

# Nonparametric Methods for Density Estimation

## Nearest Neighbour Classification

Lecturer:  
Jiří Matas

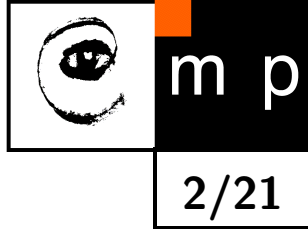
Authors:  
Ondřej Drbohlav, Jiří Matas

Centre for Machine Perception  
Czech Technical University, Prague  
<http://cmp.felk.cvut.cz>

Lecture date: 23.10.2015 & 26.10.2015



# Probability Estimation



Recall that in the previous lecture, **parametric** methods for density estimation have been dealt with. The advantage of these methods is that there is a low number of parameters to estimate. The disadvantage is that the resulting estimated density can be arbitrarily wrong if the underlying distribution does not agree with the assumed parametric model.

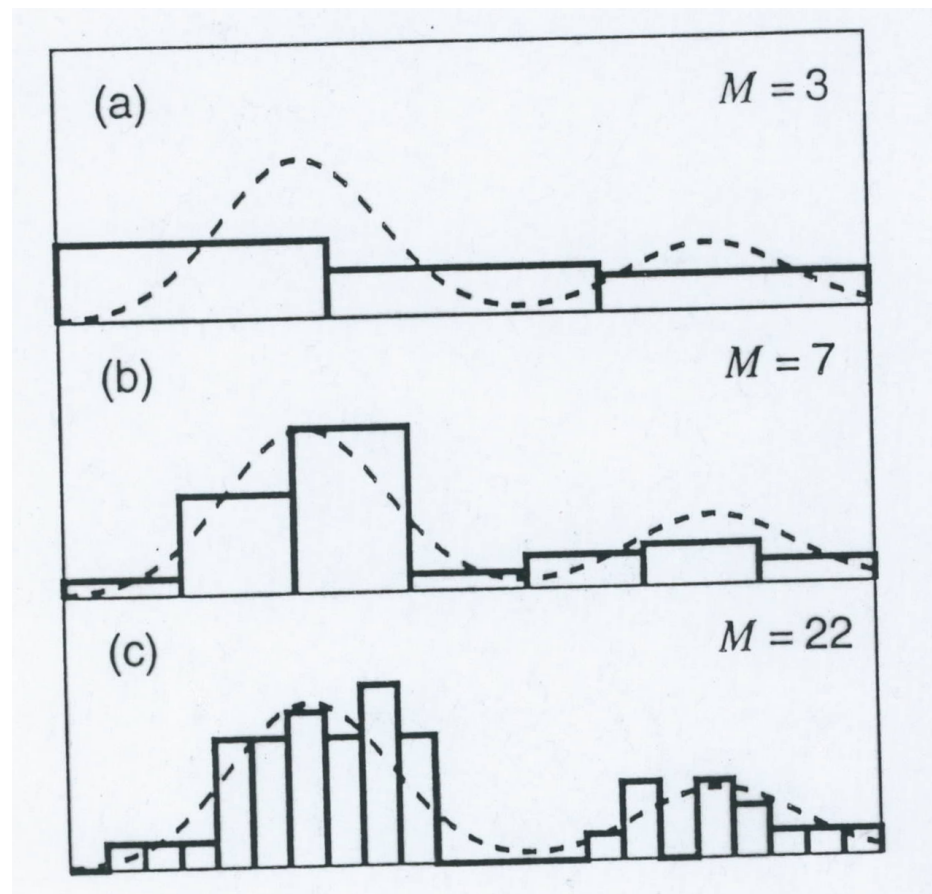
# Non-Parametric Density Estimation



- ◆ histogram
- ◆ Parzen estimation
- ◆ Nearest Neighbor approach

# Histogram

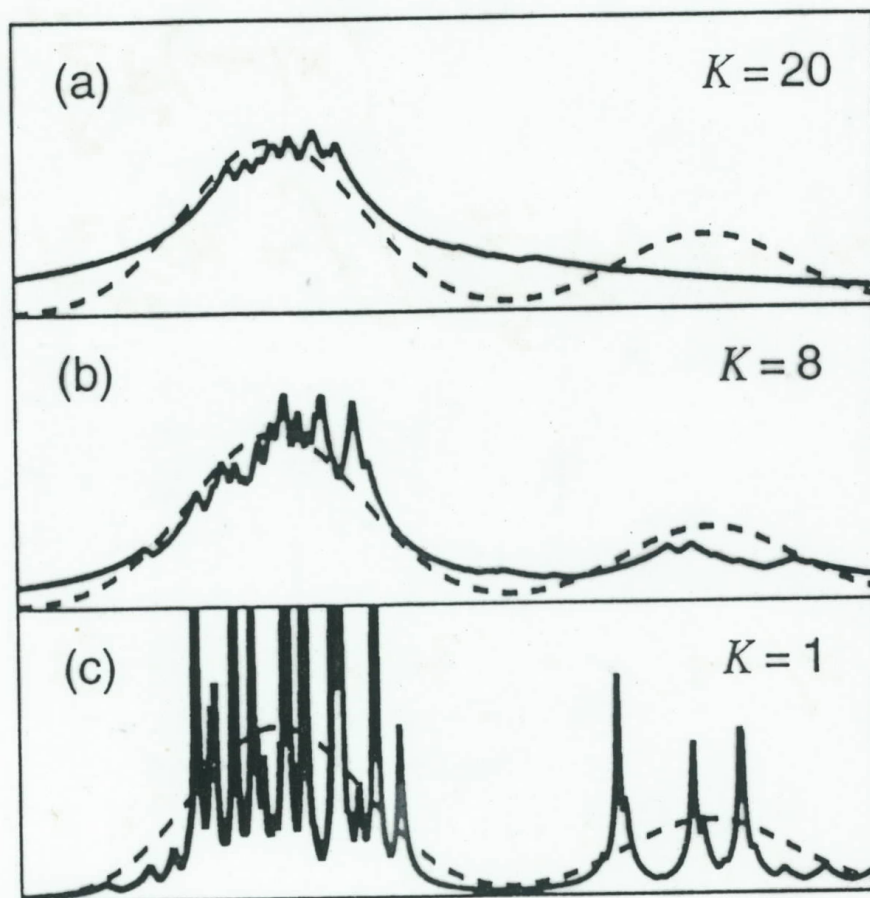
Example,  $N$  : number of bins



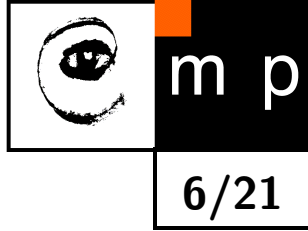
# $K$ -Nearest Neighbor Approach to Density Estimation

Find  $K$  neighbors, density estimate is  $p \sim 1/V$  where  $V$  is the volume of minimum cell in which  $K$  neighbors are located.

Example:



# $K$ -Nearest Neighbor Approach to Classification



Outline:

- ◆ Definition
- ◆ Properties
- ◆ Asymptotic error of  $NN$  classifier
- ◆ Error reduction by edit operation on the training class
- ◆ Fast NN search

# *K*-NN Definition

## Assumption:

- ◆ Training set  $\mathcal{T} = \{(x_1, k_1), (x_2, k_2), \dots, (x_N, k_N)\}$ . There are  $R$  classes (letter  $K$  is reserved for  $K$ NN in this lecture)
- ◆ A distance function  $d : X \times X \mapsto \mathbb{R}_0^+$

## Algorithm:

1. Given  $x$ , find  $K$  points  $S = \{(x'_1, k'_1), (x'_2, k'_2), \dots, (x'_K, k'_K)\}$  from the training set  $\mathcal{T}$  which are closest to  $x$  in the metric  $d$ :

$$S = \{(x'_1, k'_1), (x'_2, k'_2), \dots, (x'_K, k'_K)\} \equiv \{(x_{r_1}, k_{r_1}), (x_{r_2}, k_{r_2}), \dots, (x_{r_K}, k_{r_K})\} \quad (1)$$

$$r_i: \text{the rank of } (x_i, k_i) \in \mathcal{T} \text{ as given by the ordering } d(x, x_i) \quad (2)$$

2. Classify  $x$  to the class  $k$  which has majority in  $S$ :

$$k = \operatorname{argmax}_{l \in R} \sum_{i=1}^K \mathbb{I}[k'_i = l] \quad (x'_i, k'_i) \in S \quad (3)$$

## *K*-NN Properties

- ◆ Trivial implementation ( $\rightarrow$  good baseline method)
- ◆ 1-NN: error of classification  $\epsilon_{NN}$  is usually strictly higher than the Bayesian one  $\epsilon_B$  even when  $N \rightarrow \infty$ . But, higher bounds exist, e.g.  $\epsilon_{NN} \leq 2\epsilon_B$
- ◆ Slow when implemented naively, but can be sped up (Voronoi, k-D trees)
- ◆ High computer memory requirements (but training set can be edited and its cardinality decreased)
- ◆ How to construct the metric  $d$ ? (problem of scales in different axes)
- ◆ No generalization (Vapnik-Chervonenkis dimension =  $\infty$ , error on training set = 0)



# *K*-NN : Speeding Up the Classification

- ◆ Sophisticated algorithms for NN search:
  - Classical problem in Comp. Geometry
  - k-D trees
- ◆ Removing the samples from the training class  $\mathcal{T}$  which do not change the result of classification
  - Exactly: using Voronoi diagram
  - Approximately: E.g. use Gabriel graph instead of Voronoi
  - Condensation algorithm: iterative, also approximate.

# Condensation Algorithm

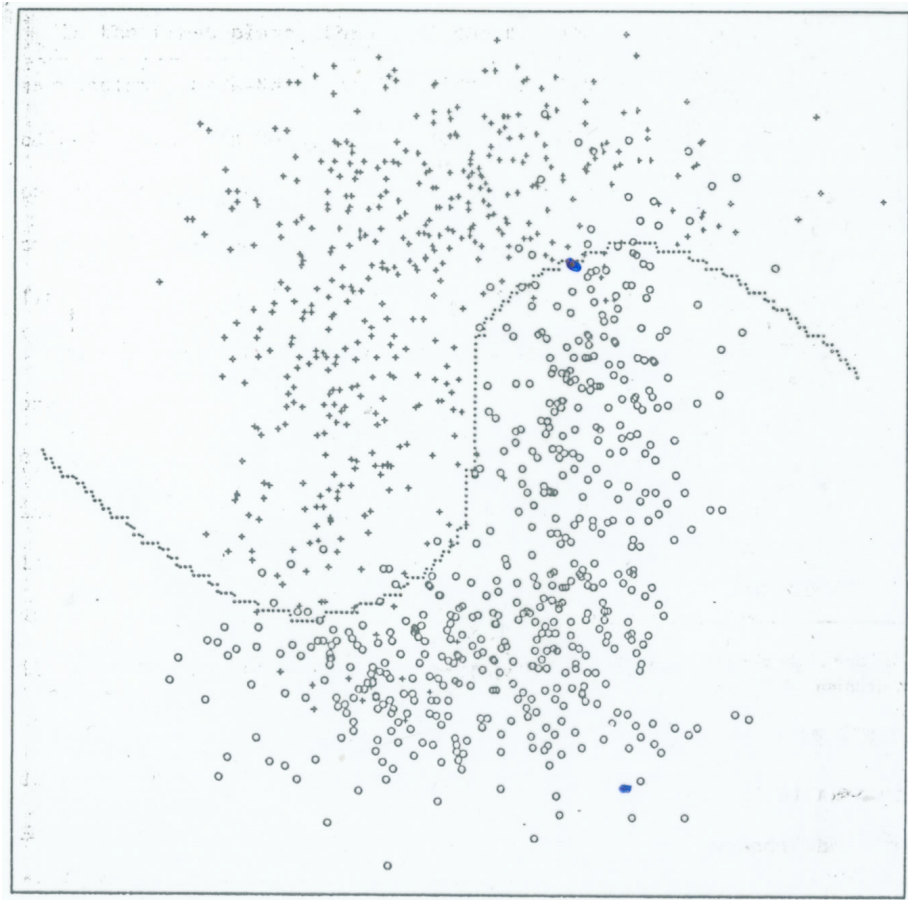
**Input:** The training set  $\mathcal{T}$ .

## Algorithm

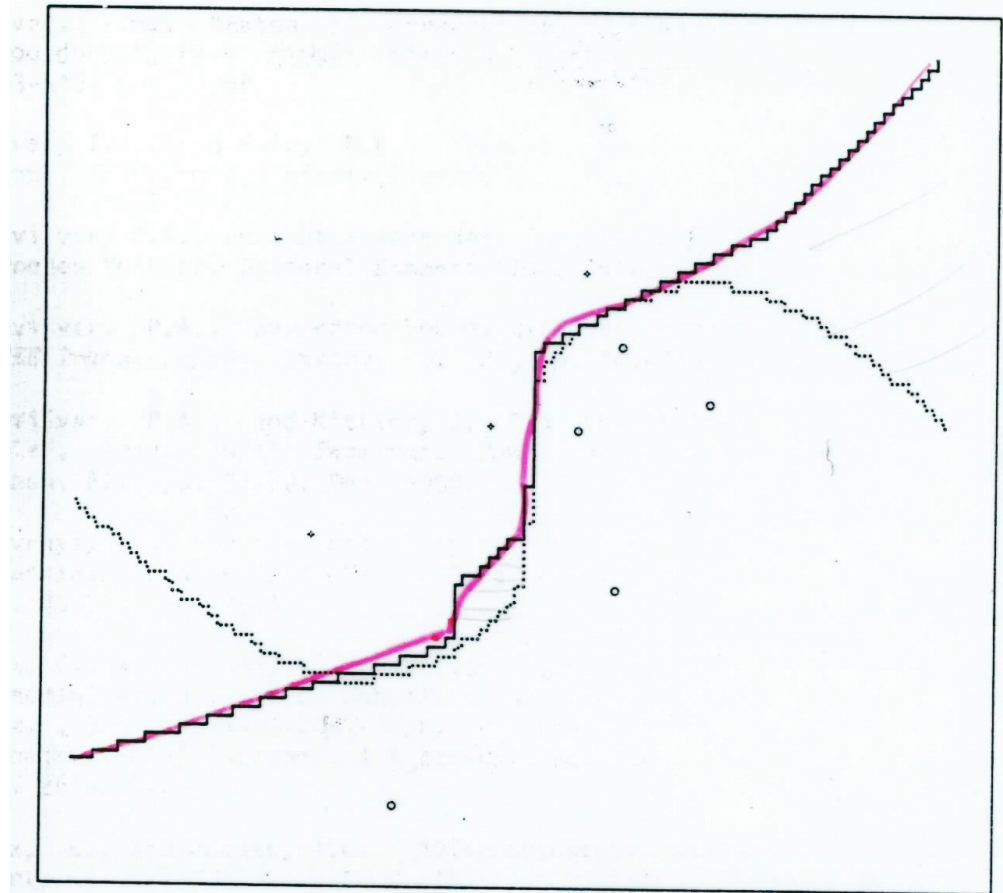
1. Create two lists,  $A$  and  $B$ . Insert a randomly selected sample from  $\mathcal{T}$  to  $A$ . Insert the rest of the training samples to  $B$ .
2. Classify samples from  $B$  using 1NN with training set  $A$ . If an  $x \in B$  is mis-classified, move it from  $B$  to  $A$ .
3. If a move has been triggered in Step 2., goto Step 2.

**Output:**  $A$  (the condensed training set for 1NN classification)

# Condensation Algorithm, Example



The training dataset



The dataset after the condensation.  
Shown with the new decision boundary.

# 1-NN Classification Error

Recall that a classification error  $\bar{\epsilon}$  for strategy  $q: X \rightarrow R$  is computed as

$$\hat{\epsilon} = \int \sum_{k:q(x) \neq k} p(x, k) dx = \int \underbrace{\sum_{k:q(x) \neq k} p(k|x) p(x)}_{\epsilon(x)} dx = \int \epsilon(x) p(x) dx. \quad (4)$$

We know that the Bayesian strategy  $q_B$  decides for the highest posterior probability  $q(x) = \operatorname{argmax}_k p(k|x)$ , thus the partial error  $\epsilon_B(x)$  for a given  $x$  is

$$\epsilon_B(x) = 1 - \max_k p(k|x). \quad (5)$$

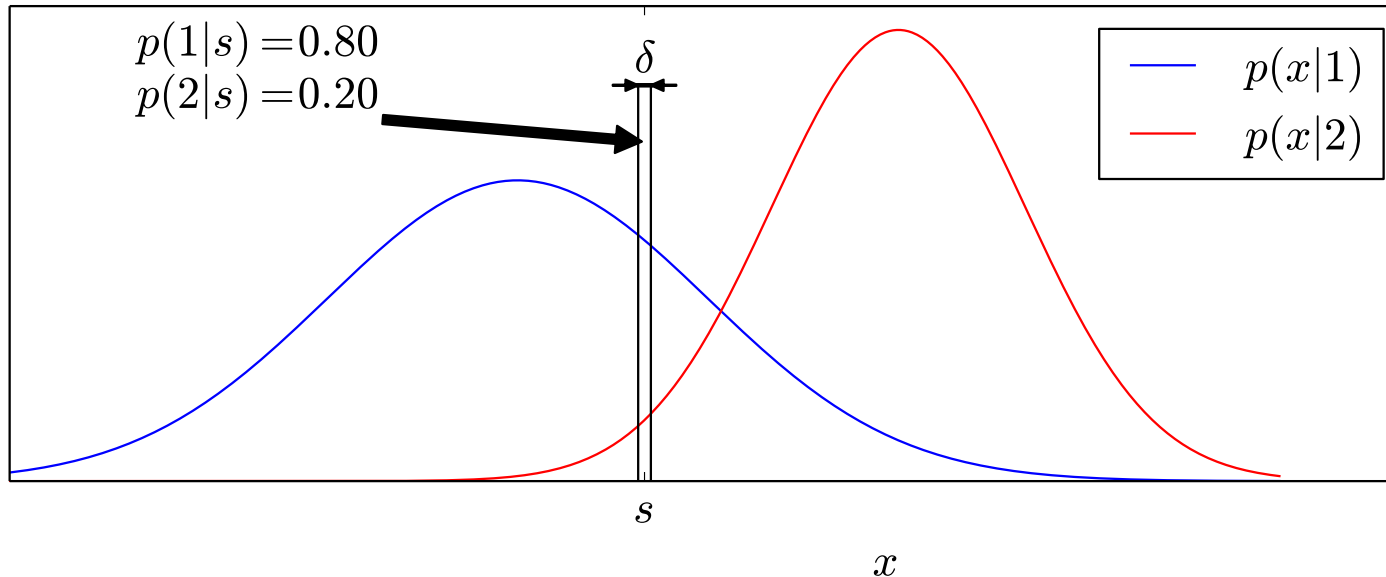
Assume the asymptotic case. We will show that the following bounds hold for the partial error  $\epsilon_{NN}(x)$  and classification error  $\bar{\epsilon}_{NN}$  in the 1-NN classification,

$$\epsilon_B(x) \leq \epsilon_{NN}(x) \leq 2\epsilon_B(x) - \frac{R}{R-1}\epsilon_B^2(x), \quad (6)$$

$$\bar{\epsilon}_B \leq \bar{\epsilon}_{NN} \leq 2\bar{\epsilon}_B - \frac{R}{R-1}\bar{\epsilon}_B^2, \quad (7)$$

where  $\hat{\epsilon}_B$  is the Bayes classification error and  $R$  is the number of classes.

# 1-NN Classification Error, Example (1)



Consider two distributions as shown, a small interval  $\delta$  on an  $x$ -axis, and a point  $s \in \delta$ . Assume that  $\delta \rightarrow 0$  and number of samples  $N \rightarrow \infty$ .

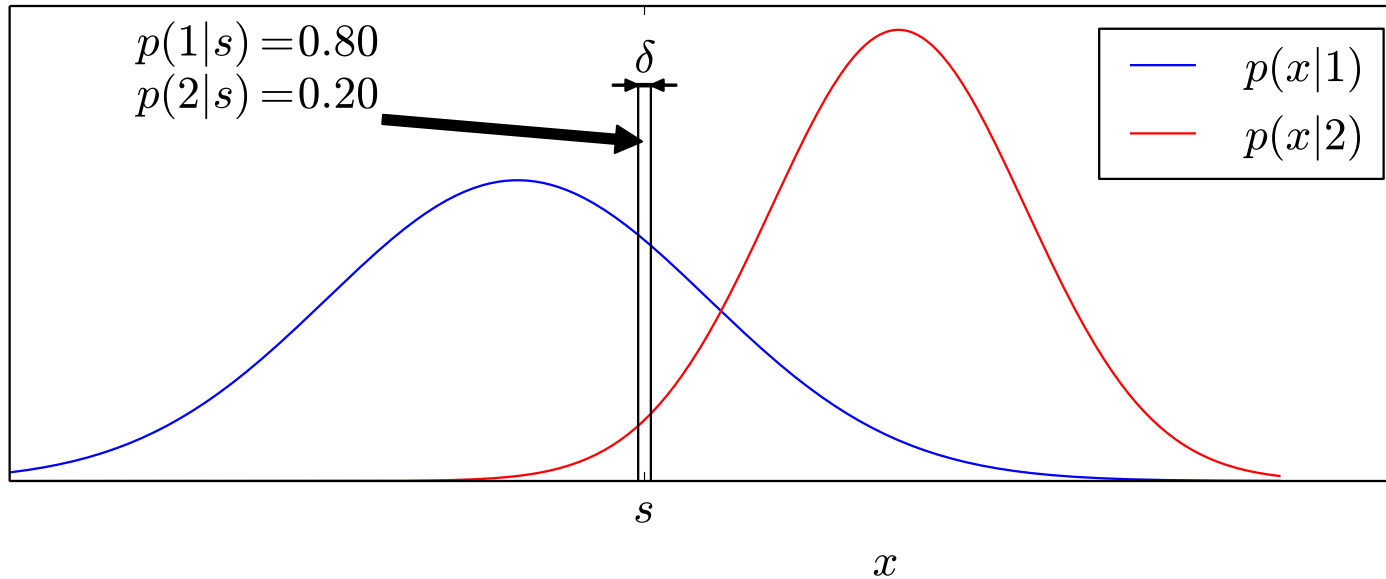
Observe the following:

$$p(1|s) = 0.8, \quad p(2|s) = 0.2, \quad (8)$$

$$p(NN = 1|s) = p(1|s) = 0.8, \quad p(NN = 2|s) = p(2|s) = 0.2, \quad (9)$$

where  $p(NN = k|s)$  is the probability that the 1-NN of  $s$  is from class  $k$  ( $k = 1, 2$ ) and thus  $s$  is classified as  $k$ .

# 1-NN Classification Error, Example (2)



The error  $\epsilon_{NN}(s)$  at  $s$  is

$$\epsilon_{NN}(s) = p(1|s)p(NN = 2|s) + p(2|s)p(NN = 1|s) \quad (10)$$

$$= 1 - p(1|s)p(NN = 1|s) - p(2|s)p(NN = 2|s) \quad (11)$$

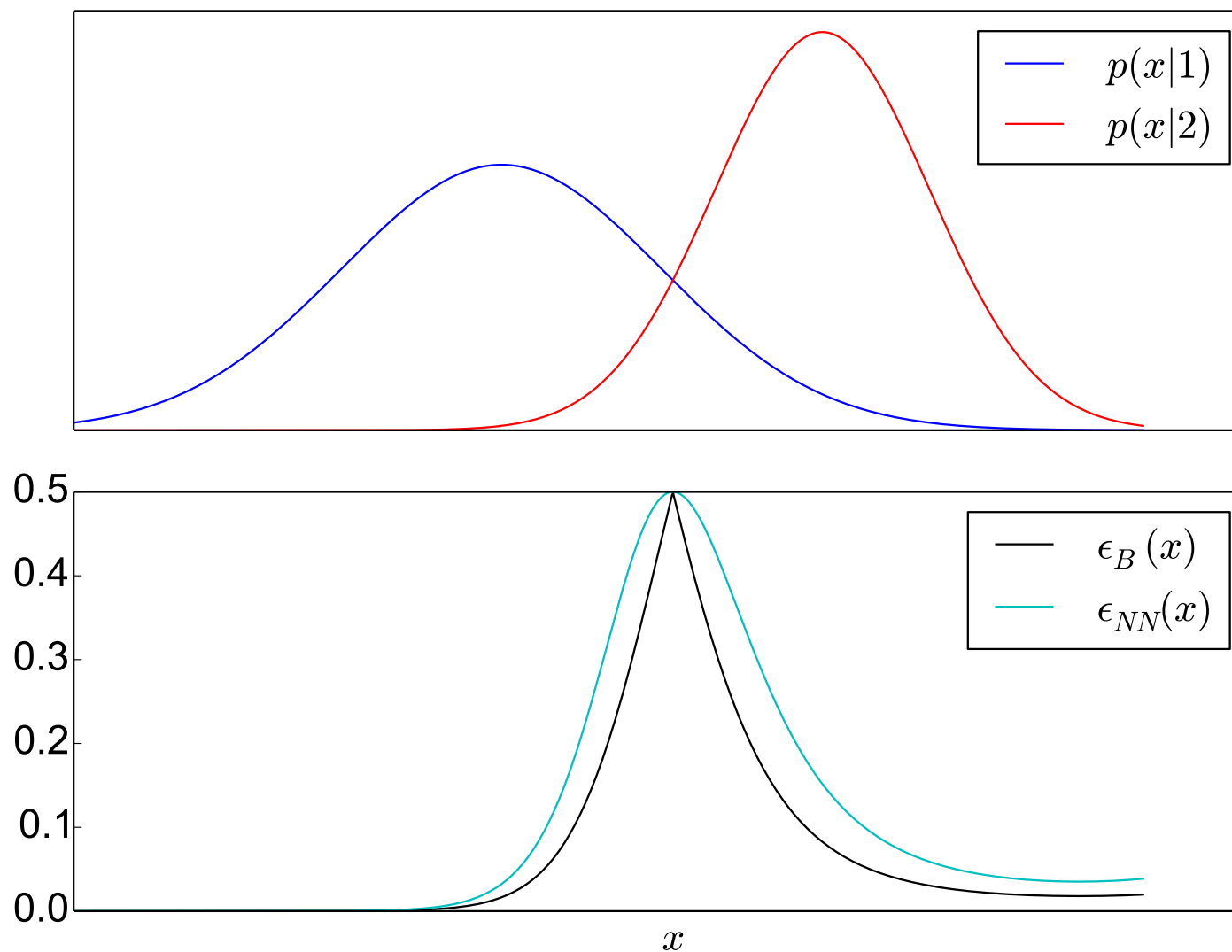
$$= 1 - p^2(1|s) - p^2(2|s). \quad (12)$$

Generally, for  $R$  classes, the error will be

$$\epsilon_{NN}(s) = 1 - \sum_{k \in R} p^2(k|s). \quad (13)$$

# 1-NN Classification Error, Example (3)

The two distributions and the partial errors  
(the Bayesian error  $\epsilon_B(x)$  and the 1-NN error  $\epsilon_{NN}(x)$ )



# 1-NN Classification Error Bounds (1)

Let us now return to the inequalities and prove them:

$$\epsilon_B(x) \leq \epsilon_{NN}(x) \leq 2\epsilon_B(x) - \frac{K}{K-1}\epsilon_B^2(x), \quad (14)$$

The **first** inequality follows from the fact that Bayes strategies are optimal.

To prove the **second** inequality, let  $P(x)$  denote the maximum posterior for  $x$ :

$$P(x) = \max_k p(k|x) \quad (15)$$

$$\Rightarrow \epsilon_B(x) = 1 - P(x). \quad (16)$$

Let us rewrite the partial error  $\epsilon_{NN}(x)$  using the Bayesian entities  $P(x)$  and  $q(x)$ :

$$\epsilon_{NN}(x) = 1 - \sum_{k \in R} p^2(k|x) = 1 - P^2(x) - \sum_{k \neq q(x)} p^2(k|x). \quad (17)$$

We know that  $p(q(x)|x) = P(x)$ , but the remaining posteriors can be arbitrary. Let us consider the worst case. i.e. set  $p(k|x)$  for  $k \neq q(x)$  such that Eq. (17) is maximized. This will provide the higher bound.



## 1-NN Classification Error Bounds (2)

There are the following constraints on  $p(k|x)$  ( $k \neq q(x)$ ):

$$\sum_{k \neq q(x)} p(k|x) + P(x) = 1 \quad (\text{posteriors sum to } 1) \quad (18)$$

$$\sum_{k \neq q(x)} p^2(k|x) \rightarrow \min \quad (19)$$

It is easy to show that this optimization problem is solved by setting all the posteriors to the same number. Thus,

$$p(k|x) = \frac{1 - P(x)}{R - 1} = \frac{\epsilon_B(x)}{R - 1} \quad (k \neq q(x)) \quad (20)$$

The higher bound can then be rewritten in terms of the Bayes partial error  $\epsilon_B(x) = 1 - P(x)$ :

$$\epsilon_{NN}(x) \leq 1 - P^2(x) - \sum_{k \neq q(x)} p^2(k|x) = 1 - (1 - \epsilon_B(x))^2 - (R - 1) \frac{\epsilon_B^2(x)}{(R - 1)^2}. \quad (21)$$

## 1-NN Classification Error Bounds (3)

$$\epsilon_{NN}(x) \leq 1 - P^2(x) - \sum_{k \neq q(x)} p^2(k|x) = 1 - (1 - \epsilon_B(x))^2 - \frac{\epsilon_B^2(x)}{R-1}. \quad (22)$$

After expanding this, we get

$$\epsilon_{NN}(x) \leq 1 - (1 - \epsilon_B(x))^2 - \frac{\epsilon_B^2(x)}{(R-1)} \quad (23)$$

$$= 1 - 1 + 2\epsilon_B(x) - \epsilon_B^2(x) - \epsilon_B^2(x) \frac{R}{R-1} \quad (24)$$

$$= 2\epsilon_B(x) - \epsilon_B^2(x) \frac{K}{K-1} \quad (25)$$

**Note** that for  $R = 2$ , the bound is tight because using  $\epsilon_B(x) = 1 - P(x)$  in Eq. (22) gives

$$\epsilon_{NN}(x) \leq 1 - P^2(x) - \frac{(1 - P(x))^2}{1} = \epsilon_{NN}(x). \quad (26)$$

# 1-NN Classification Error Bounds (4)

The inequality for the local errors has been proven:

$$\epsilon_{NN}(x) \leq 2\epsilon_B(x) - \epsilon_B^2(x) \frac{K}{K-1} \quad (27)$$

Is there a similar higher bound for the classification error  $\bar{\epsilon}_{NN} = \int \epsilon_{NN}(x)p(x)dx$ , based on the Bayes error  $\bar{\epsilon}_B = \int \epsilon_B(x)p(x)dx$ ?

Multiplying Eq. (28) by  $p(x)$ , and integrating, gives

$$\bar{\epsilon}_{NN} \leq 2\bar{\epsilon}_B - \frac{K}{K-1} \int \epsilon_B^2(x)p(x)dx \quad (28)$$

Let us use the known identity (where  $E(\cdot)$  is the expectation operator)

$$\text{var}(x) = E(x^2) - E^2(x) \quad (\geq 0) \quad (29)$$

Thus,  $\int \epsilon_B^2(x)p(x)dx \geq \left(\int \epsilon_B(x)p(x)dx\right)^2$ , and

$$\bar{\epsilon}_{NN} \leq 2\bar{\epsilon}_B - \frac{K}{K-1} \int \epsilon_B^2(x)p(x)dx \leq 2\bar{\epsilon}_B - \frac{K}{K-1} \bar{\epsilon}_B^2 \quad (30)$$

## $K$ -NN Classification Error Bound

It can be shown that for  $K$ -NN, the following inequality holds:

$$\bar{\epsilon}_{KNN} \leq \bar{\epsilon}_B + \bar{\epsilon}_{1NN} / \sqrt{K} \text{ const} \quad (31)$$

# Edit algorithm

The primary goal of this method is to reduce the classification error (not the speed-up of classification.)

**Input:** The training set  $\mathcal{T}$ .

## Algorithm

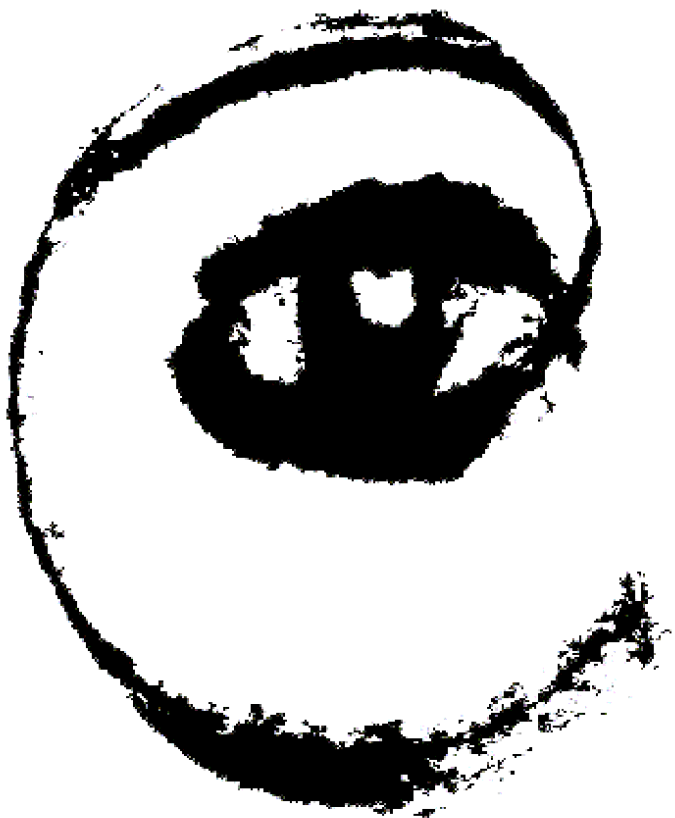
1. Partition  $\mathcal{T}$  to two sets,  $A$  and  $B$  ( $\mathcal{T} = A \cup B$ ,  $A \cap B = \emptyset$ .)
2. Classify samples in  $B$  using **K**NN with training set  $A$ . Remove all samples from  $B$  which have been mis-classified.

**Output:**  $B$  the training set for **1**NN classification.

Asymptotic property:

$$\bar{\epsilon}_{edit} = \bar{\epsilon}_B \frac{1 - \bar{\epsilon}_B}{1 - \bar{\epsilon}_{KNN}} \quad (32)$$

If  $\bar{\epsilon}_{KNN}$  is small (e.g. 0.05) then the edited 1NN is quasi-Bayes (almost the same performance as Bayesian Classification.)



m p

