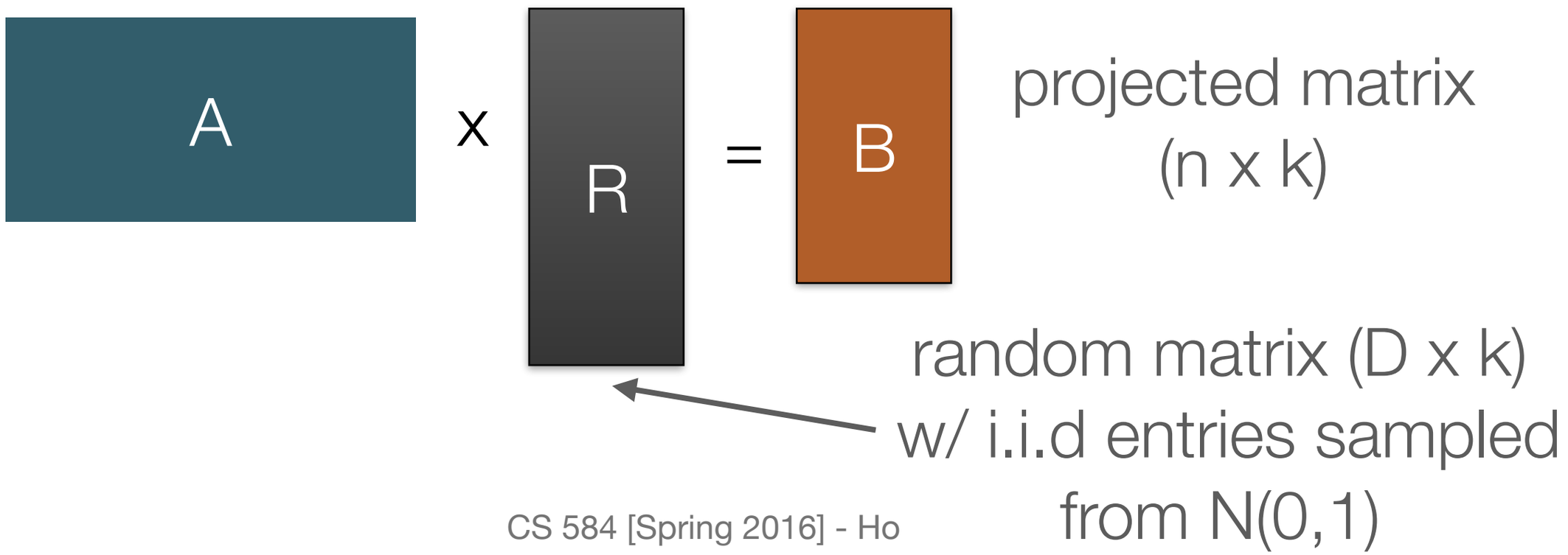
Coding for Random Projects

CS 584: Big Data Analytics

Material adapted from Li's talk at ICML 2014
(http://techtalks.tv/talks/coding-for-random-projections/61085/)

Random Projections for High-Dimensional Data

- Replace original data matrix A by B, where B = A x R
- B approximately preserves the Euclidean distance and inner products between any two rows of A
- Feed B into SVM or logistic regression solvers



Classification Experiment on Webspam Data: Very Sparse Random Projections

- Dataset: 350K text samples, 16 million dimensions, about 4000 nonzeros on average, 24GB disk space

- Task: Binary classification for spam vs non-spam

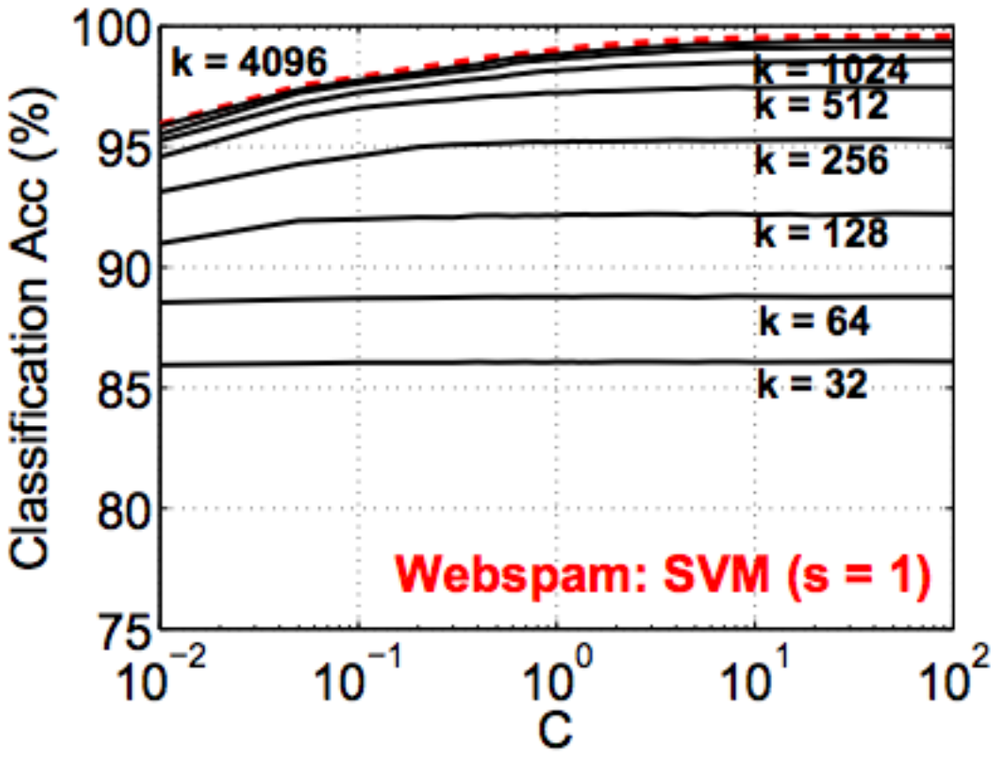
- Projection: Instead of sampling from normal, sample from sparse distribution parameterized by s

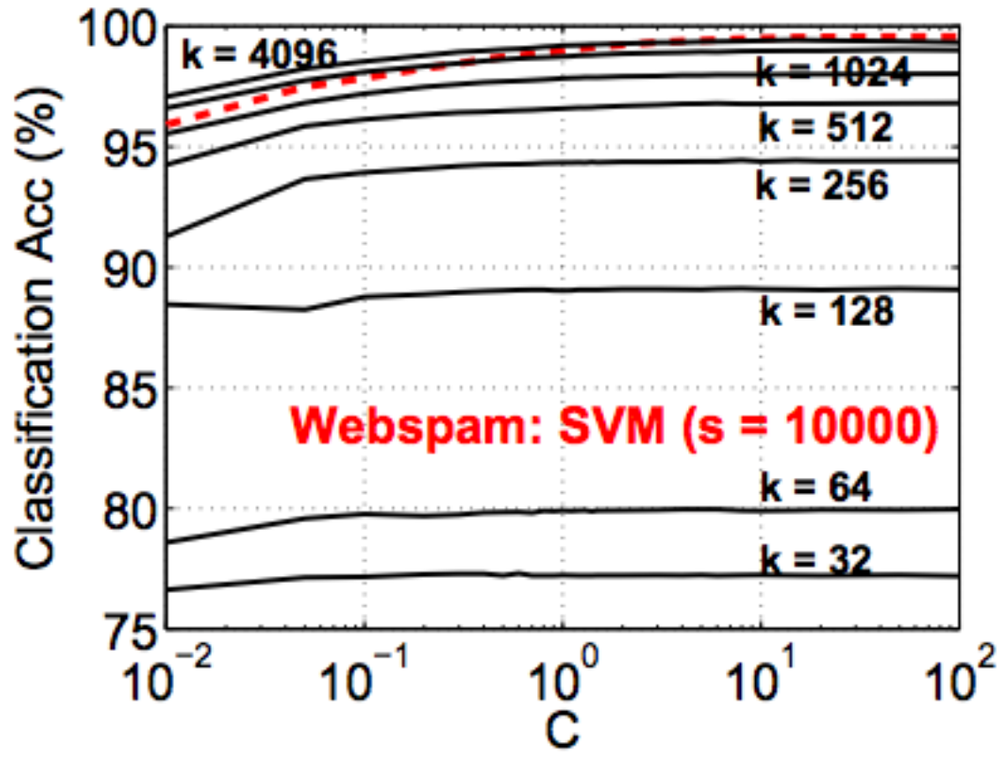
$$r_{ij} = \begin{cases} -1 & \text{with prob. } \frac{1}{2s} \\ 0 & \text{with prob. } 1 - \frac{1}{s} \\ 1 & \text{with prob. } \frac{1}{2s} \end{cases}$$

s = 100 means on average 99% of entries are 0

http://web.stanford.edu/group/mmds/slides2012/s-pli.pdf

Sparse Random Projection Results





- Need a large number of projections for high accuracy
- Random matrix can be very sparse (as long as k is large enough)

http://web.stanford.edu/group/mmds/slides2012/s-pli.pdf

Learning with Random Projections Summary

- Reasonable method when data are dense and inner product is a good kernel

- Usually needs high amount of projects to achieve highly accurate results

- Projected data are real-valued which are inconvenient for string and cannot be used for indexing

A coding scheme is necessary (need integers)!

Notations

- Random projections:

$$x = u \times R \in \mathbb{R}^k, \qquad y = v \times R \in \mathbb{R}^k$$

$$R = \{r_{ij}\}_{i=1,j=1}^{D,k}, \qquad r_{ij} \sim N(0,1)$$

- Assume input data has been normalized (one linear scan through the data)

$$||u||_2 = ||v||_2 = 1$$

- Joint distribution of (x_j, y_j) is bi-variant normal

$$\begin{bmatrix} x_j \\ y_j \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} \right), \rho = \sum u_i v_i$$

Prior Coding Scheme [Datar et. al 2004]

- Standard implementation in LSH packages
- Width must be pre-chosen => quantization is irreversible
- Random offset is needed for closed form version of collision probability

$$h^{(j)}_{w,q}(u) = \left\lfloor \frac{x_j + q_j}{w} \right\rfloor, \quad h^{(j)}_{w,q}(v) = \left\lfloor \frac{y_j + q_j}{w} \right\rfloor$$

$$q_j \sim \text{uniform}(0, w)$$

Prior Coding Scheme [Datar et. al 2004] (2)

- Collision probability

$$P_{w,q} = Pr(h^{(j)}_{w,q}(u) = h^{(j)}_{w,q}(v))$$

$$= \int_0^2 \frac{1}{\sqrt{d}} 2\phi\left(\frac{t}{\sqrt{d}}\right)\left(1 - \frac{t}{w}\right) dt$$

$d = ||u - v||_2^2 = 2(1 - \rho)$

standard normal pdf

Proposal: Uniform Quantization

- Drop the offset

$$h_w^{(j)}(u) = \left\lfloor \frac{x_j}{w} \right\rfloor, \quad h_w^{(j)}(v) = \left\lfloor \frac{x_j}{w} \right\rfloor$$

- Scheme is simpler than prior coding scheme

- For fixed w, this scheme is more accurate

- For a wide range of ρ values, the optimal w values for the scheme are relatively larger (means less number of bits to encode)

Uniform Quantization: Collision Probability

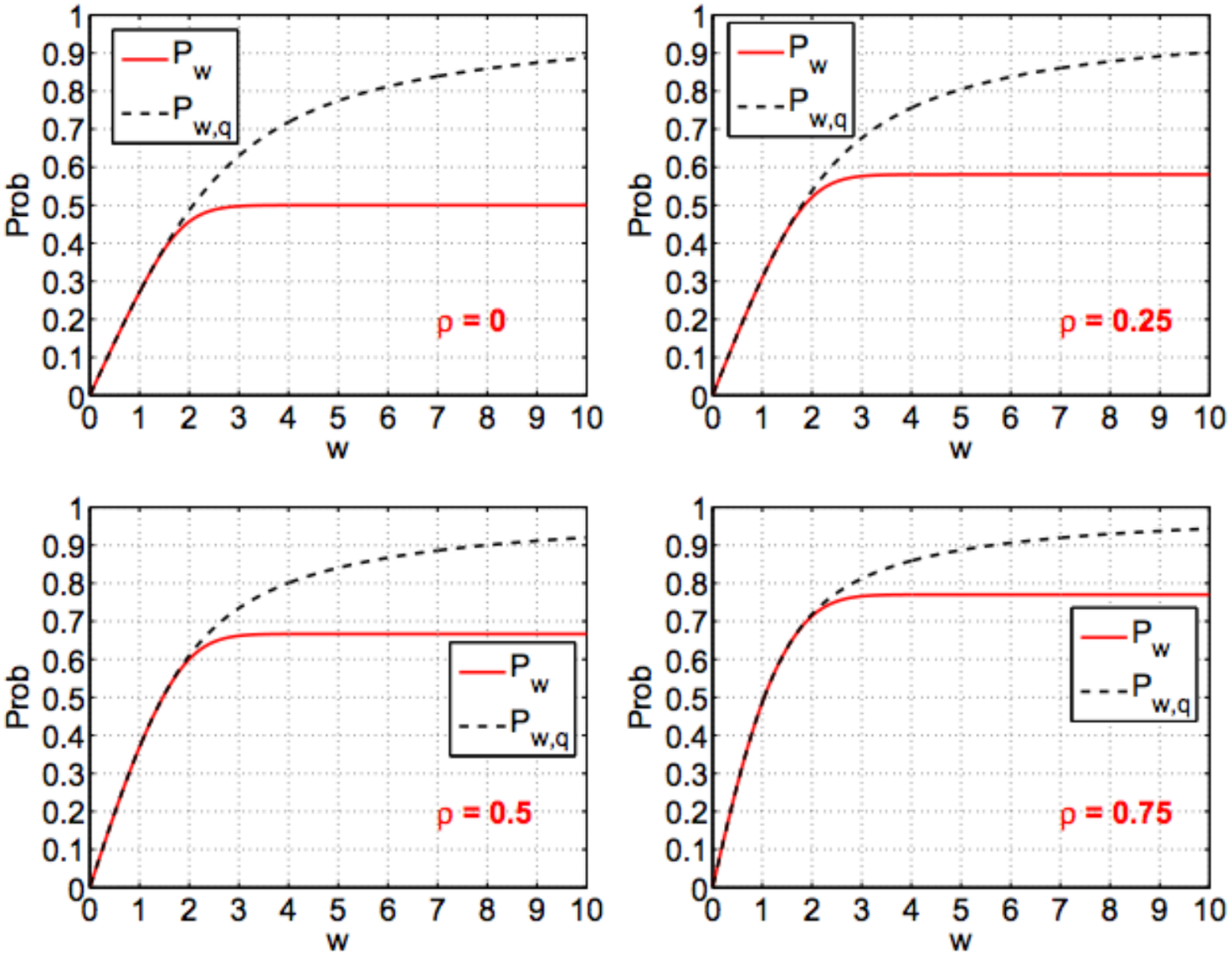
Basic requirement: collision probability should be monotonically increasing function of the similarity. It does not matter whether it has a closed form expression

Theorem:

$$P_w = 2\sum_{i=0}^{\infty}\int_{iw}^{(i+1)w}\phi(z)\times \tag{7}$$
$$\left[\Phi\left(\frac{(i+1)w-\rho z}{\sqrt{1-\rho^2}}\right)-\Phi\left(\frac{iw-\rho z}{\sqrt{1-\rho^2}}\right)\right]dz$$

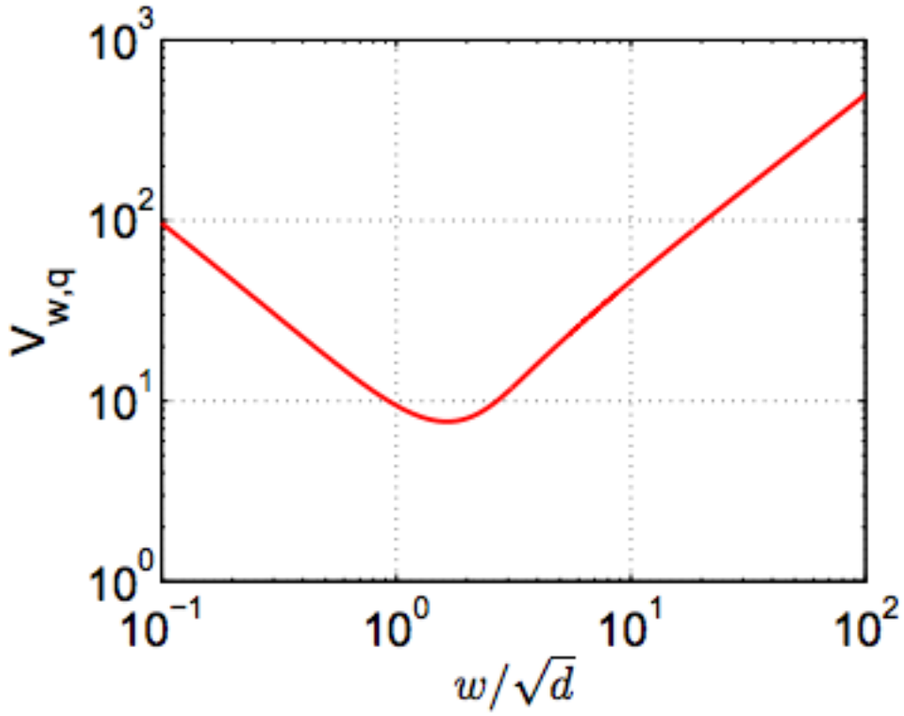
which is a monotonically increasing function of $\rho \geq 0$.

Collision Probability Comparison



- Difference becomes noticeable at w > 2
- Smaller collision probabilities than existing scheme
- Note bad behavior in previous scheme for orthogonal vectors approaches 1

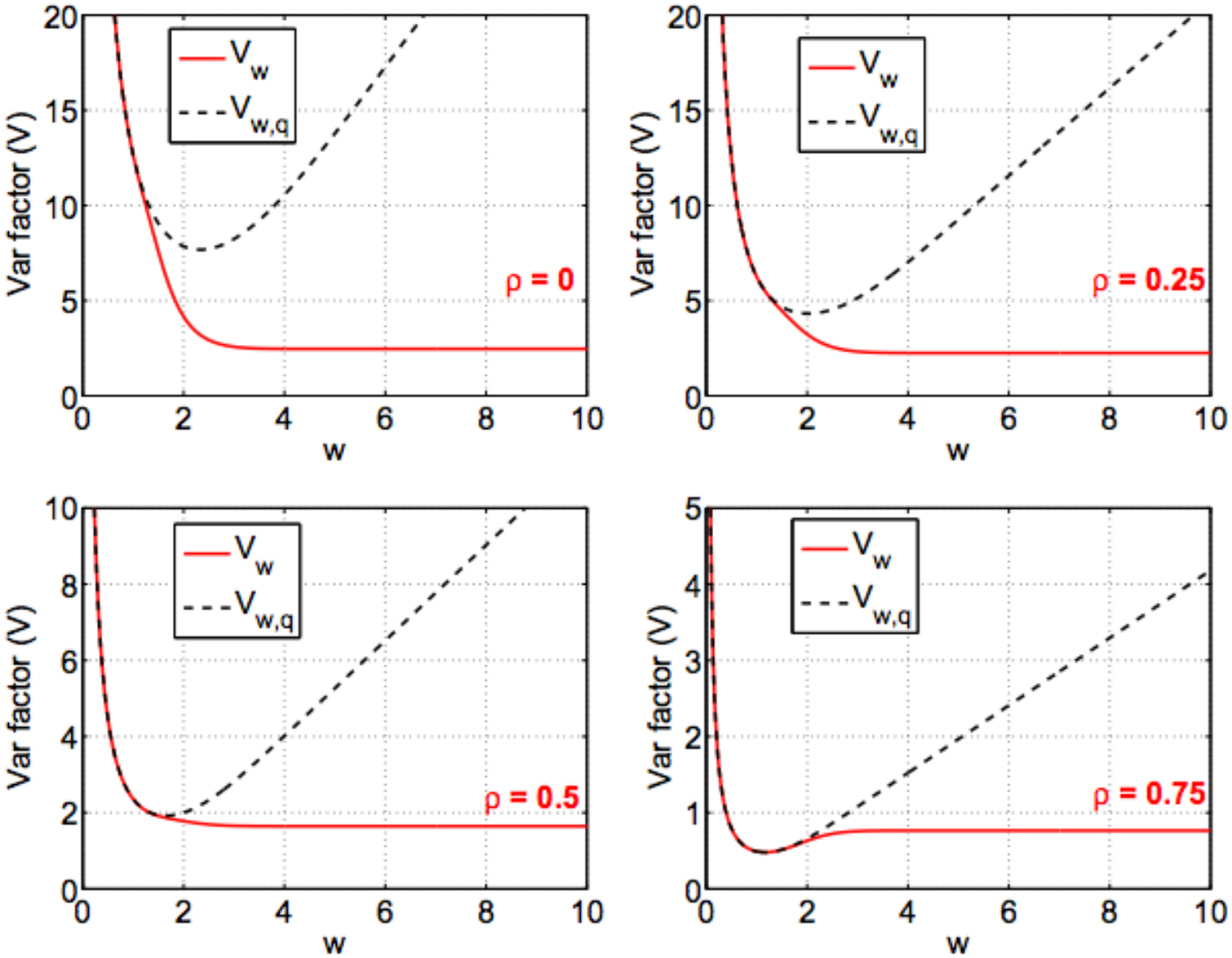
Variance of Existing Scheme



- Minimum variance of 7.6797 (quite large) is attained at 1.6476

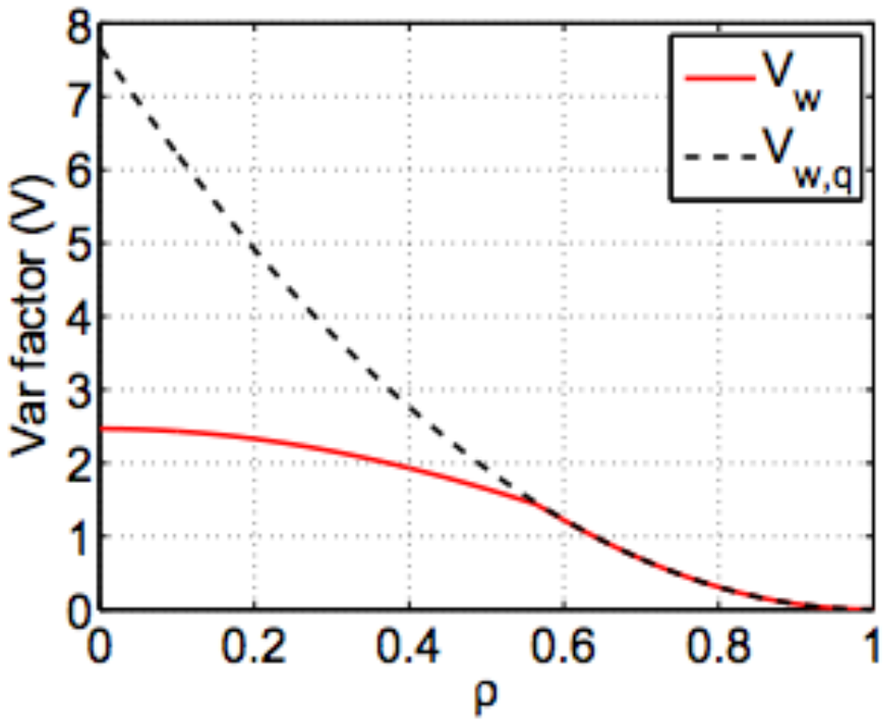
- Performance can be sensitive to choice of bin width - practical disadvantage

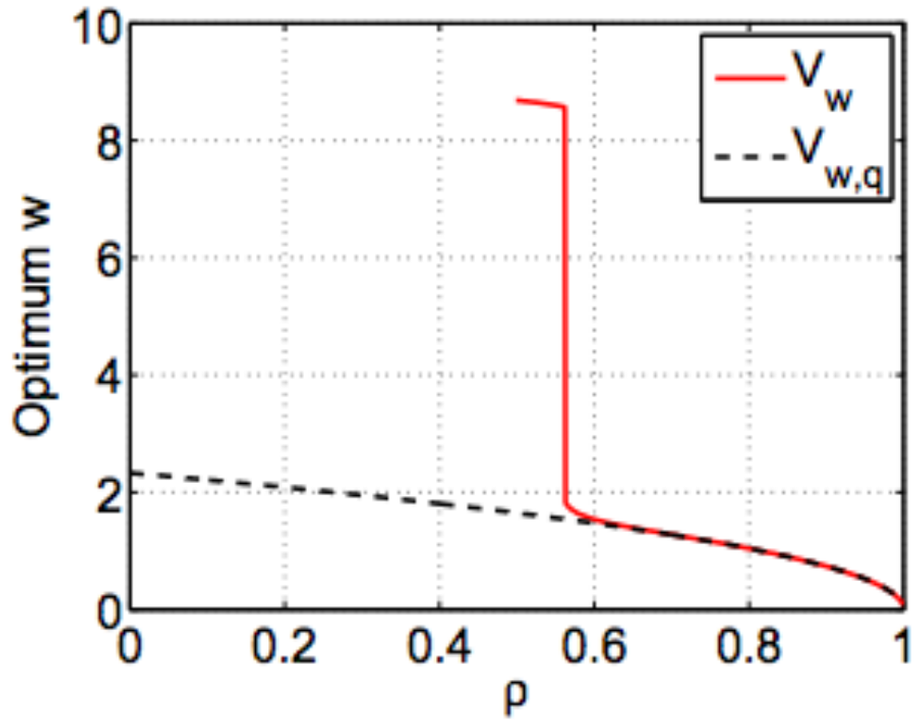
Variance Comparison at Fixed Bin Width w



- Variance can be significantly lower than existing scheme
- Performance of proposed scheme is not as sensitive to choice of w

Variance Comparison at Optimal Bin Widths w





- Uniform quantization variance is significantly lower at smaller similarity values

- Sufficient to use 1 bit (i.e., sign of project data) if the similarity is below 0.56

2-Bit Non-Uniform Coding Scheme

Motivation for developing non-uniform coding schemes

- In practice, we don't know similarity in advance and we often care about high similarities

- When $\rho > 0.56$, we might want to choose small w values (e.g., w < 1)

- However using a small w value will hurt the performance in low similarities

2-Bit Non-Uniform Coding Scheme (2)

- Quantize the projected data into four regions

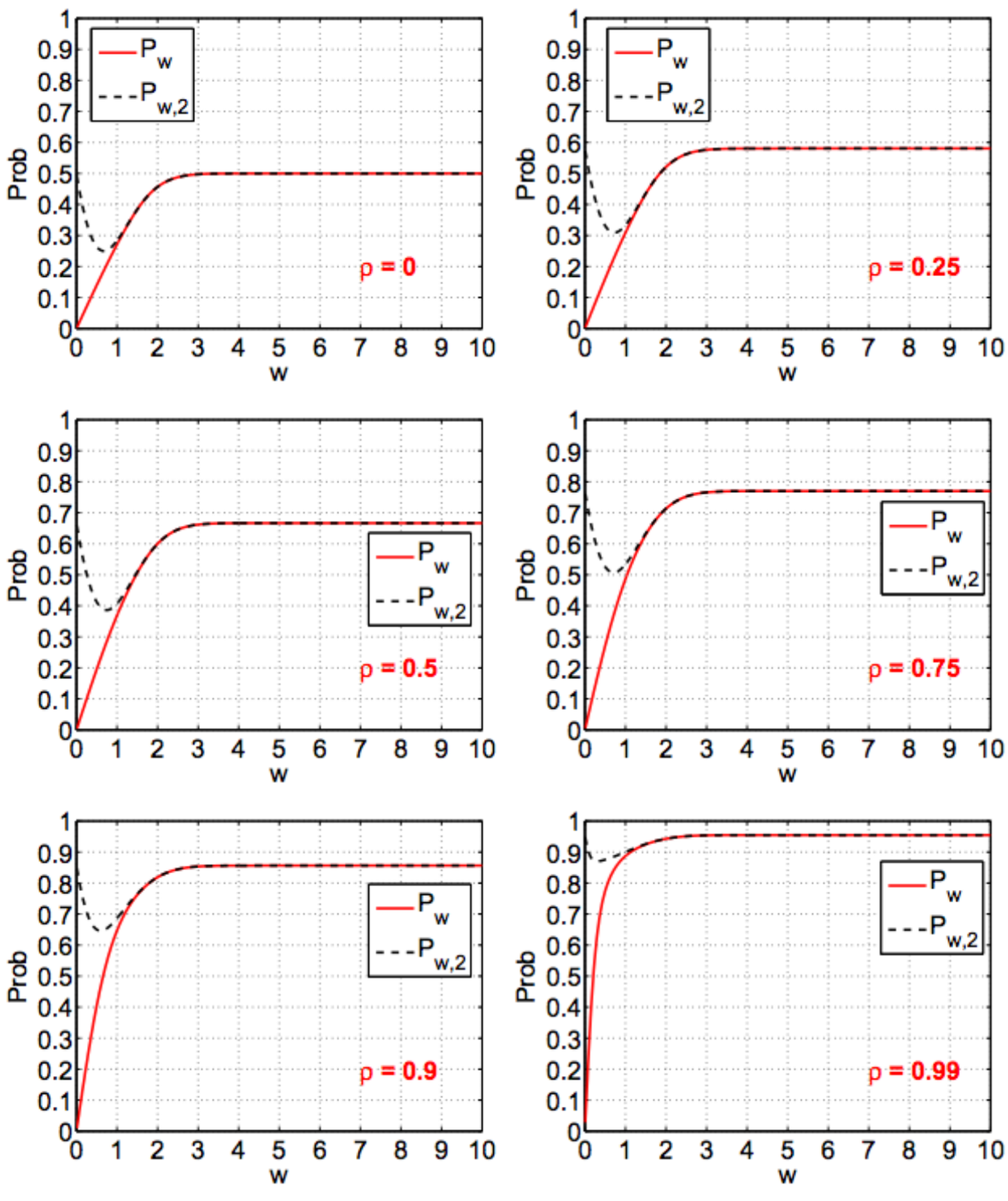
$$(-\infty, -w), [-w, 0), [0, w), [w, \infty)$$

- Collision probability

$$P_{w,2} = \mathbf{Pr}\left(h^{(j)}_{w,2}(u) = h^{(j)}_{w,2}(v)\right) \tag{13}$$

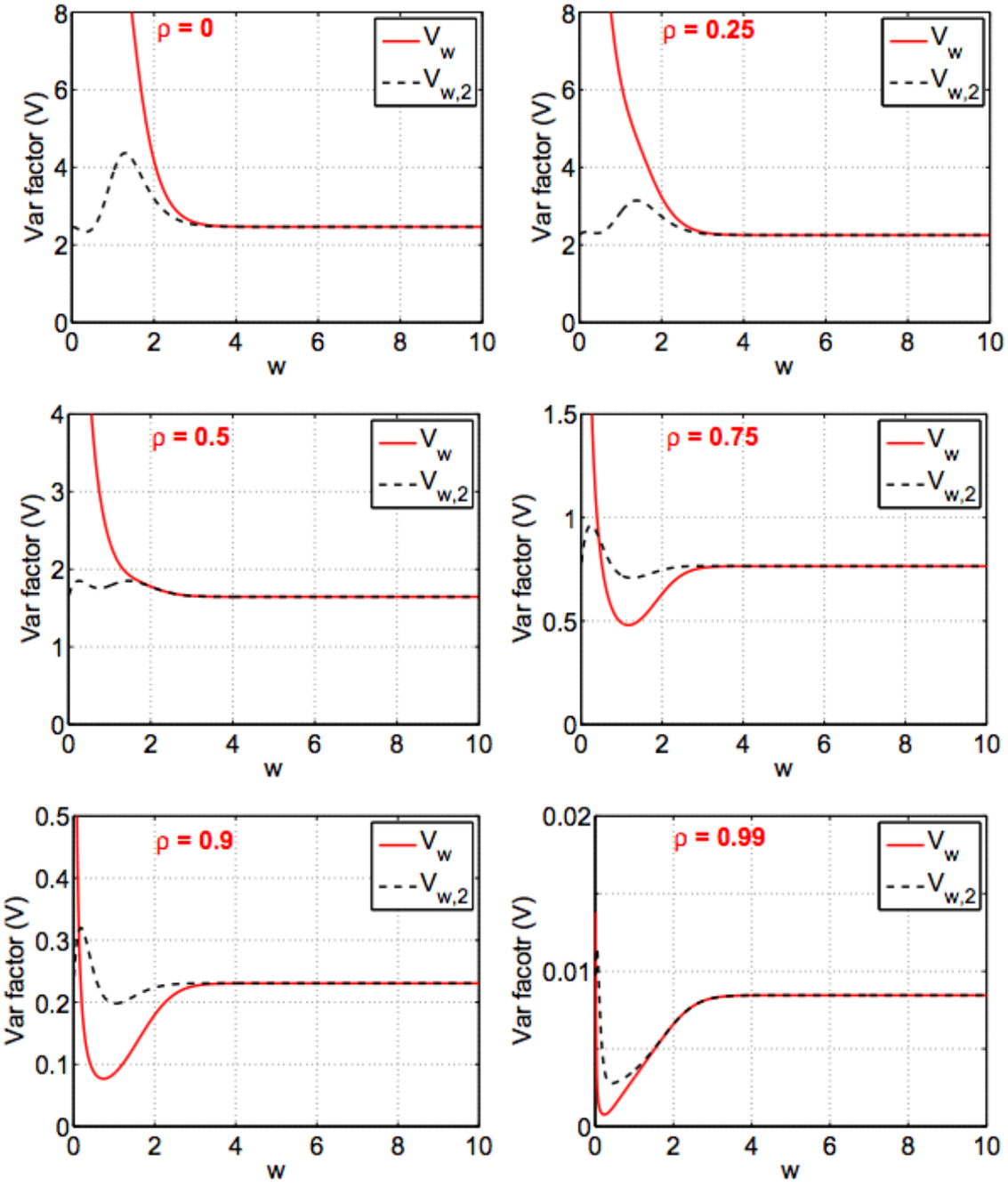
$$= \left\{1 - \frac{1}{\pi}\cos^{-1}\rho\right\} - 4\int_0^w \phi(z)\Phi\left(\frac{-w+\rho z}{\sqrt{1-\rho^2}}\right)dz$$

Collision Comparison: Uniform and 2-Bit



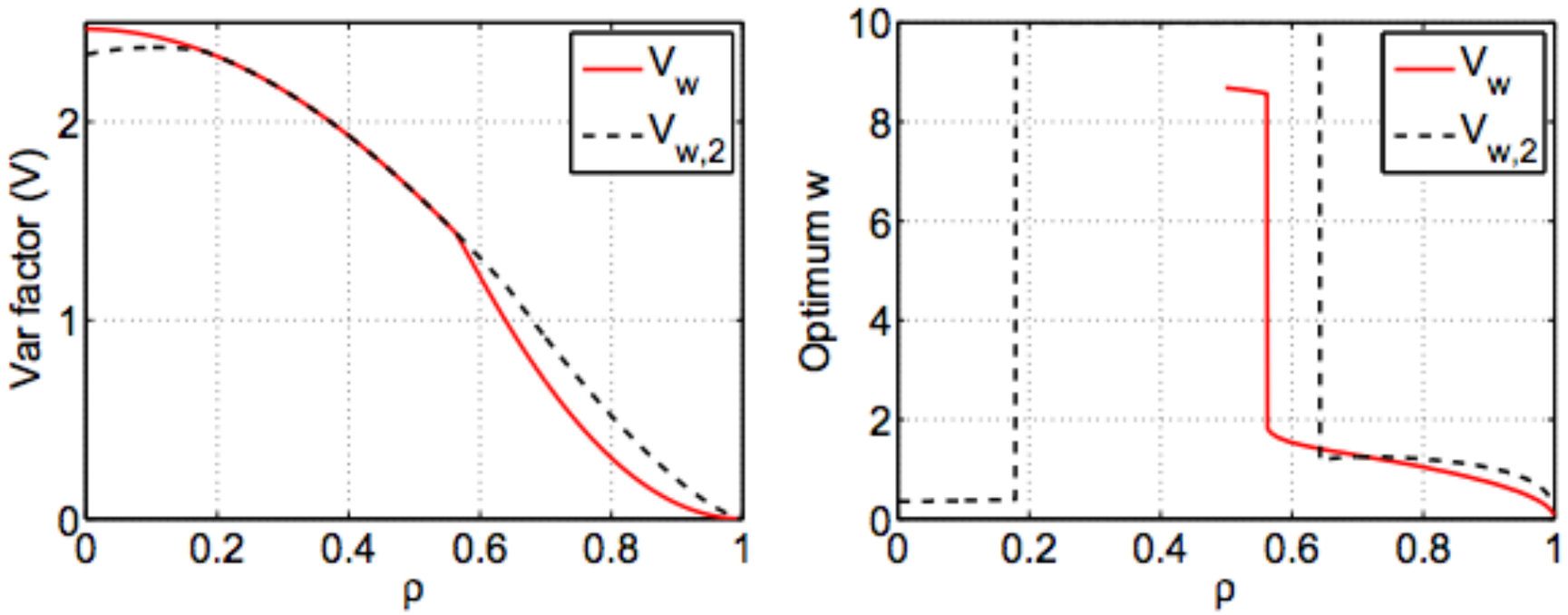
- For small w, both behave very differently (as expected)
- $P_{w,2}$ monotonically increases in similarity, no longer monotone in w

Variance Comparison: Uniform and 2-Bit



- Performance of 2-bit non-uniform coding scheme will be similar to uniform quantization
- For applications that care about highly similar pairs, uniform quantization will have slightly better performance at the cost of more bits

Optimal Comparison: Uniform and 2-Bit



- Performance is similar in most regions
- For similarity between 0.2 and 0.62, it is preferable to use 1 bit instead of 2 bits

Linear SVM Experiments

5 coding schemes

- Original: no coding
- $h_{w,q}$: prior coding scheme [Datar et al 2004]
- h_w: uniform quantization
- $h_{w,2}$: 2-bit coding
- h_1: 1-bit coding (no bin width w is necessary)

Example of Coding

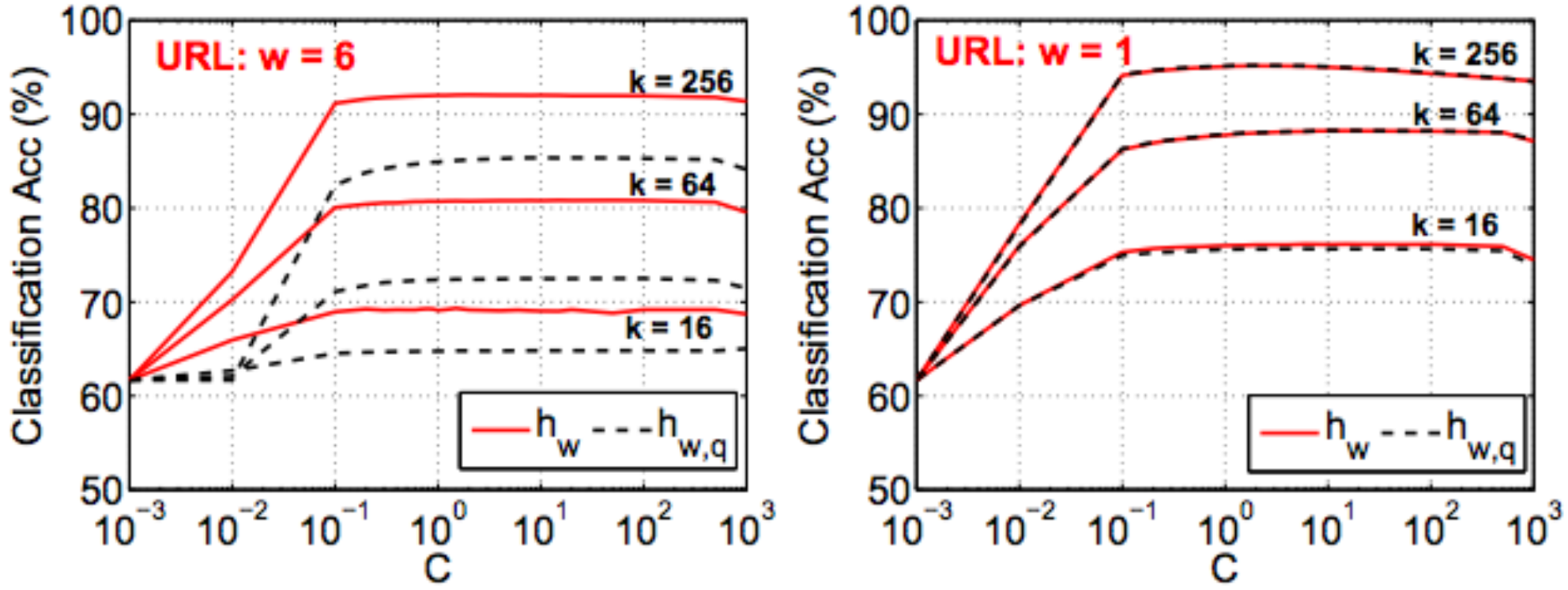
$h_{w,2}$ and w = 0.75

For projected value x:

$$
\begin{aligned}
x \in (-\infty, 0.75) &\implies \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \\
x \in [0.75, 0) &\implies \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} \\
x \in [0, 0.75) &\implies \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \\
x \in [0.75, +\infty) &\implies \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

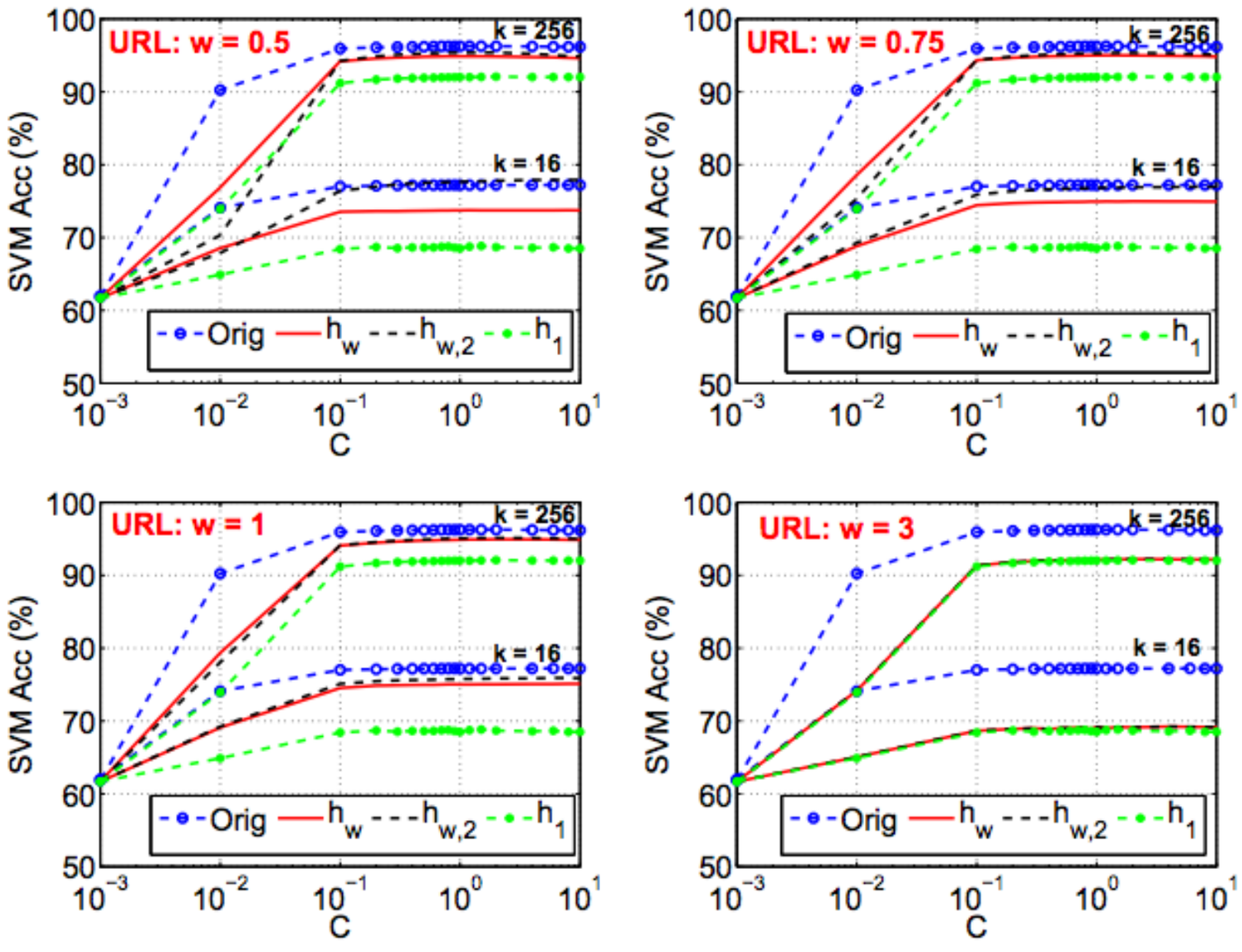
Same trick as in b-bit minwise hashing (NIPS 2011)

Linear SVM: Uniform vs. Old Scheme



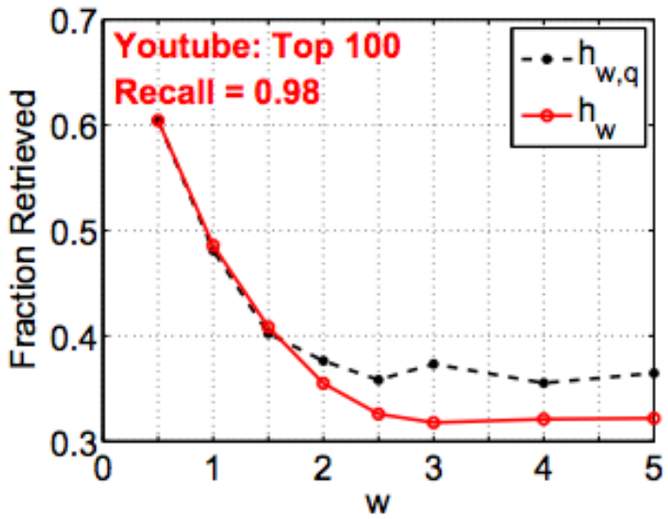
- For small bin width, two schemes are very similar
- Step of random offset from the old scheme is not necessary

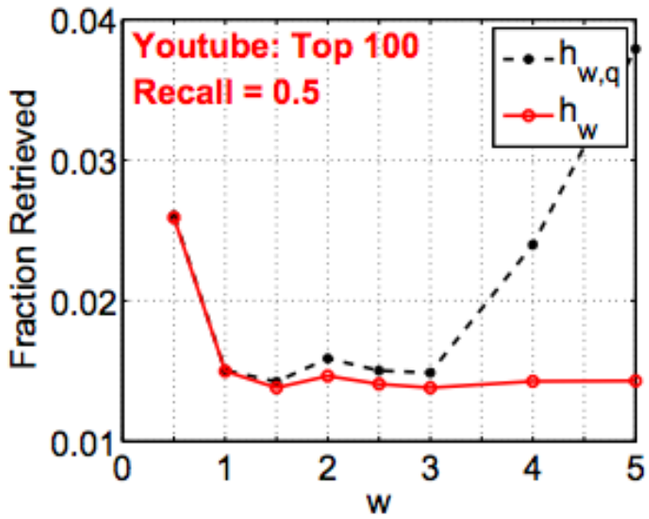
Linear SVM: No Coding vs. Proposed

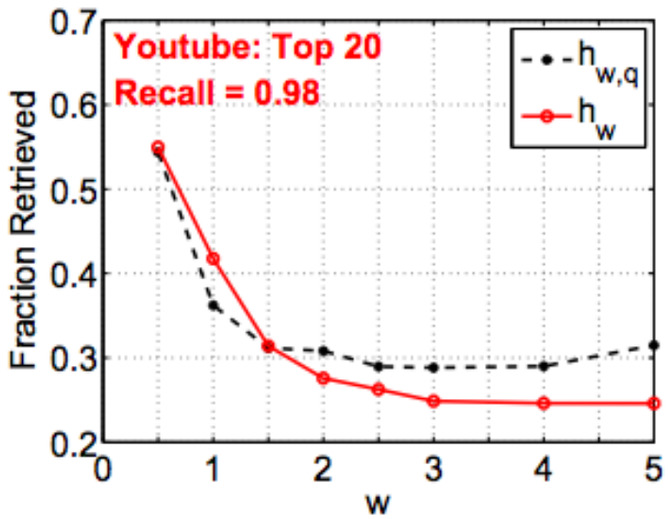


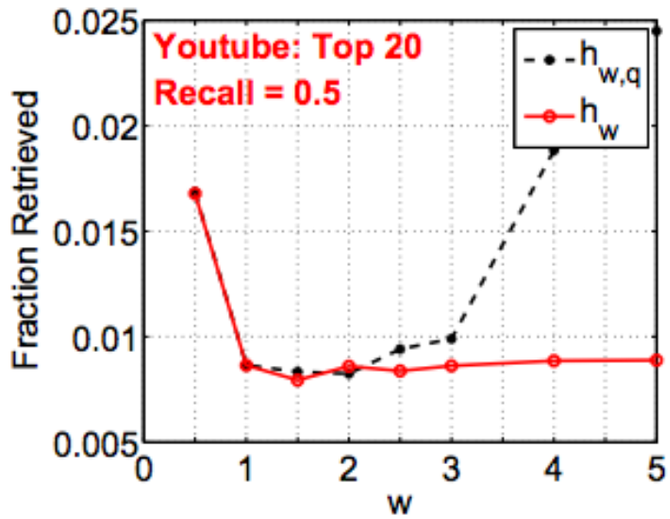
- When w = 0.5 to 1, the uniform and 2-bit are similar as using projected data
- 1 bit scheme is less competitive than most of the others

Coding for LSH Hash Tables









- Use coded values to determine which buckets correspond points are placed in
- Always preferable to use no random offset
- Often only a small number of bits are needed

Summary

- Method of random projections is standard approach for machine learning and data mining

- Compact representation of projected data is crucial for efficient transmission, retrieval, and energy consumption

- Introduced uniform quantization that is operationally simpler, more accurate, less sensitive to parameters, and uses fewer bits

- Introduced 2-bit non-uniform coding scheme which performs similarly to uniform quantization