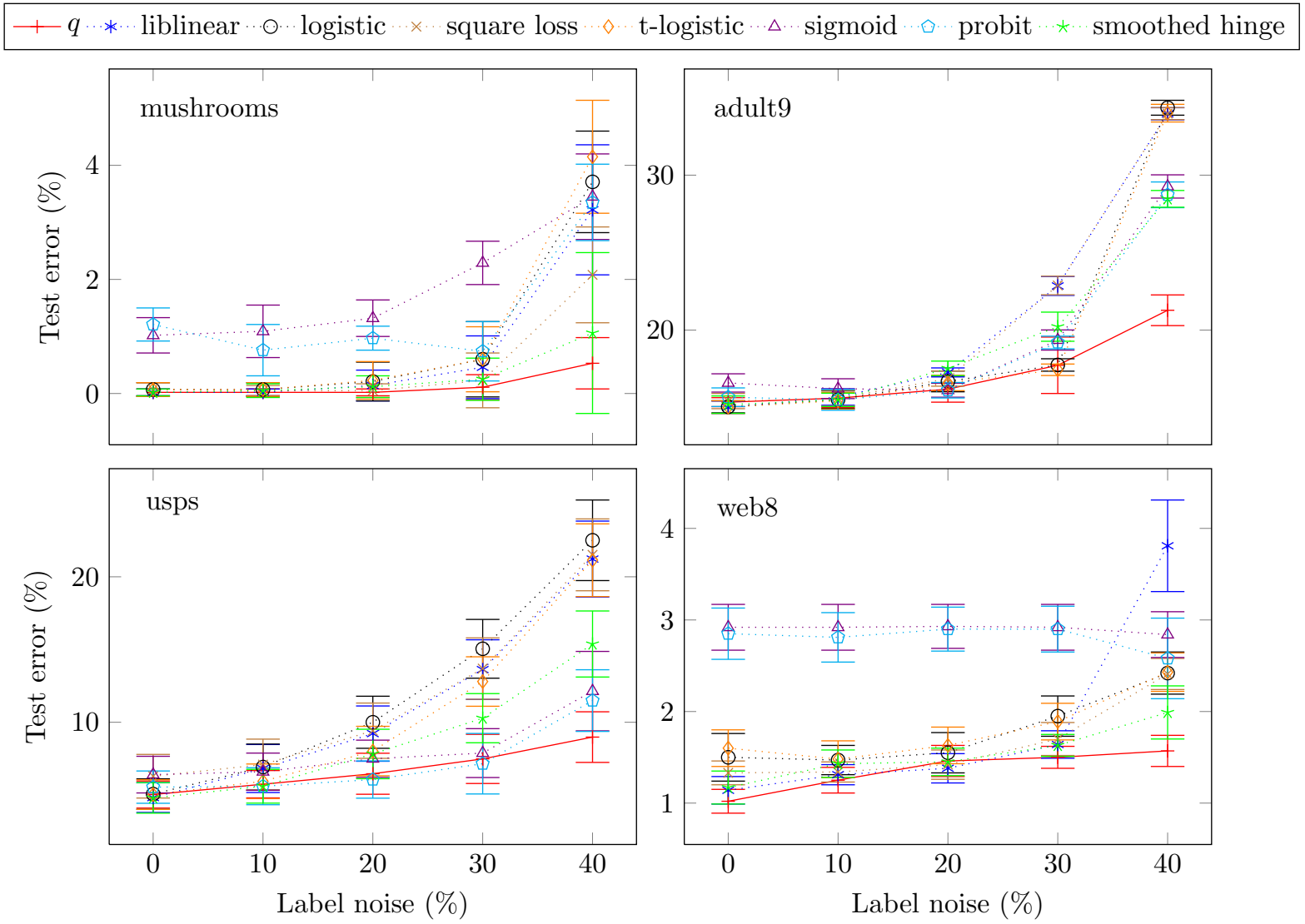
Calls to quantum hardware with 52-variable optimization problems

NIPS Demo 2009

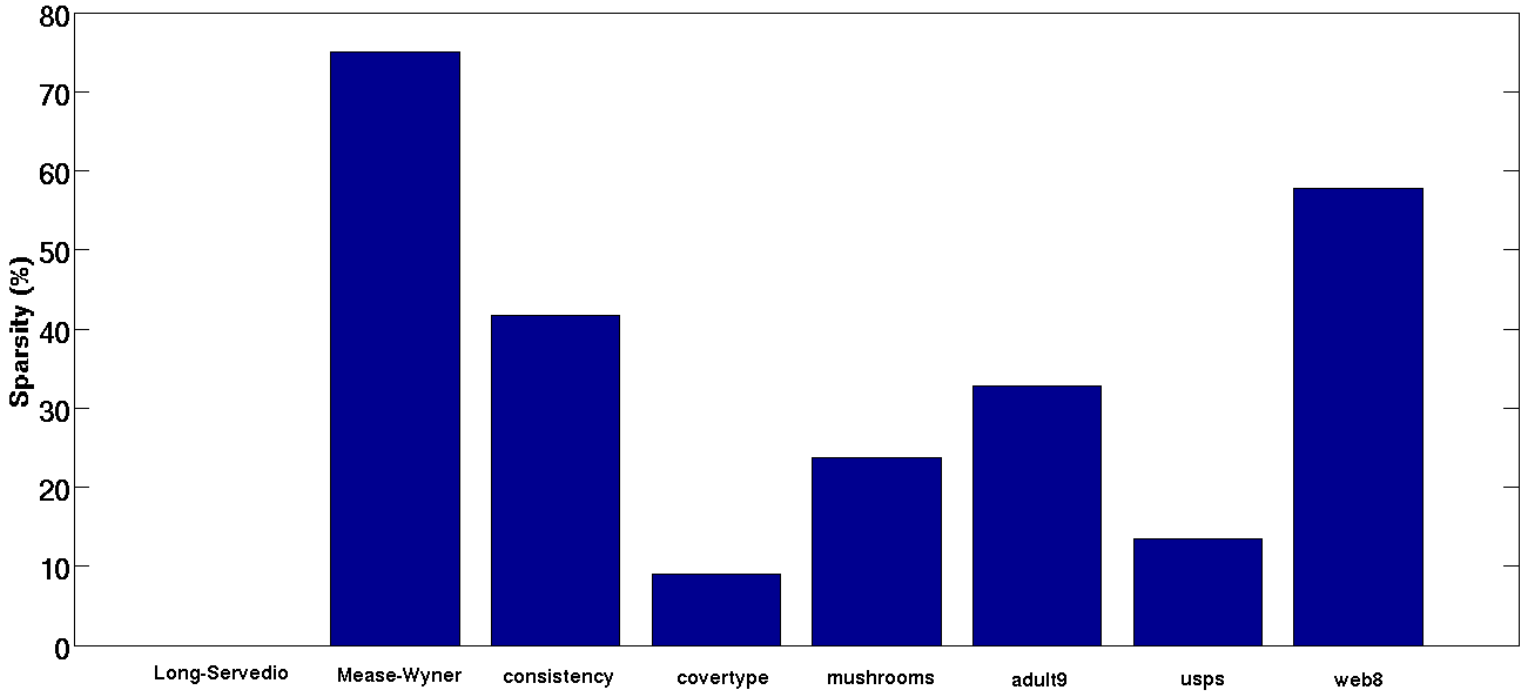
Numerical experiments: non-convex loss



Numerical experiments: non-convex loss



Training with low-precision discrete weights yields significant levels of sparsity



Challenges mapping to D-Wave hardware

Problems

Objective needs to be quadratic.

Solutions

Higher order interactions can be reduced to quadratic by introducing auxiliary qubits.

More qubits!

Qubit connectivity needs to adhere to Chimera graph.

By stringing physical qubits together to form logical qubits that can be arbitrarily connected.

More qubits!

Indications that this may not scale!

New approach!

Sculpt QUBO that can be represented by the hardware to fit samples from objective.

Optimization becomes heuristic. Narrow minima may be overlooked.

Parameter Warping

Will go away with better electronics.

Underexplored opportunity

Use D-Wave hardware as a sampling engine

Many machine learning methods are based on sampling from a probability distribution.

Non-zero temperature gives Boltzmann distribution:

$$P(\boldsymbol{s}) = \frac{\exp(-\beta E(\boldsymbol{s}))}{Z}$$

Open quantum systems analysis: How does the probability distribution look if the adiabatic condition is violated?

Native hardware mode is sampling!

Approximate solutions by AQO not well studied

> But post-PCP-theorem work indicates that approximate solutions can be used to optimally solve NP-hard problems in polynomial time (Dickson and Amin'11; Zuckerman, STOC'06)

Summary

Non-convex training of a binary classifier can be mapped to a QUBO format amenable to quantum optimization at "negative translation cost"

- Lower generalization error
- More compact classifiers
- Can better cope with outliers
- Less training cycles (when using boosting)

Experiments underway to run problem instances on D-Wave One quantum computer

Thank you!

Vasil S. Denchev[1,2] William G. Macready and Team[3]
Hartmut Neven[1]

[1]Google

[2]Purdue University

[3]D-Wave Systems