Exploring the Limits of Classification Accuracy

Carolyn Kim [1] Lester Mackey [2]

[1]Computer Science Department, Stanford University

[2]Statistics Department, Stanford University

December 7, 2015

Classification

- Setup: random variable (X, Y), where X describes the observations, and Y describes the class label
  - In our case, X takes values in $\mathbb{R}^d$ (jet images), and Y takes values in ± 1 ("signal" W-jets or "background" QCD-jets).
- We can construct a classifier: $g : \mathbb{R}^d \to \{\pm 1\}$, with loss $L(g) := \mathbb{P}\{g(X) \neq Y\}$
- We want the optimal classifier (*Bayes Classifier*)

$$g^* = \operatorname*{argmin}_{g:\mathbb{R}^d \to \{\pm 1\}} \mathbb{P}\{g(X) \neq Y\}$$

$$L^* := L(g^*)$$

- g^* is the classifier that outputs 1 if $\mathbb{P}\{1|x\} > \mathbb{P}\{-1|x\}$

k-Nearest Neighbors

The k-nearest neighbor classifier $g_{k,n}$ given n samples $(X_1, Y_1), \ldots, (X_n, Y_n)$ with weights $w_1, \ldots, w_n$ is

$$g_{k,n}(x) = \begin{cases} 1 & \sum_{\substack{X_i \in \{\text{k-nearest neighbors}(x)\} \\ Y_i = 1}} w_i > \sum_{\substack{X_i \in \{\text{k-nearest neighbors}(x)\} \\ Y_i = -1}} w_i \\ -1 & \text{otherwise} \end{cases}$$

Theorem (Universal Consistency of k-Nearest Neighbors, Deyvroye and Gyorfi, 1985, Zhao (1987))

For any distribution of (X, Y), as $k \to \infty$, $k/n \to 0$, $n \to \infty$, i.i.d. samples, then $L(g_{k,n}) \to L^*$.

Theorem (Devroye, 1981)

For $k \geq 3$ and odd, $\lim_{n\to\infty} L(g_{1,n}) \leq L^*(1 + \sqrt{\frac{2}{k}})$.

Experimental setup

- Generate data: simulated signal and background events with $p_t \in [200, 400]$ GeV; each event is defined by a weight and 20-40 particles defined by (ϕ, η, energy).
- Bin data, resulting in a jet image, a vector in $\mathbb{R}^d$.
- Optionally, whiten the data so the training covariance matrix is the identity.
- Compute the distances to the k-th nearest signal and background neighbors (this is enough information to do $2k-1$-nearest neighbor) in the "distance training set" (900K or 10M in size). In practice, this requires a lot of computational power!
- create a rejection versus efficiency curve

Step 1: Binning

Multiple possible binning strategies: equal size binning or equal weight (event weight only vs. event weight * energy bin bounds, energy only bin values vs energy density bin values)

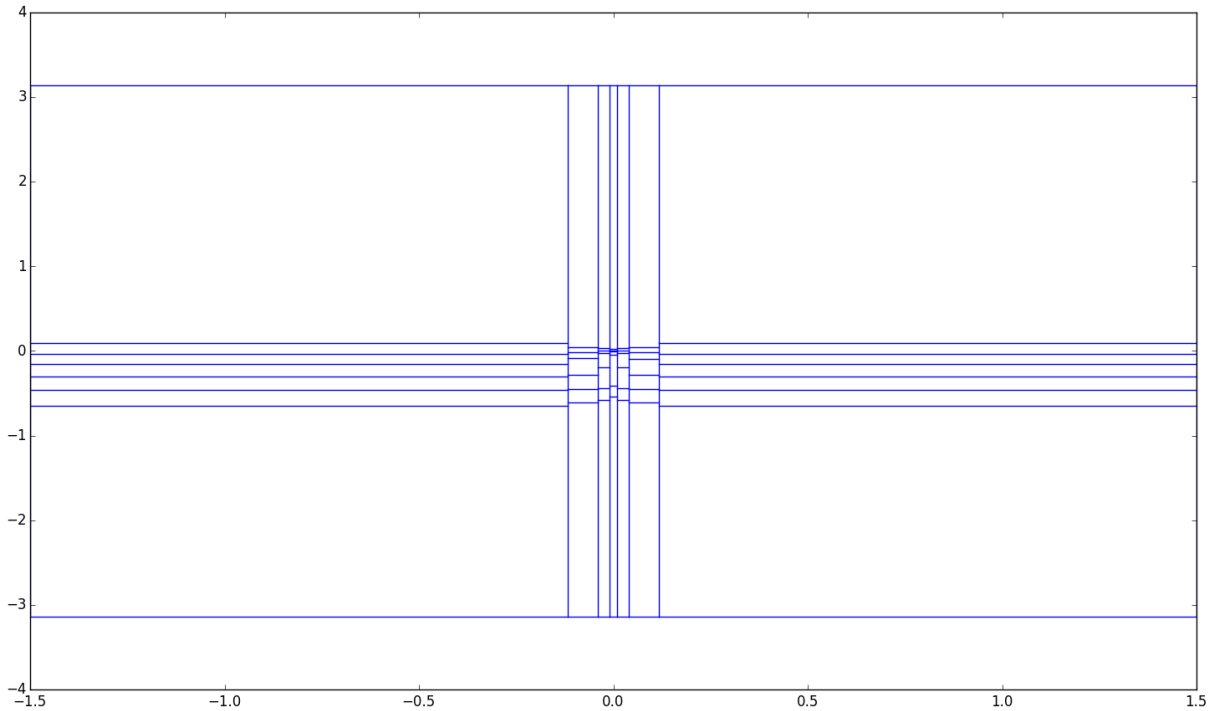
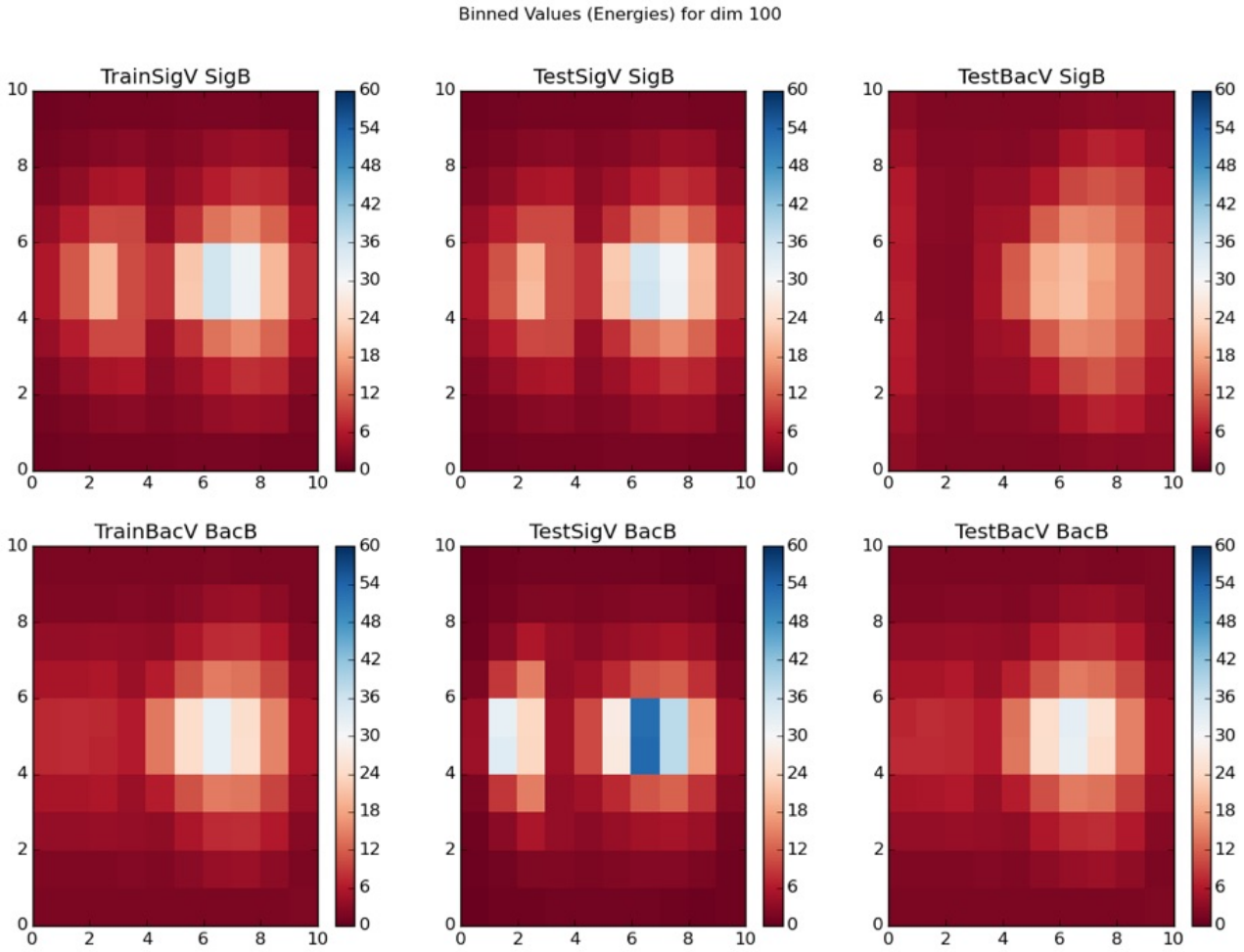


Figure 1: Sample bin bounds for an equal weighting scheme

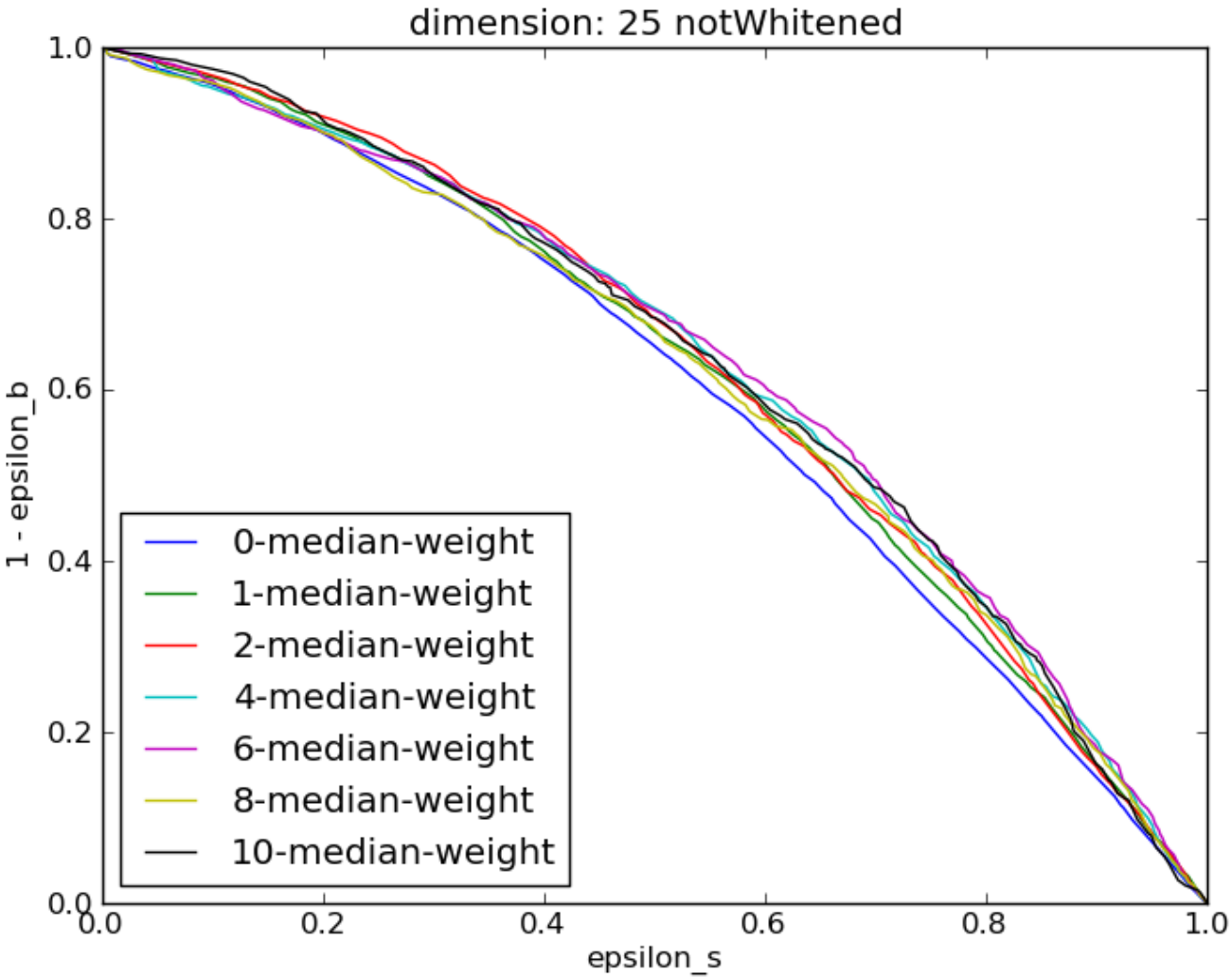
Mean heatmap of one binning strategy



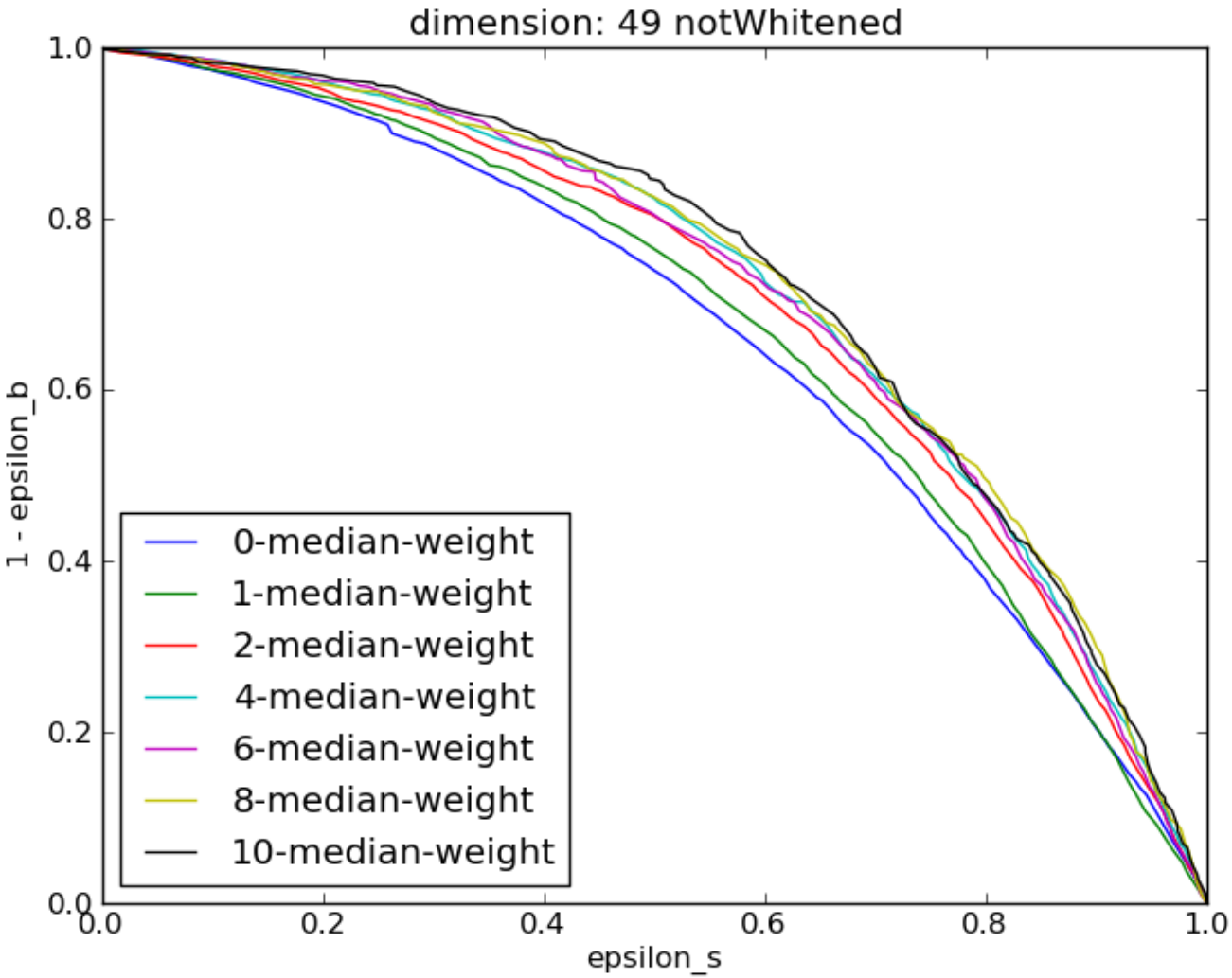
Plotting rejection versus efficiency curve

- x-axis is signal efficiency (proportion of signal classified as signal)
- y-axis is $1-$background efficiency
- The 1-D discriminant is the ratio between the probability densities of distances to the k-th nearest signal and background neighbor. (2D-likelihood without taking the ratio has empirically not been better.)
- use one set of distances as a "curve training" set to estimate the densities, and another set of distances as the "curve testing" set to plot the curve

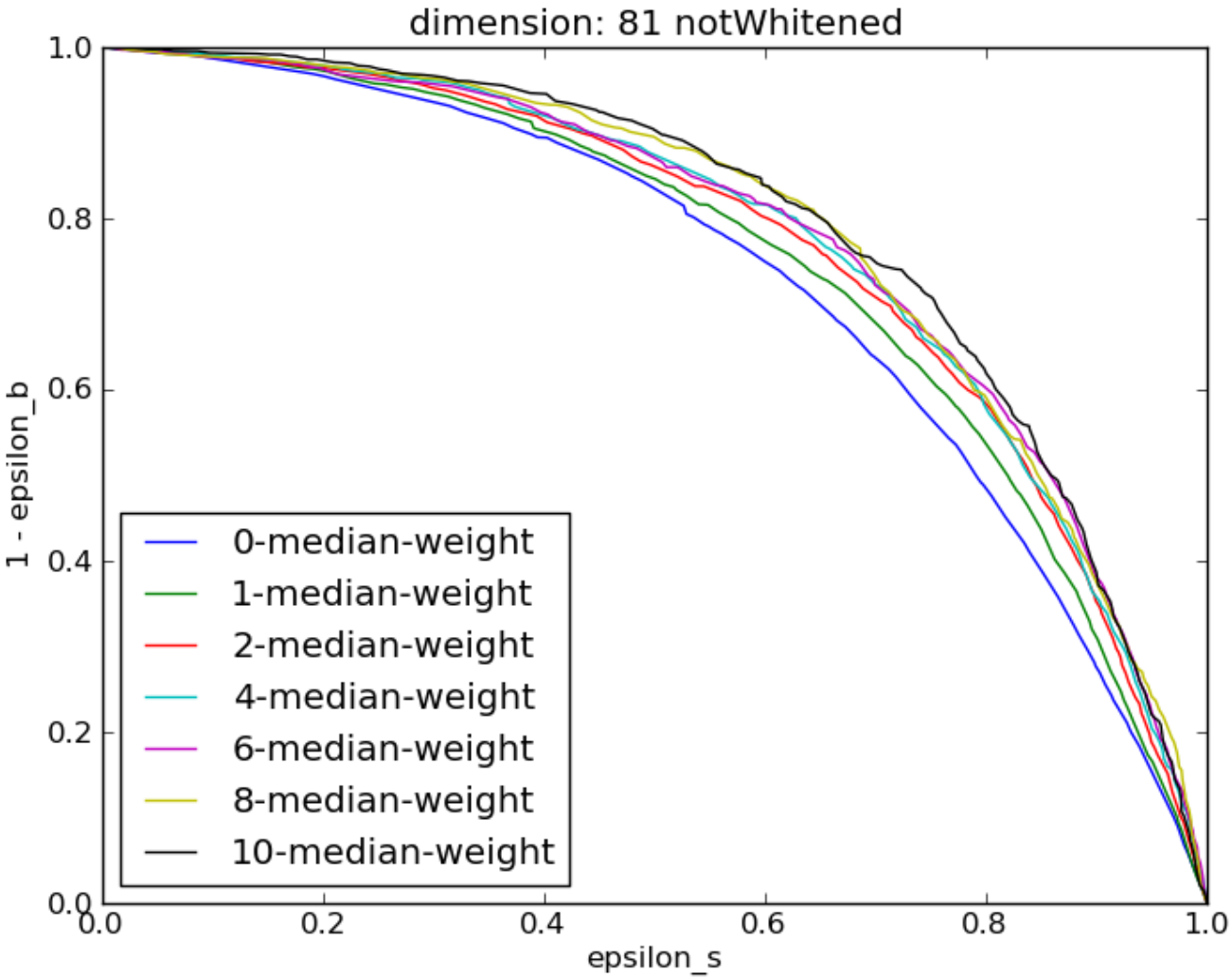
curve training, testing: 100K; distance training: 900K



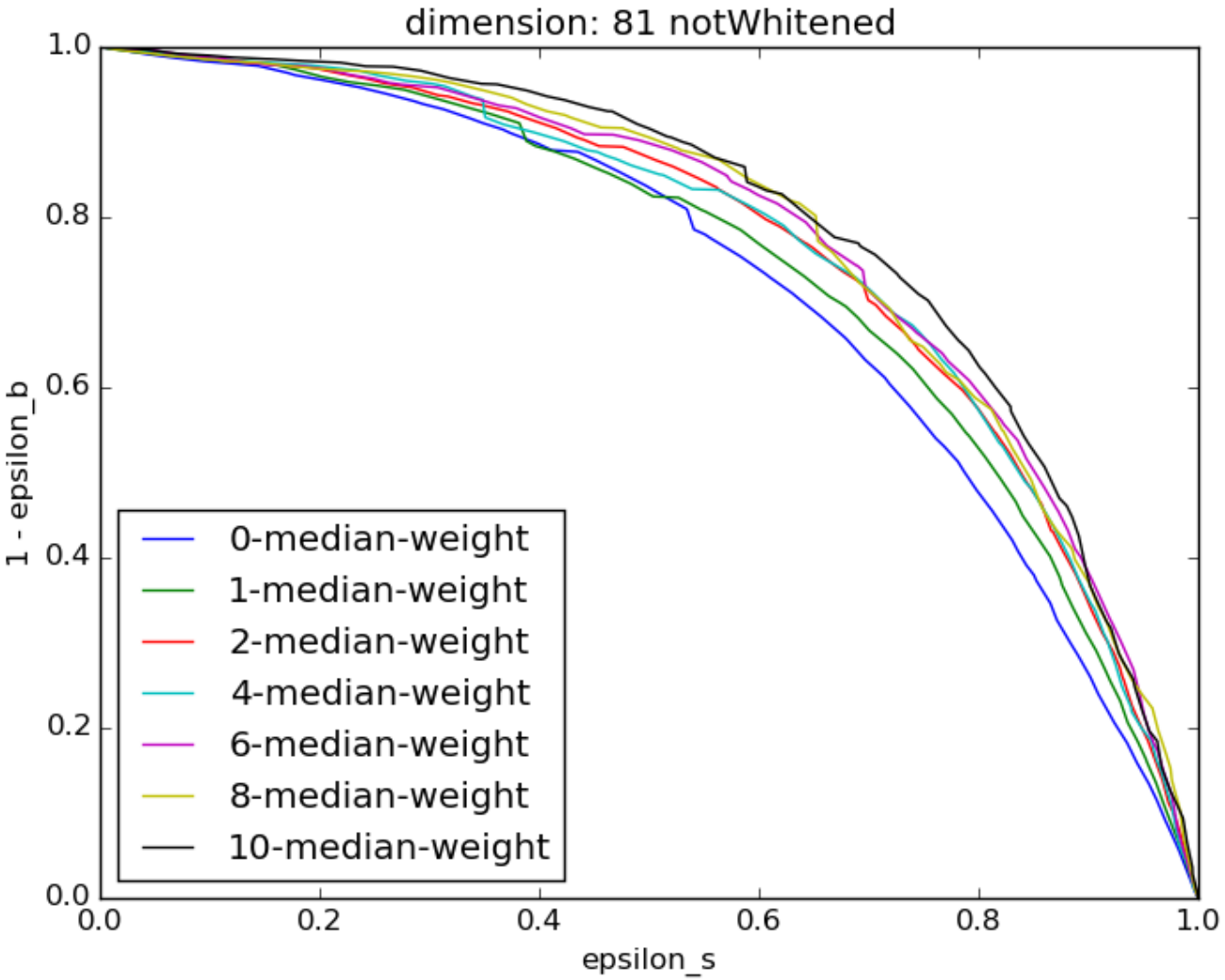
curve training, testing: 100K; distance training: 900K



curve training, testing: 100K; distance training: 900K

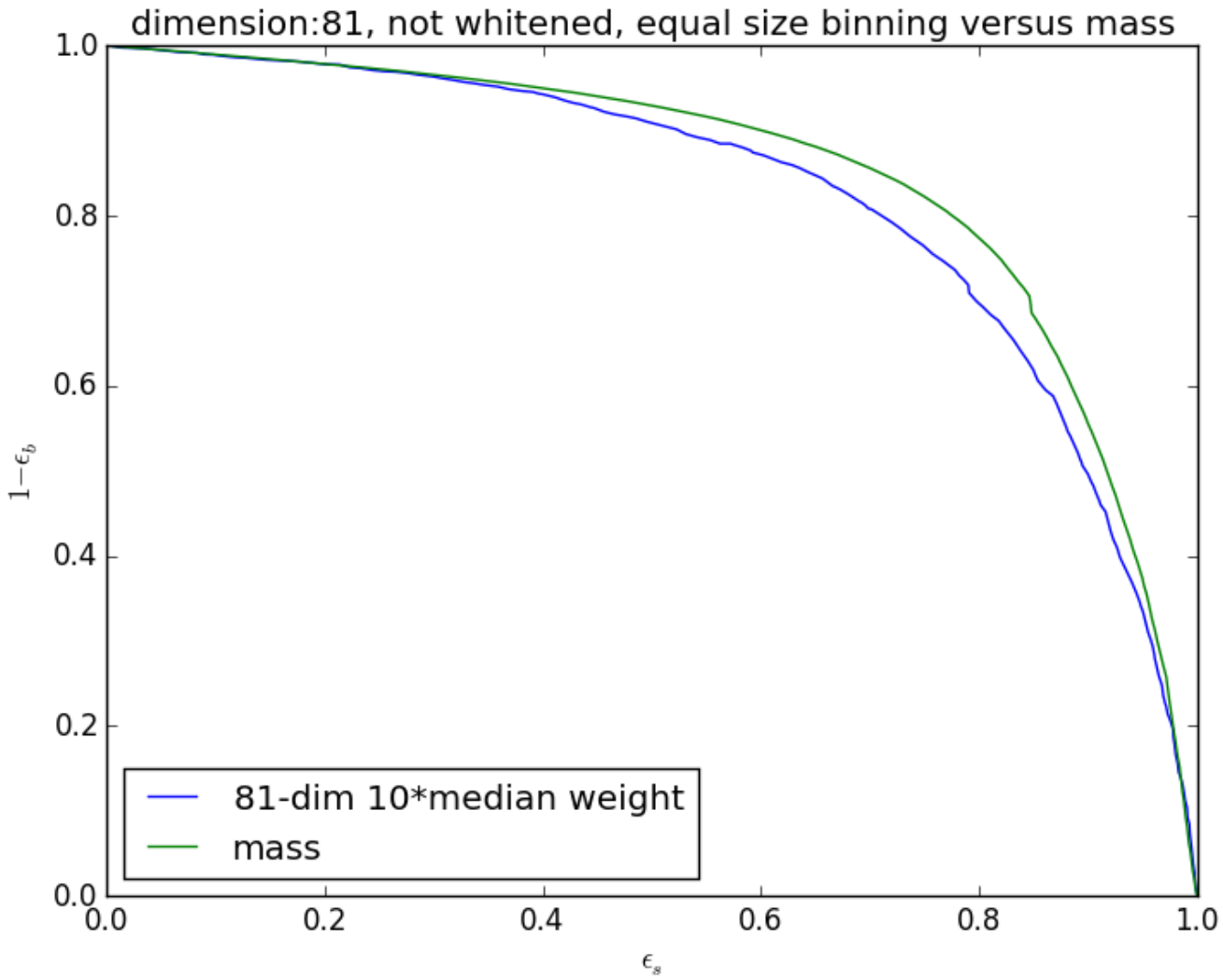


curve training, testing: 1M; distance training: 10M



curve training, testing: 1M; distance training: 10M

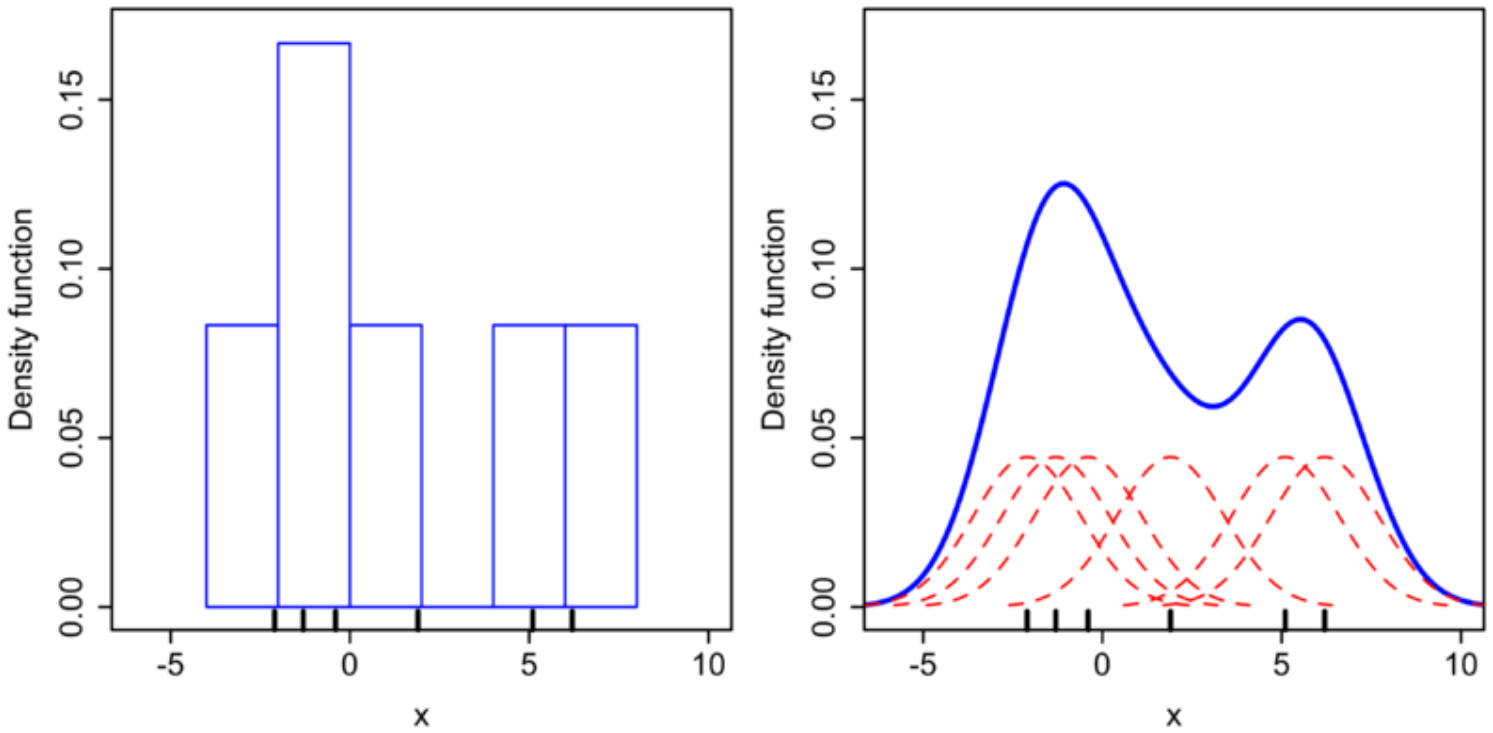
How well are we doing? Unfortunately, worse than mass...



Kernels

A kernel function $K : \mathbb{R}^d \to \mathbb{R}$ intuitively creates "bumps" around 0 (e.g. Gaussian kernel $K(x) = e^{-\|x\|^2}$).

We can estimate the probability density function by summing up kernel functions centered at the data points: $P(y_j|x) \propto \sum_{Y_i=y_j} w_i K(x - X_i)$



Credit: *http://en.wikipedia.org*

The kernel classifier $g_{K,n}$ for a kernel function K given n samples $(X_1, Y_1), \ldots, (X_n, Y_n)$ with weights $w_1, \ldots, w_n$ is

$$g_{K,n}(x) = \begin{cases} 1 & \sum_{Y_i=1} w_i K(\frac{x-X_i}{h}) > \sum_{Y_i=-1} w_i K(\frac{x-X_i}{h}) \\ -1 & \text{otherwise} \end{cases}$$

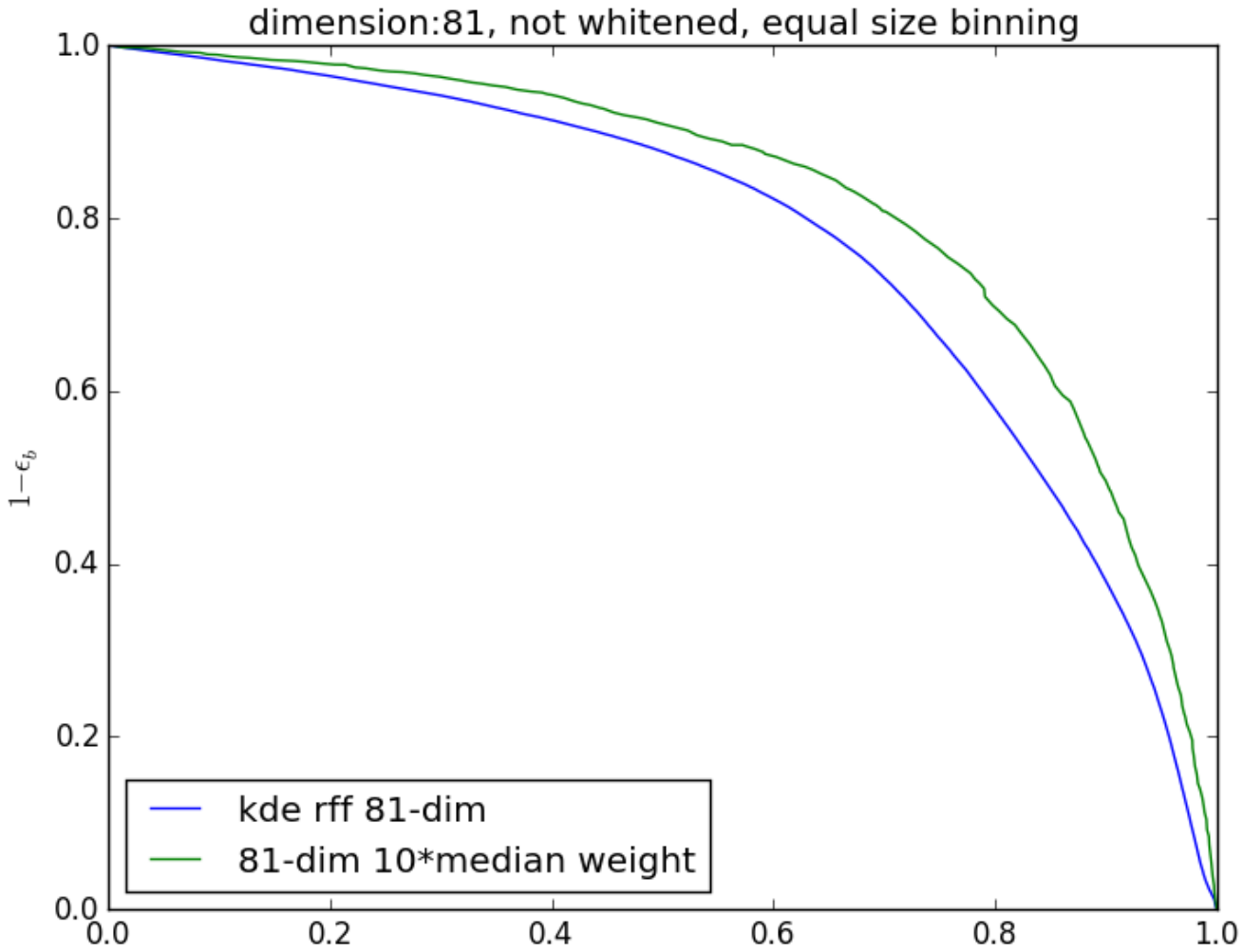
Theorem (Devroye and Kryzyzak, 1989)

For any distribution of (X, Y), if $h \to 0$ and $nh^d \to \infty$ as $n \to \infty$, i.i.d. samples, then $L(g_{\text{Gaussian},n}) \to L^*$.

This classifier can converge faster than the k-NN estimator if the conditional densities are smooth.

Random Fourier Feature Kernel Density Estimation

Randomized algorithm to approximate the Gaussian kernel, which makes it more efficient (at least a 10x speedup.)



Next Steps

- Use FLANN, a library for fast approximate nearest neighbors.
- Scale to higher dimensions: currently it takes 10 hours to run 81-dimensional data; use more data!
- Tune random Fourier Feature parameters
- Other strategies: e.g. use independent component analysis