CS340 Machine learning
Lecture 5
Notes

Outline

- HW1
- Finish KNN
- Start info theory

Office hours Tue 4-5, CS187

Standard error

- Suppose we want to estimate E[X] from n samples, $X_1, \ldots, X_n$ (eg X is generalization error)
- Suppose X ~ p(), where E[X]=μ, Var[X]=σ^2
- Construct an *estimator* using the empirical mean

$$\hat{\mu}(D) = \frac{1}{n}\sum_{i=1}^{n} X_i$$

- What is the mean and variance of this estimator?

$$E[\hat{\mu}(D)] = \frac{1}{n}\sum_i E[X_i] = \frac{n}{n}\mu = \mu$$

$$Var[\hat{\mu}(D)] = \frac{1}{n^2}\sum_i Var[X_i] = \frac{n}{n^2}\sigma^2 = \frac{\sigma^2}{n}$$

- Standard error is *estimated* standard deviation

$$se = \sqrt{\frac{\hat{\sigma}^2}{n}} = \frac{\hat{\sigma}}{\sqrt{n}}$$

Application to M-CV

- For k=1:K (K=20), and m=1:M (M=5 fold), we compute the empirical generalization error on X_m, training on X_{-m}, using a kNN with value k

$$err(k,m) = \frac{1}{|X_m|} \sum_{i \in X_m} I(\hat{y}(x_i|k, X_{-m}) \neq y_i)$$

- The average error for k is

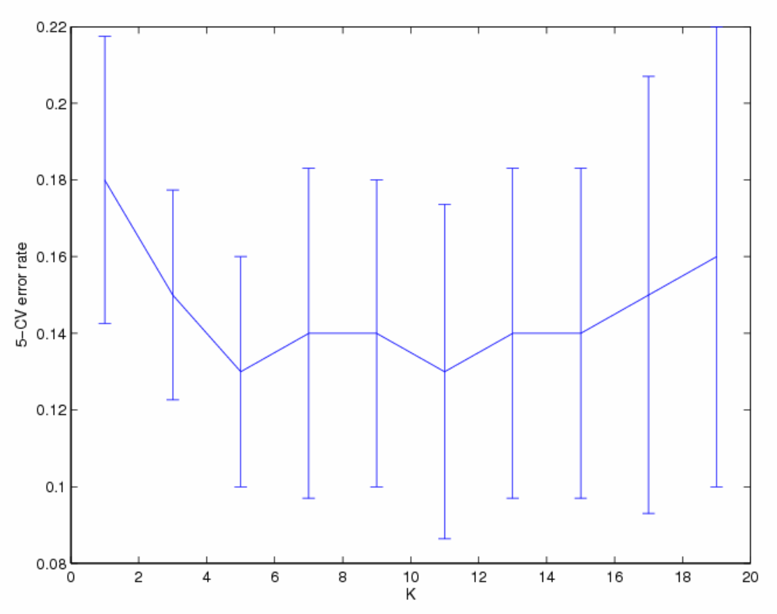
$$\hat{\mu_k} = mean(err(k,:)) = \frac{1}{M} \sum_{m=1}^{M} err(k,m)$$

- The standard error is

$$se_k = \frac{std(err(k,:))}{\sqrt{M}}$$

CV for kNN

CV error

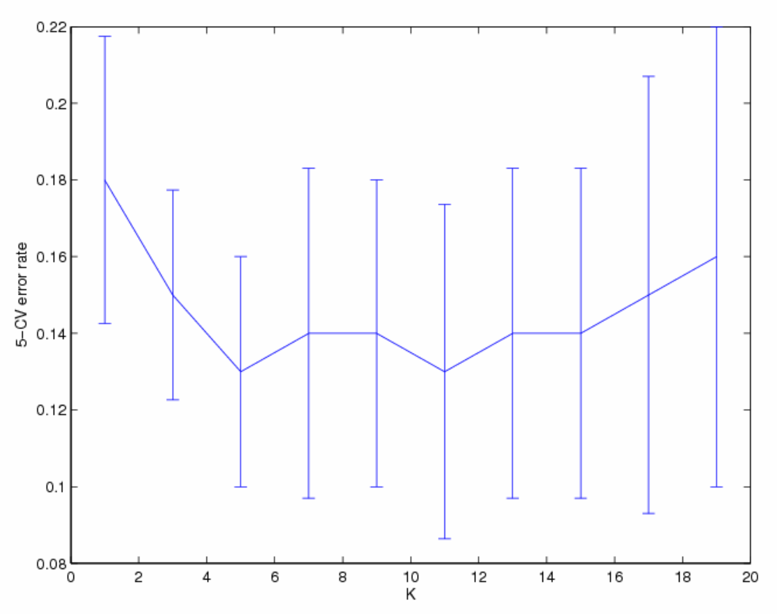


K

Picking K

- Can use the “one standard error” rule*, where we pick the simplest model whose error is no more than 1 se above the best.
- For KNN, dof=N/K, so we would pick K=11.

CV error



K

* HTF p216

Outline

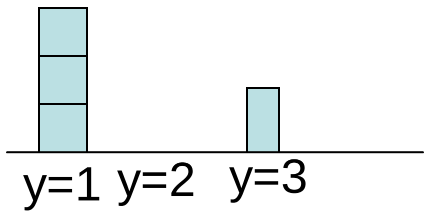
- HW1
- Finish KNN
- Start info theory

Probabilistic kNN

- We can compute the empirical distribution over labels in the K-neighborhood

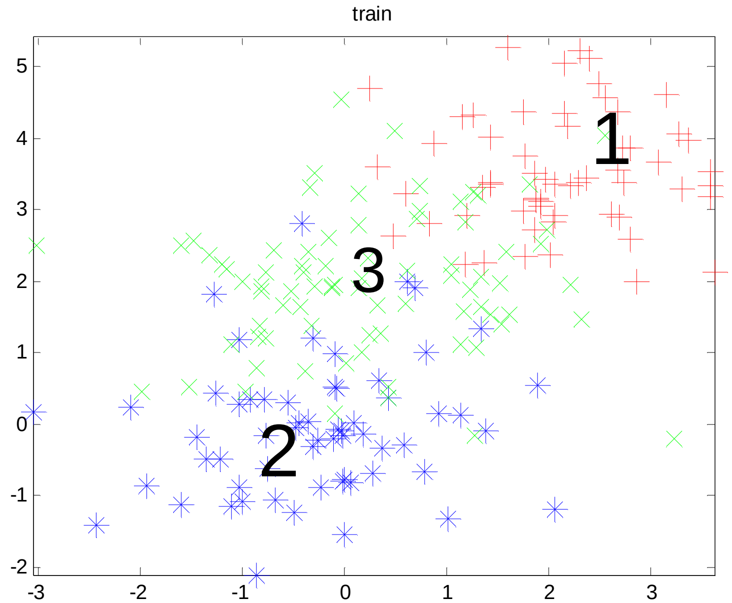
$$p(y|x, D) = \frac{1}{K} \sum_{j \in nbr(x,K,D)} I(y = y_j)$$

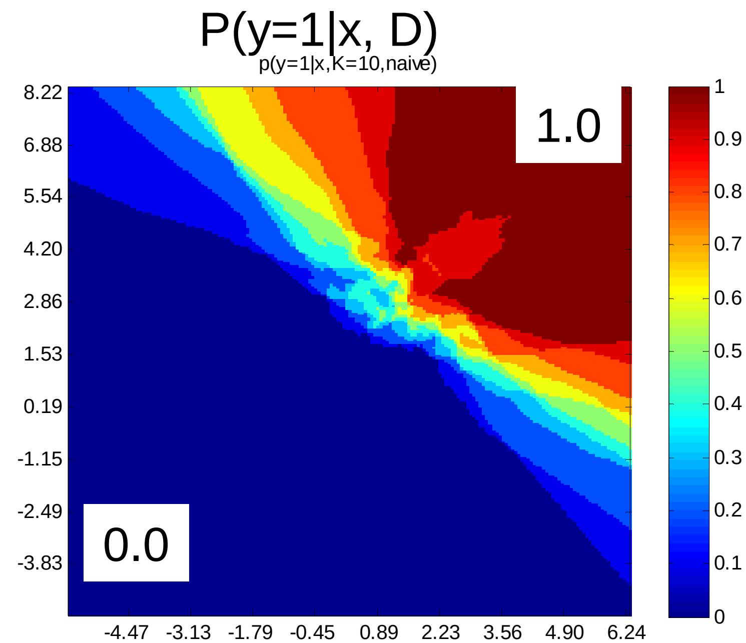
K=4, C=3

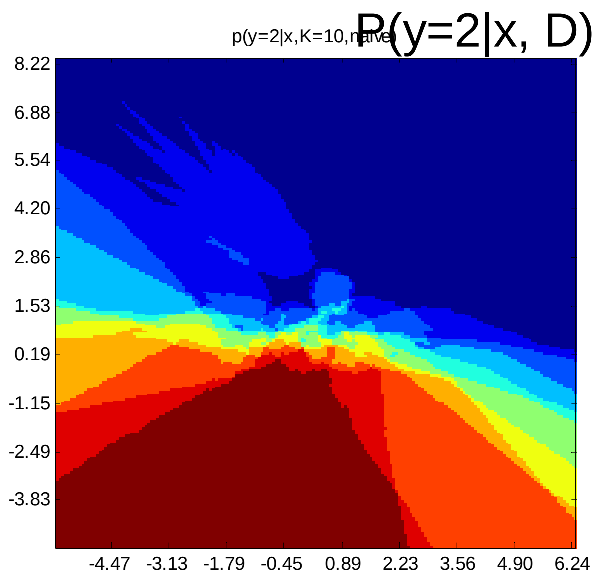


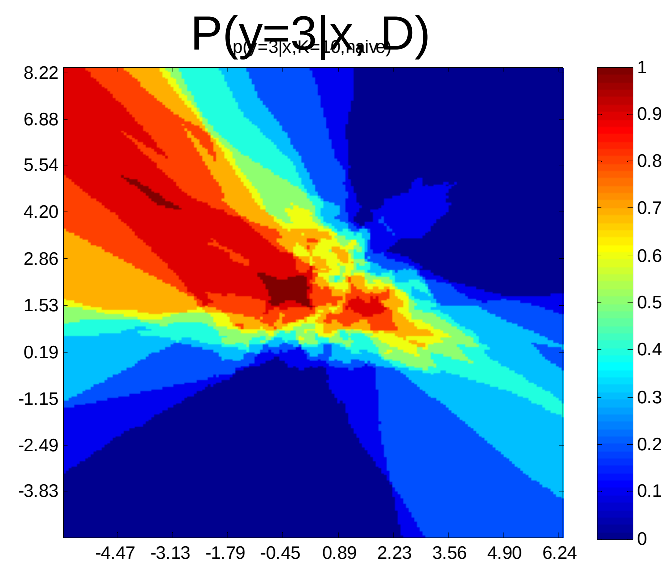
P = [3/4, 0, 1/4]

Probabilistic kNN





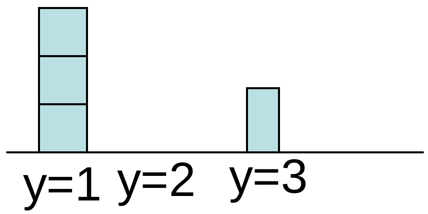




Smoothing empirical frequencies

- The empirical distribution will often predict 0 probability due to sparse data
- We can add *pseudo counts* to the data and then normalize

K=4, C=3



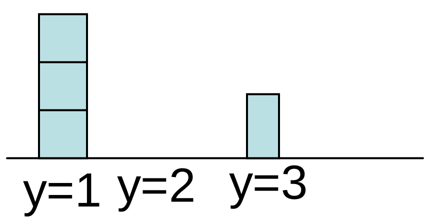
P = [3 + 1, 0 + 1, 1 + 1] / 7 = [4/7, 1/7, 2/7]

Softmax (multinomial logit) function

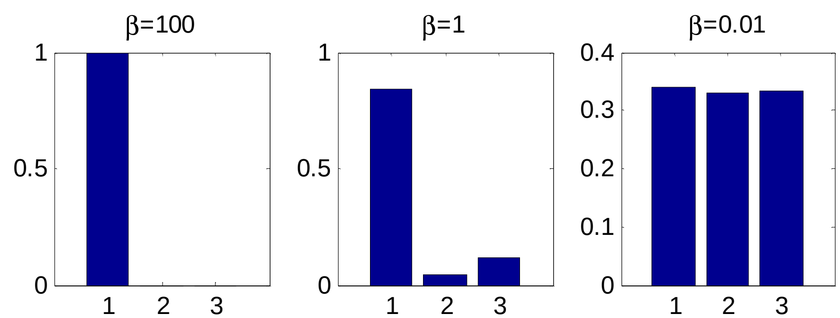
- We can “soften” the empirical distribution so it spreads its probability mass over unseen classes
- Define the softmax with inverse temperature β

$$S(x, \beta)_i = \frac{\exp(\beta x_i)}{\sum_j \exp(\beta x_j)}$$

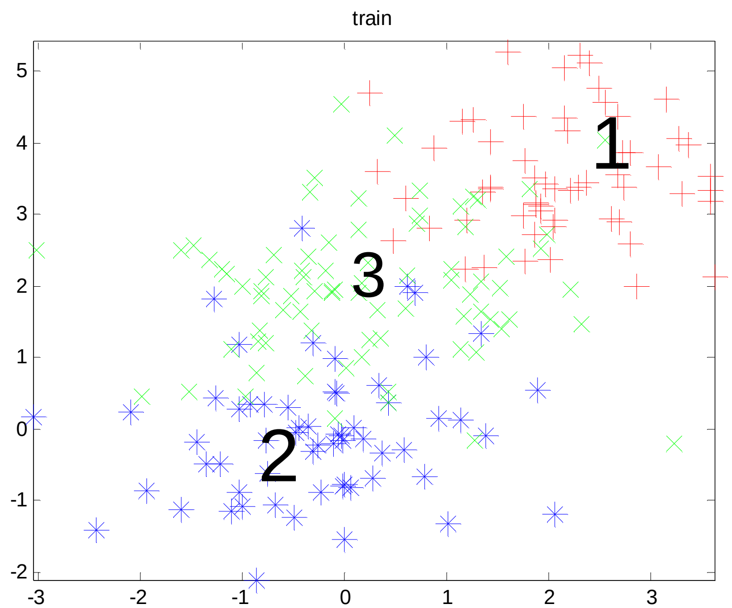
- Big beta = cool temp = spiky distribution
- Small beta = high temp = uniform distribution

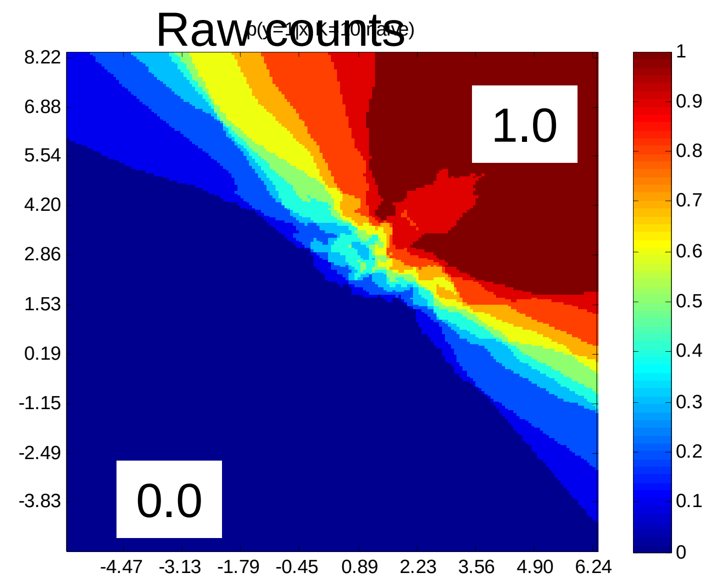


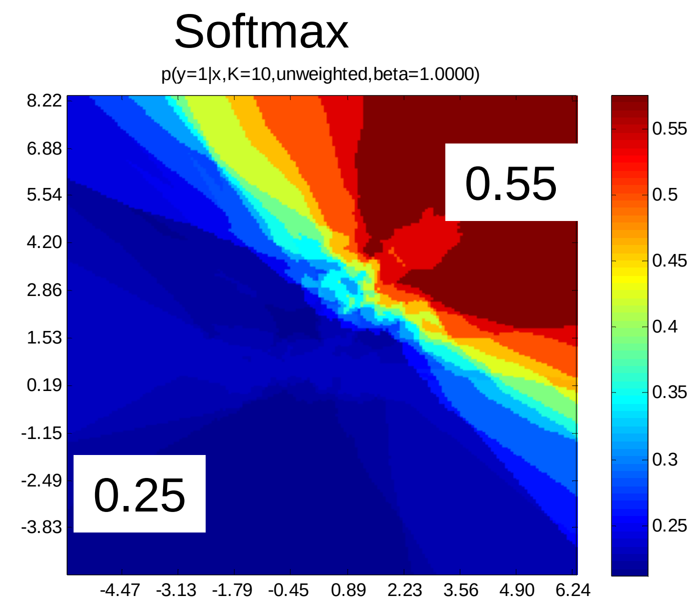
X = [3 0 1]



Softened Probabilistic kNN



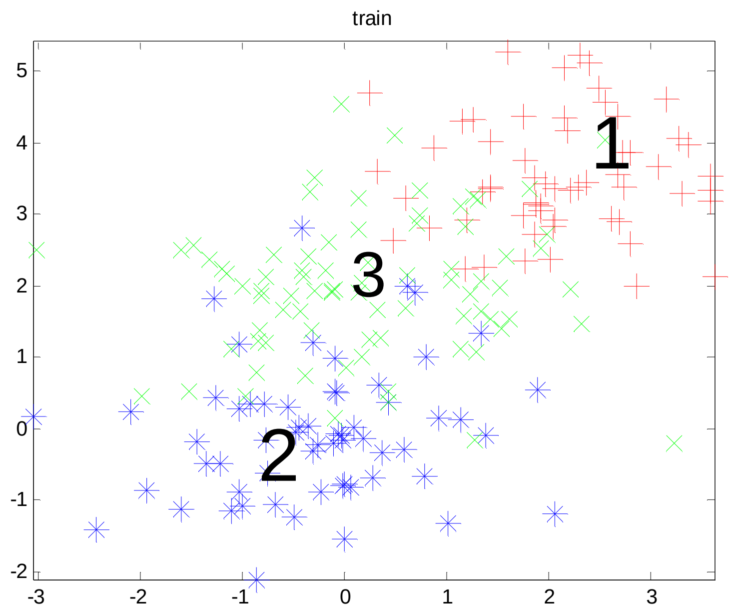




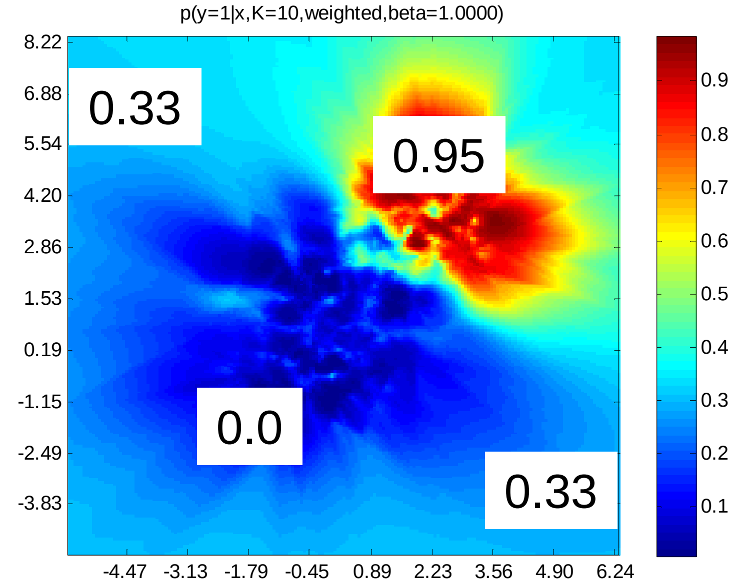
Sum over Knn

$$p(y|x, D, K, \beta) = \frac{\exp[(\beta/K)\sum_{j\sim x} I(y = y_j)]}{\sum_{y'} \exp[(\beta/K)\sum_{j\sim x} I(y' = y_j)]}$$

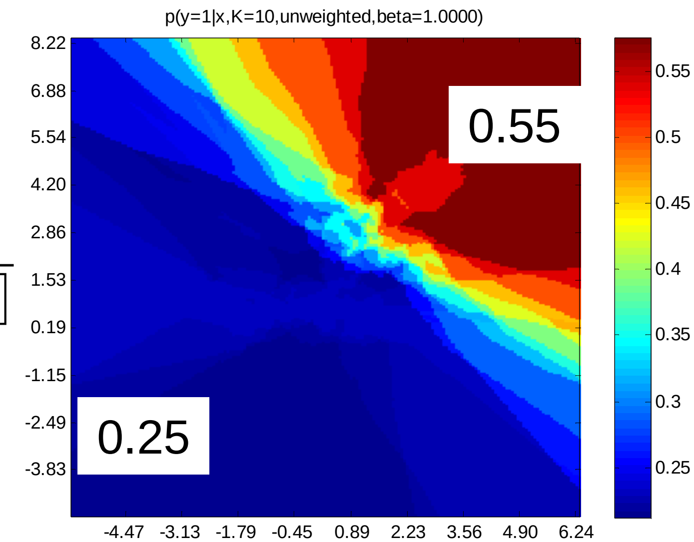
Weighted Probabilistic kNN



Weighted



Softmax



Weighted sum over Knn

$$p(y|x, D, K, \beta) = \frac{\exp[(\beta/K) \sum_{j \sim x} w(x, x_j) I(y = y_j)]}{\sum_{y'} \exp[(\beta/K) \sum_{j \sim x} w(x, x_j) I(y' = y_j)]}$$

Local kernel function

Kernel functions

Any smooth function K such that

$$K(x) \geq 0, \int K(x)dx = 1, \int xK(x)dx = 0 \text{ and } \int x^2 K(x)dx > 0$$

- Epanechnikov quadratic kernel

$$K_\lambda(x_0, x) = D\left(\frac{|x - x_0|}{\lambda}\right) \quad D(t) = \begin{cases} \frac{3}{4}(1-t^2) & \text{if } |t| \leq 1; \\ 0 & \text{otherwise.} \end{cases}$$

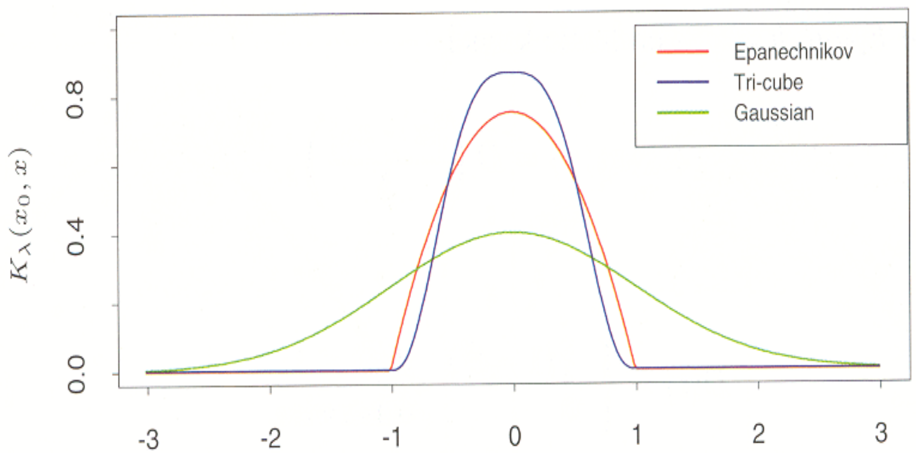
λ = bandwidth

- tri-cube kernel

$$K_\lambda(x_0, x) = D\left(\frac{|x - x_0|}{\lambda}\right) \quad D(t) = \begin{cases} (1-|t|^3)^3 & \text{if } |t| \leq 1; \\ 0 & \text{otherw} \end{cases}$$

- Gaussian kernel

$$K_\lambda(x_0, x) = \frac{1}{\sqrt{2\pi}\lambda}\exp\left(-\frac{(x - x_0)^2}{2\lambda^2}\right)$$



Kernel characteristics

Compact support – vanishes beyond a finite range (Epanechnikov, tri-cube)
Everywhere differentiable (Gaussian, tri-cube)

HTF 6.2

Kernel functions on structured objects

- Rather than defining a feature vector x, and computing Euclidean distance D(x, x'), sometimes we can directly compute distance between two *structured objects*
- Eg string/graph matching using dynamic programming
- Kernels let us apply geometric methods to data that does not consist of just real-valued features

Not on exam