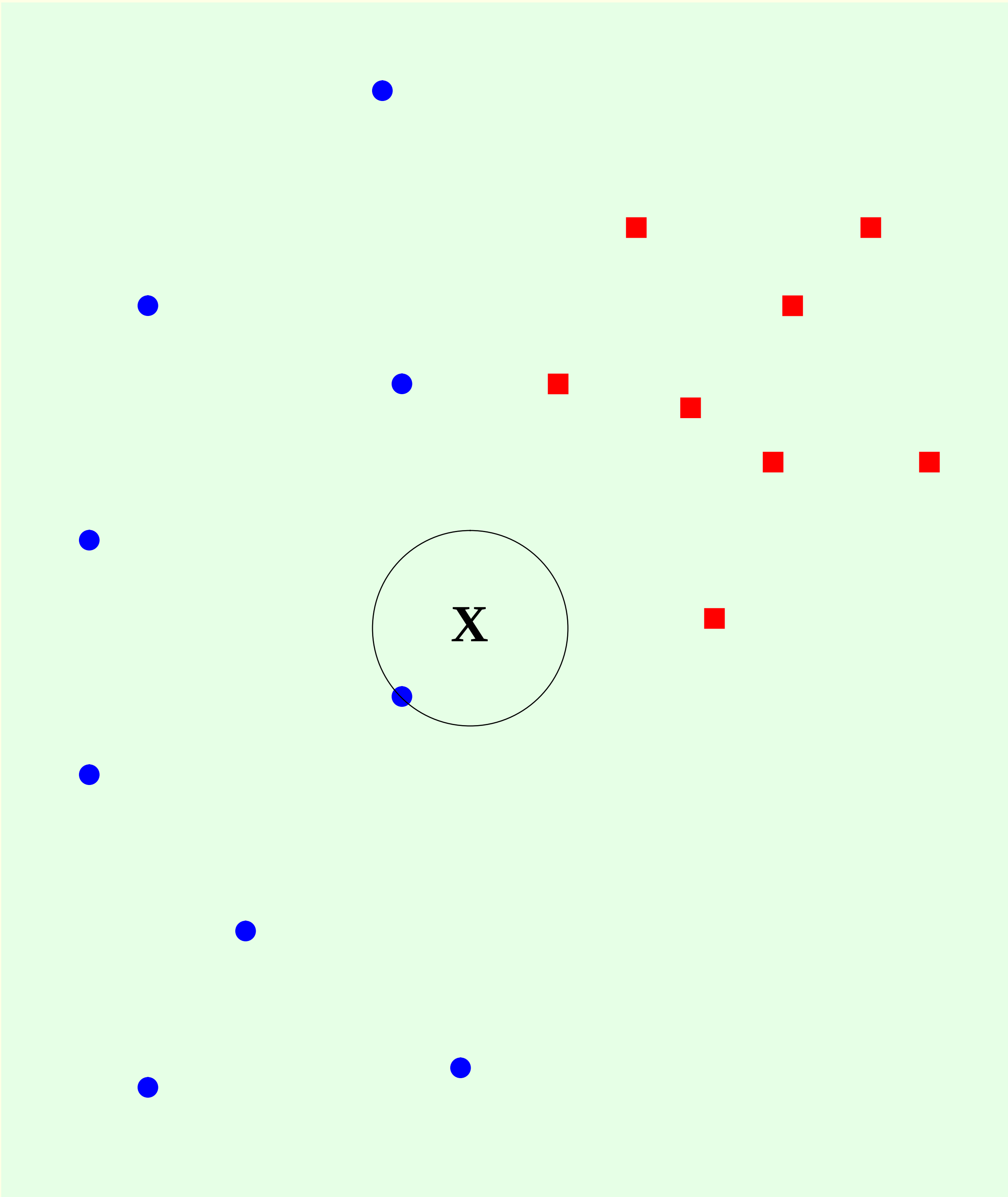
Proximity Graph Methods for Data Mining

Godfried Toussaint

Computational Geometry Laboratory

McGill University

The Nearest Neighbor Decision Rule



The Nearest Neighbor Decision Rule - *cont.*

1951 - *conceived by* E. Fix *and* J. Hodges

1967 - T. Cover *and* P. Hart *gave asymptotic performance bounds in terms of the Bayes error for "nice" distributions.*

$$P_e \leq P_e(1-NN) \leq 2P_e\left(1-P_e\right)$$

These bounds are proved for all distributions by:

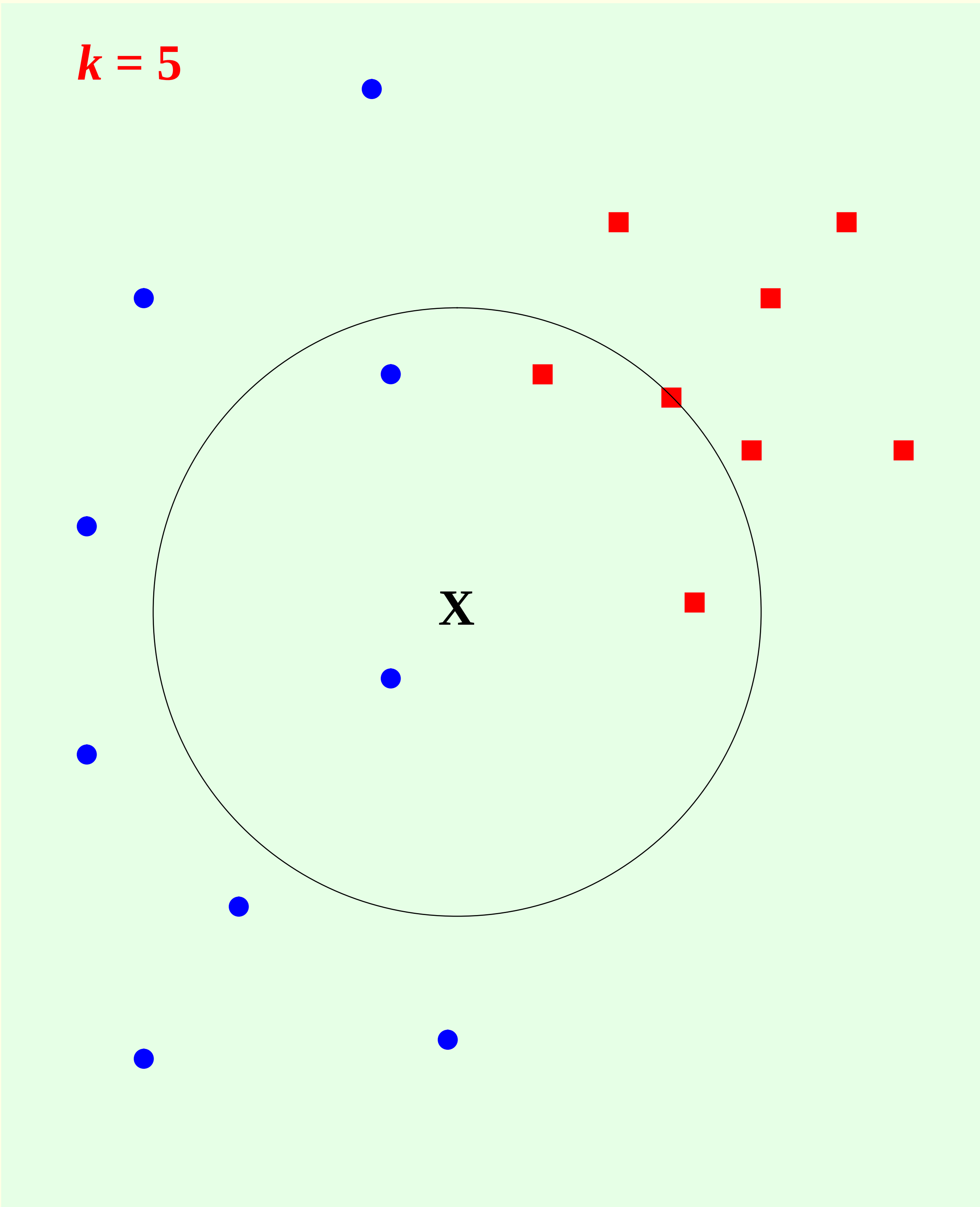
1977 - C. Stone

1981 - L. Devroye

The 1-NN rule has a long history of avoidance in practice based on several incorrect assumptions:

1. *All the training data must be stored.*
2. *Distances between the unknown X and all the training data must be computed to classify X.*
3. *It is unsuitable for implementation in parallel.*

The *k*-Nearest Neighbor Decision Rule



The *k*-Nearest Neighbor Decision Rule - *cont.*

1981 - L. Devroye *showed that for training data* $\{X_1, X_2, \ldots, X_n\}$, *and all distributions*

$$P_e(k-NN) \rightarrow P_e$$

when:

1. *n approaches infinity*
2. *k approaches infinity*
3. *k/n approaches zero*

Extended also to the case when the choice of k is dependent on the training data
(Devroye, Gyorfy & Lugosi, 1996).

In practice n and k are finite and a number of additional questions arize.

The *k*-Nearest Neighbor Decision Rule in Practice: Finite Sample Size

1. *How can the storage of the training set be reduced without degrading performance?*
2. *How should the reduced training set be selected to represent the different classes?*
3. *How large should k be? How should k be chosen?*
4. *Should all k neighbors be weighted equally? If not, how should weights be chosen?*
5. *Should all the measurements be weighted equally? If not, how should these weights be chosen?*
6. *How can the rule be made robust to overlapping classes and noise?*
7. *How can the neighbors of a new point be computed efficiently?*
8. *What is the smallest neural network that can implement the 1-NN rule? (minimum number of nodes, neurons, TLU's)*

The Condensed Nearest Neighbor Rule

P. Hart - 1968

Given the training set $\{\mathbf{X}\} = \{X_1, X_2, \ldots, X_n\}$ *and two (initially empty) storage locations* STORE *and* GRABBAG.

1. *Transfer a random element from* $\{\mathbf{X}\}$ *into* STORE.
2. *For each remaining element in* $\{\mathbf{X}\}$: *classify it using the 1-NN rule with* STORE *and if classified correctly put it in* GRABBAG. *Otherwise put it in* STORE.
3. *For each element in* GRABBAG: *classify it using the 1-NN rule with* STORE *and if classified incorrectly transfer to* STORE.
4. *Repeat step 3 until no transfers are made from* GRABBAG *to* STORE.
5. *Exit with* STORE *as the condensed subset of* $\{\mathbf{X}\}$.

Properties:

a) STORE *is training-set consistent.*

b) STORE *can be computed in* $O(n^3)$ *time.*

Condensing with Nearest-Unlike Neighbors

Belur Dasarathy - 1994

Given an element X_i *of the training set* $\{\mathbf{X}\} = \{X_1, X_2, \ldots, X_n\}$, *the element of* $\{\mathbf{X}\}$ *closest to* X_i *but belonging to a different class is called a nearest unlike neighbor.*

The nearest unlike subset of $\{\mathbf{X}\}$ *consists of all elements of* $\{\mathbf{X}\}$ *that are nearest unlike neighbors of at least one element of* $\{\mathbf{X}\}$.

Dasarathy gives a complicated algorithm called MCS *and conjectures it gives a* ***M****inimal* ***C****onsistent* ***S****ubset but counter-examples are found*.

Gordon Wilfong - 1991

Proves computing MCS *is* NP-*Complete for 3 or more classes.*

In Machine Learning literature condensing is called instance pruning.

Wilson and Martinez - 1997

nearest unlike neighbor is called nearest enemy and 3 algorithms are given.

Combined Editing and Condensing

B. Dasarathy, J. Sánchez and S. Townsend - 2000

In-depth experimental comparison of 26 algorithms which are combinatorial combinations of different editing and condensing algorithms.

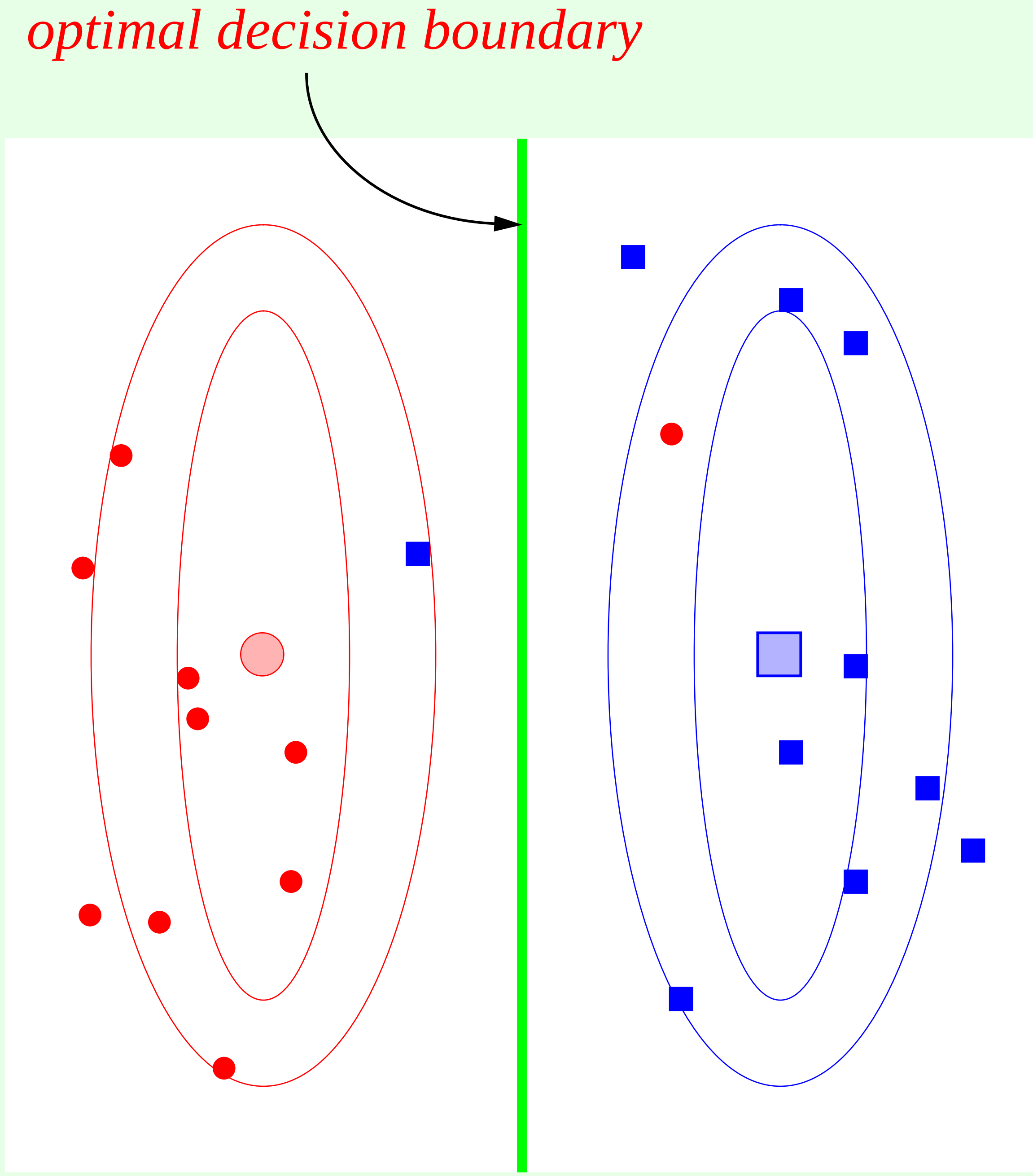
Results:

The best algorithm is obtained by performing:

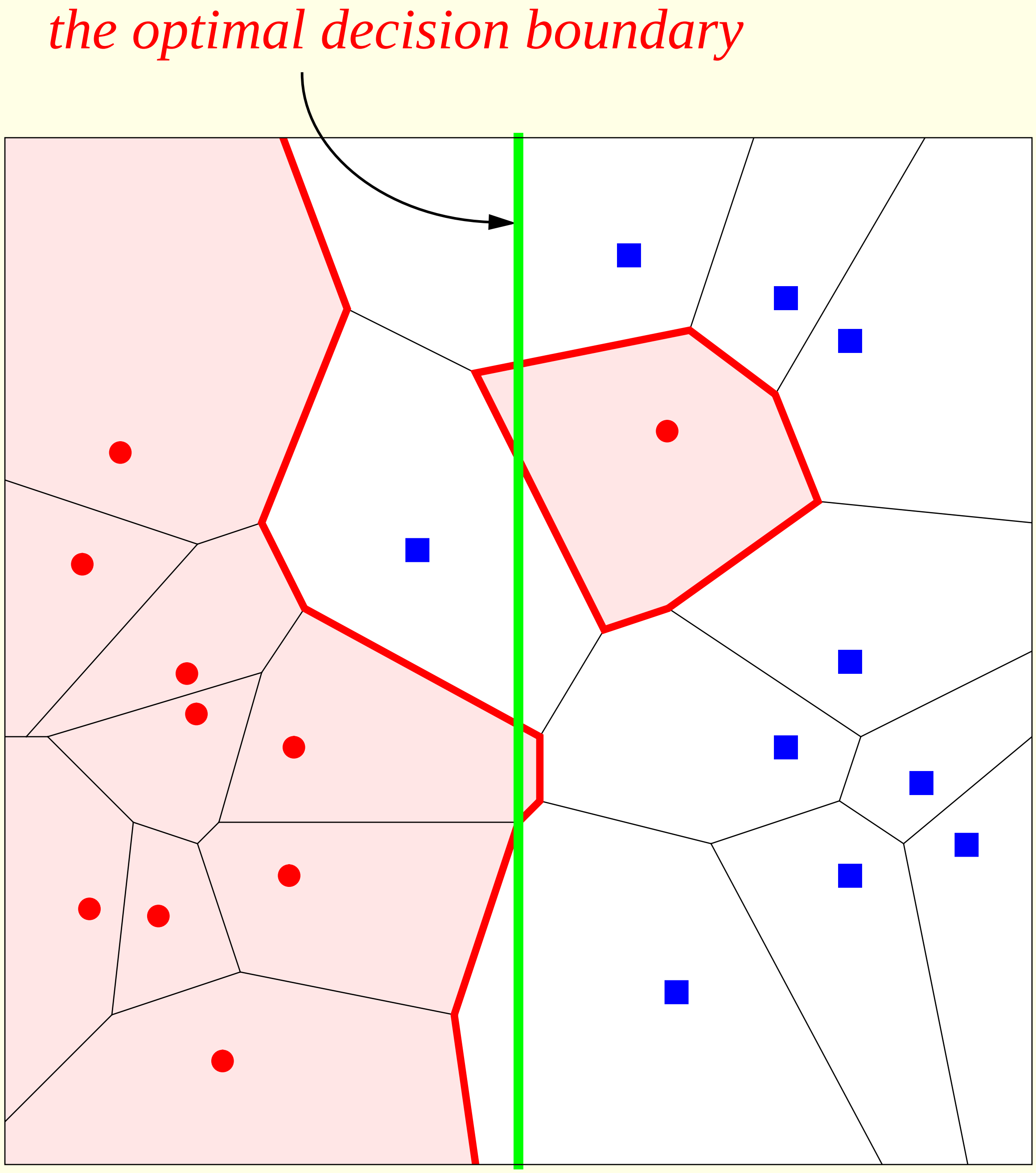
First: *proximity-graph editing* (RNG or GG)

Second: MCS *condensing*

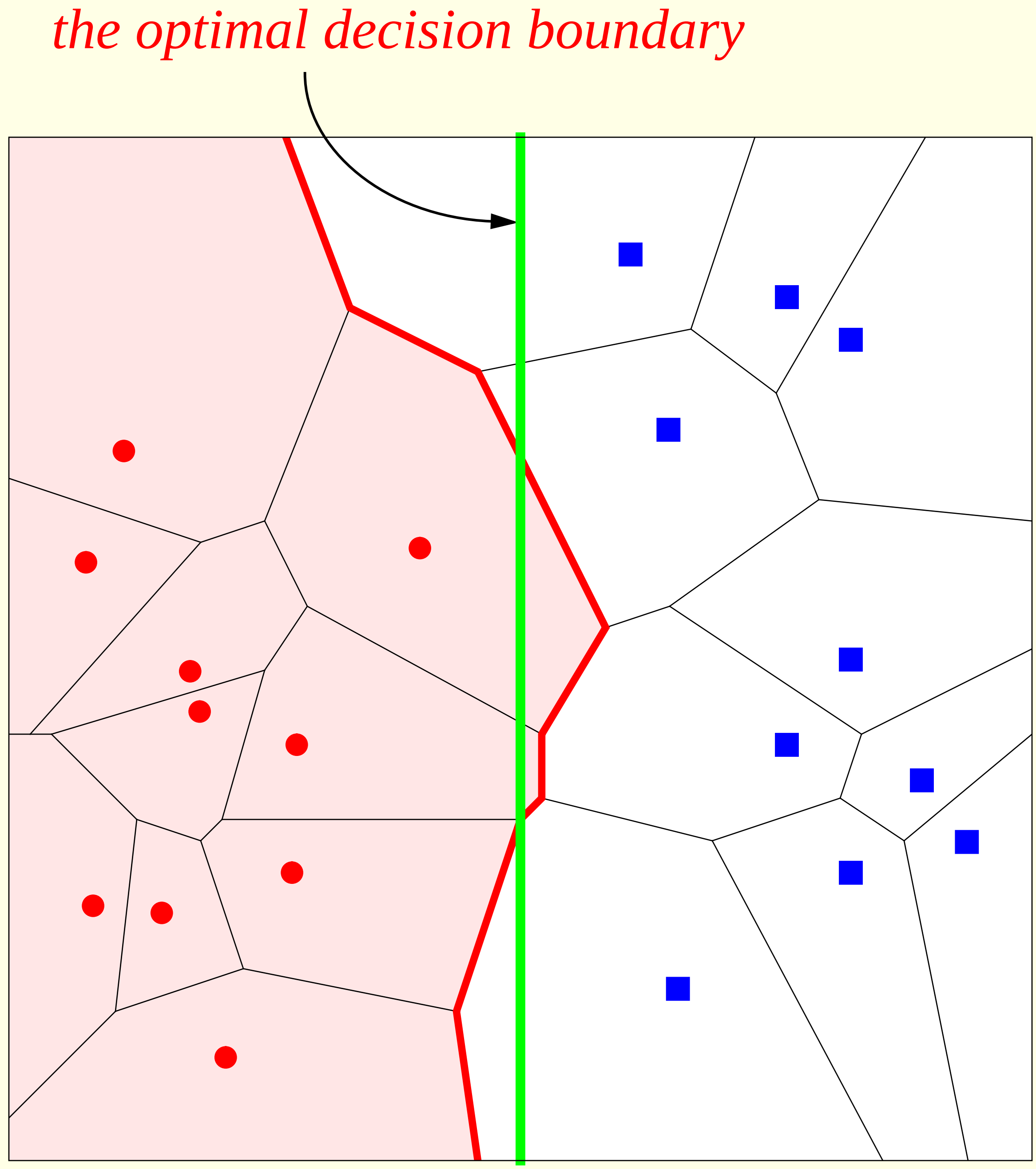
The **Optimal** Classifier for Gaussian Data



The *1-NN* Classifier for the Gaussian Data



Editing the *1-NN* Classifier for the Gaussian Data



The Edited Nearest Neighbor Rule

Denis L. Wilson - 1972

Given the training set $\{\mathbf{X}\} = \{X_1, X_2, \ldots, X_n\}$.

PREPROCESSING

I. for each i:

1. *Find the* k*-nearest neighbors to* X_i *among* $\{X_1, X_2, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n\}$.
2. *Classify* X_i *to the class associated with the largest number of points among the* k*-nearest neighbors, breaking ties randomly.*

II. edit $\{\mathbf{X}\} = \{X_1, X_2, \ldots, X_n\}$ *by deleting all the elements misclassified in the foregoing.*

DECISION RULE

1. *Classify a new pattern* X *using the 1-NN rule with the edited subset of* $\{\mathbf{X}\}$.

Editing Nearest Neighbor Rules with Proximity Graphs

J. S. Sánchez, F. Pla & F. J. Ferri - 1997

Given the training set $\{\mathbf{X}\} = \{X_1, X_2,...,X_n\}$.

PREPROCESSING

I. *Compute the proximity graph of* $\{\mathbf{X}\}$.

II. for each *i*:

Classify X_i *to the class associated with the largest number of points among the graph neighbors, breaking ties randomly.*

III. edit $\{\mathbf{X}\} = \{X_1, X_2,...,X_n\}$ *by deleting all the points misclassified in the foregoing.*

DECISION RULE

1. *Classify a new pattern* X *using the 1-NN rule with the edited subset of* $\{\mathbf{X}\}$.

Recognition acuracy:

Editing: *relative neighborhood graph was best*

Editing & Condensing: *Gabriel graph was best*

Data reduction: *similar*

Proximity Graph Neighbor Decision Rules

J. S. Sánchez, F. Pla & F. J. Ferri - 1997
L. Devroye, L. Györfy and G. Lugosi - 1996
Given the training set $\{\mathbf{X}\} = \{X_1, X_2, \ldots, X_n\}$.

DECISION RULE

Classify an unknown pattern Z *to the class associated with the largest number of points among the proximity graph neighbors of* Z *in* $\{\mathbf{X}\}$*, breaking ties randomly.*

This rule takes care of the selection of the size of k (number of neighbors) and how they are distributed around Z in a natural and fully automatic way.

They conclude that the Relative Neighbor Decision Rule is the best.

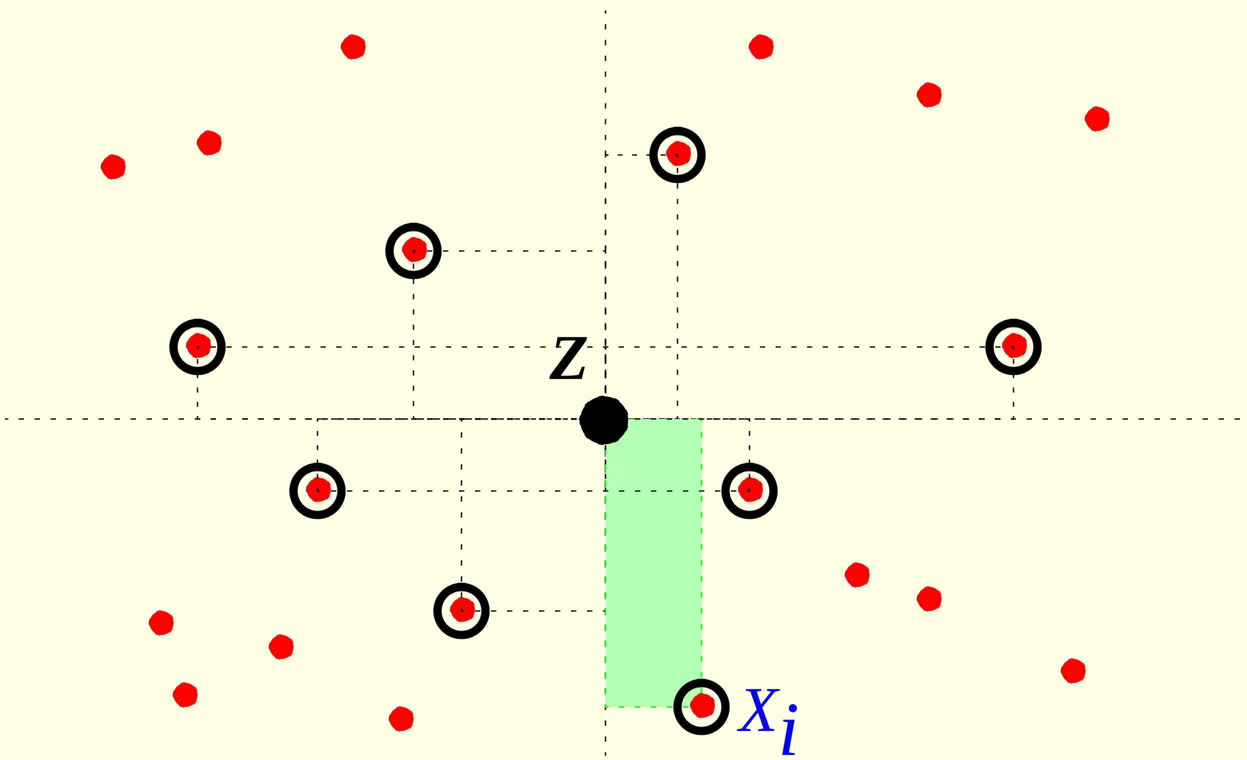
Devroye et al. have various theoretical results for the Gabriel nearest neighbor rule.

The Rectangle-of-Influence Neighbor Rule

M. Ichino and J. Sklansky - 1985
L. Devroye, L. Györfy and G. Lugosi - 1996
*(layered nearest neighbor rule - **scale invariant**)*

DECISION RULE

Classify an unknown pattern Z *to the class associated with the largest number of points among the rectangle-of-influence neighbors of* Z *in* {**X**}, *breaking ties randomly*.

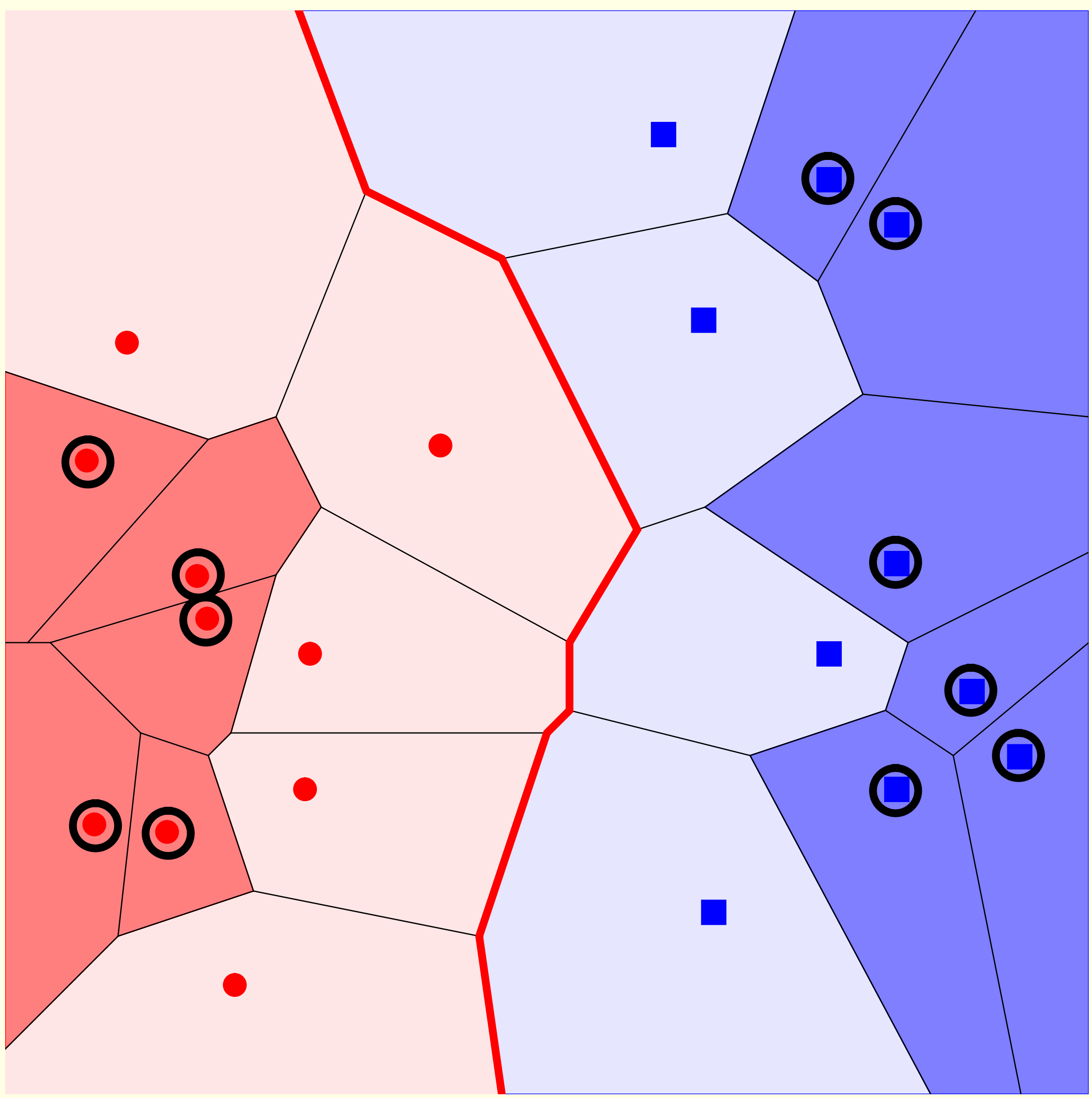


Devroye et al. showed that when there are no ties this rule is asymptotically Bayes optimal.

The *1-NN* rule with Voronoi Condensing --
The decision-boundary consistent subset

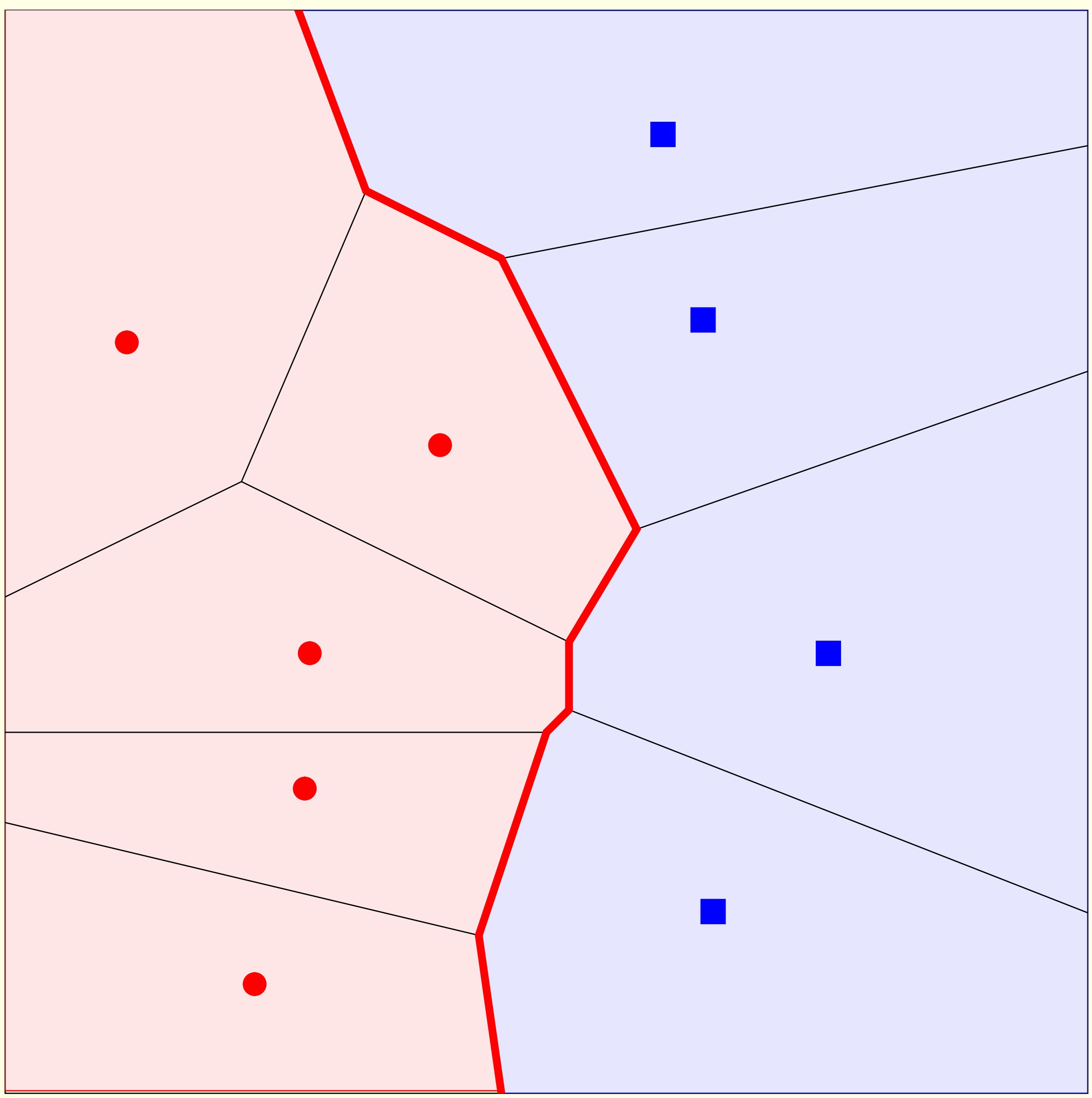
G. Toussaint & R. Poulsen - 1979

1. *Mark a point* X_i *if all its Voronoi neighbors belong to the same class as that of* X_i.
2. *Delete all marked points.*
3. *Use 1-NN rule on remaining set.*

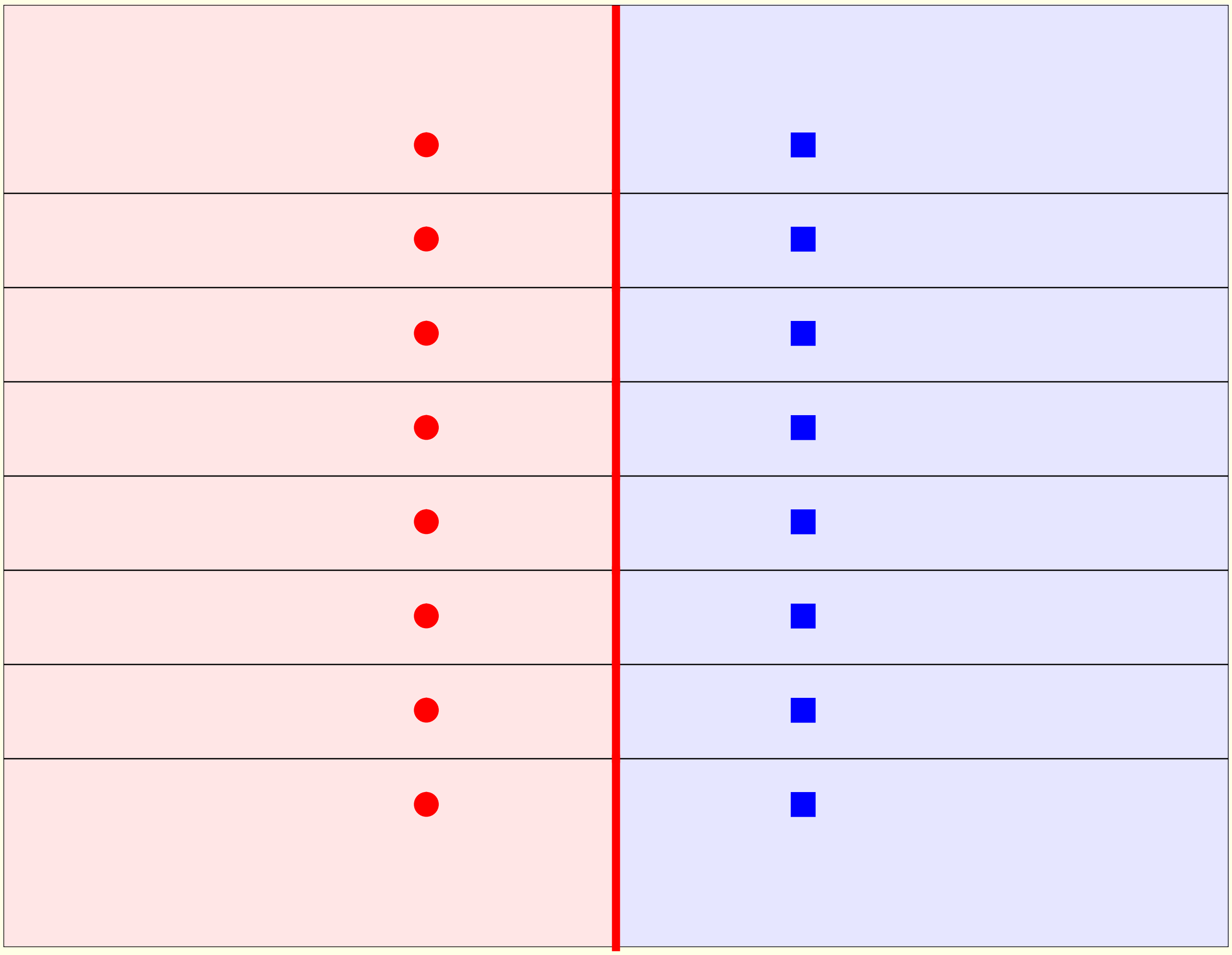


The resulting Voronoi condensed decision-boundary consistent subset

G. Toussaint & R. Poulsen- 1979



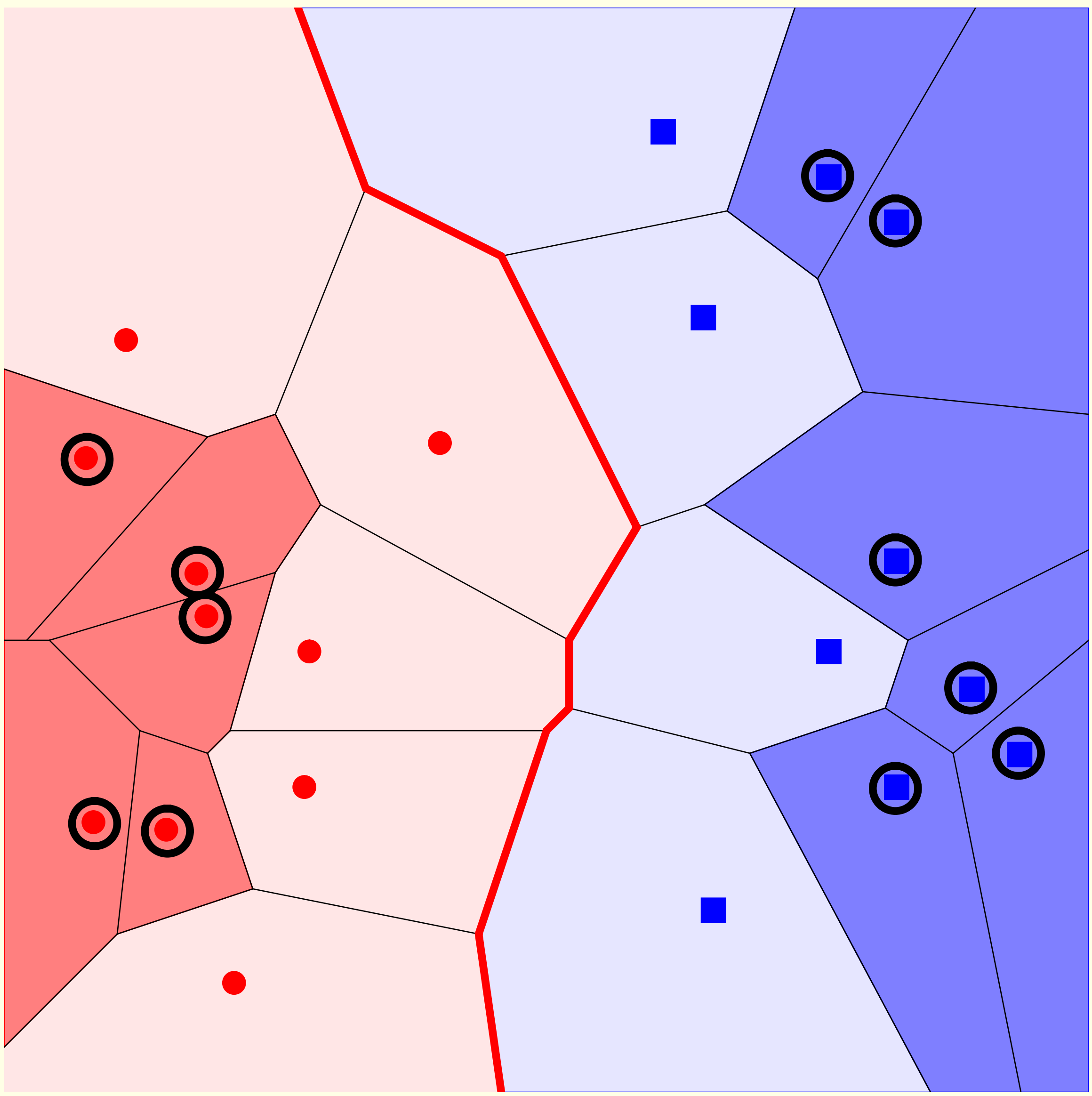
The Voronoi condensed subset is not necessarily the minimum-size consistent subset



Computing Nearest-Neighbor Decision Boundaries

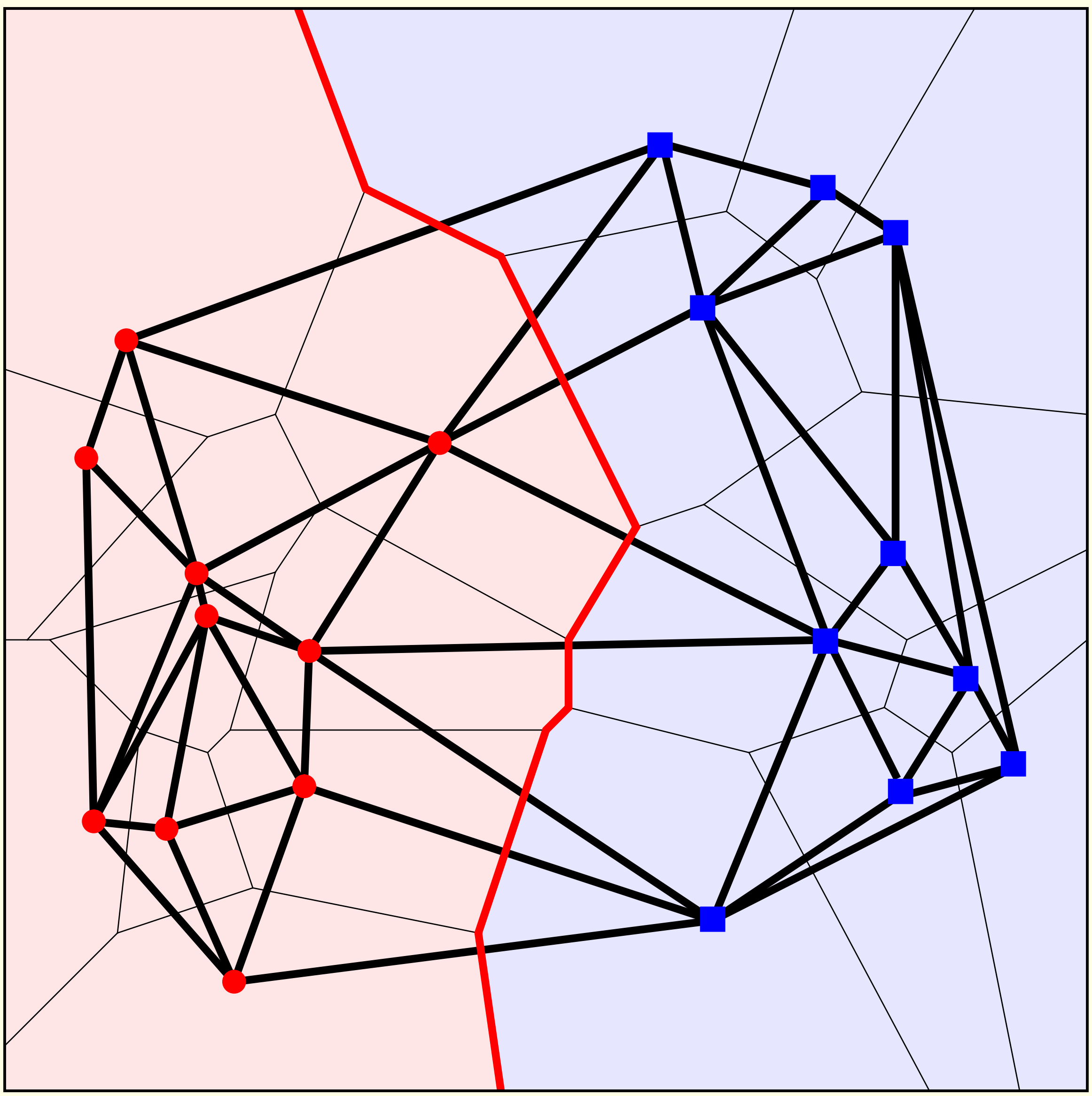
D. Bremner, E. Demaine, J. Erickson, J. Iacono, S. Langerman, P. Morin and G. Toussaint - 2003

In 2D: O(n log $\boldsymbol{k}$), where $\boldsymbol{k}$ is the number of points that contribute to the boundary.



Proximity-graph condensed subsets

G. Toussaint, B. Bhattacharya & R. Poulsen - 1985



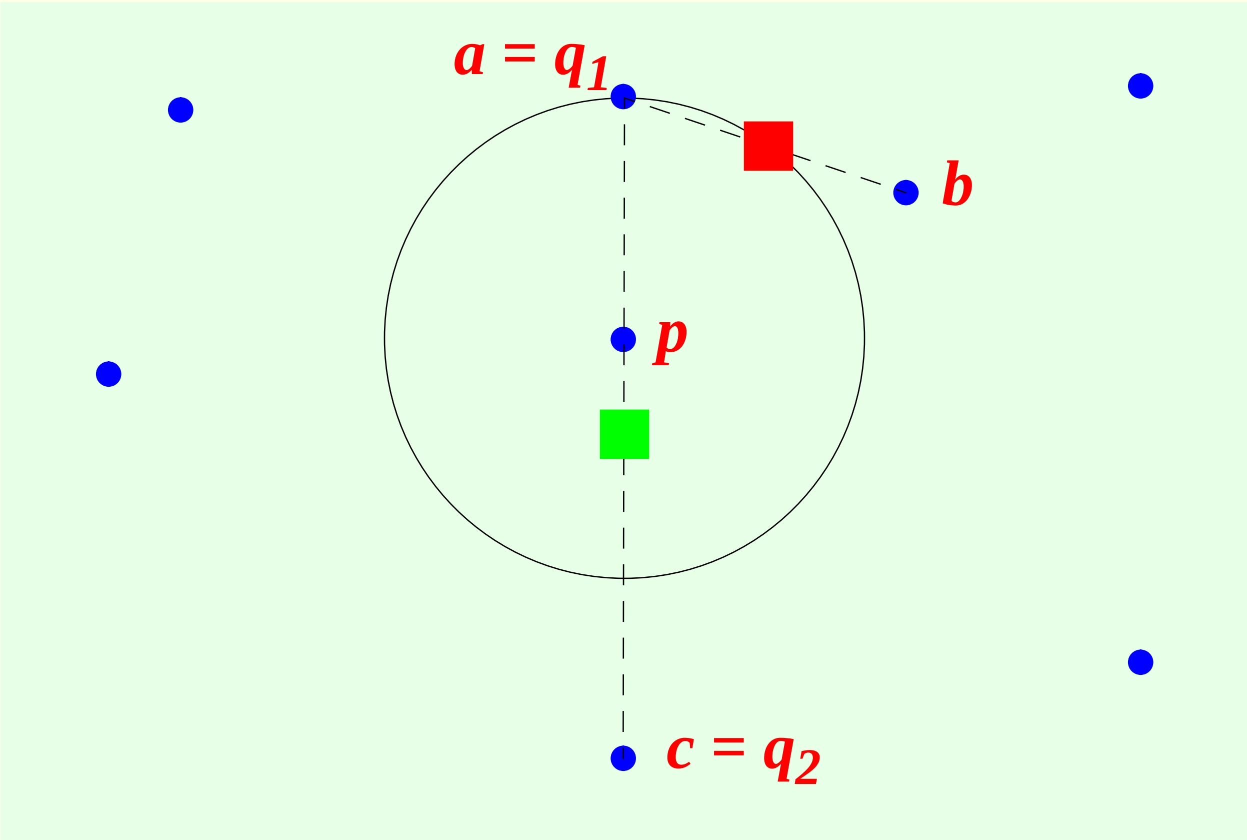
The Surrounding Neighborhood of a Point

The Nearest Centroid Neighborhood
B. Chaudhuri, 1996

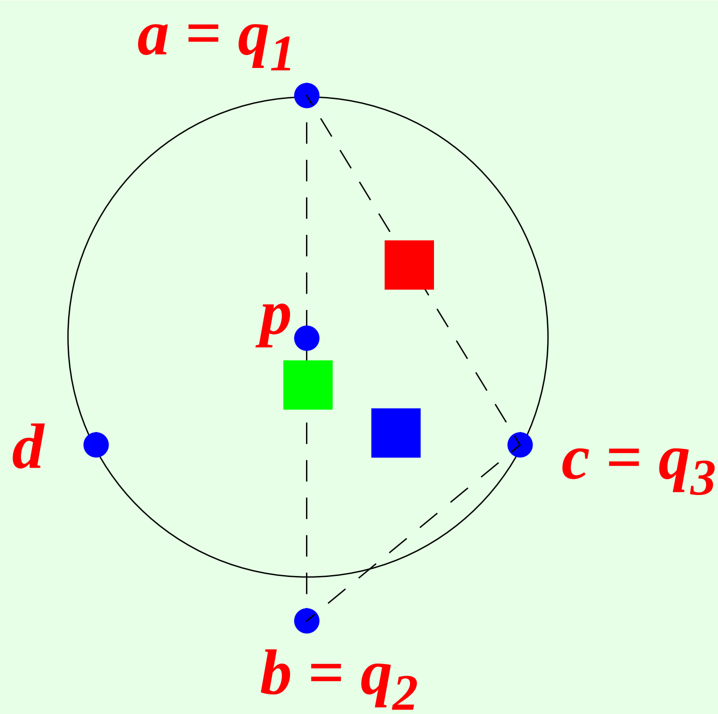
Given a training set ***T*** of ***n*** data points:

1. The *1st* centroid neighbor of a new point ***p*** is the closest point in ***T***.

2. For k=1,2,... the *k-th* centroid neighbor of ***p*** is the point $\boldsymbol{q_k}$ in ***T*** such that the centroid $\boldsymbol{Q_k}$ of $\boldsymbol{q_1, q_2, \ldots, q_k}$ is closest to ***p***.

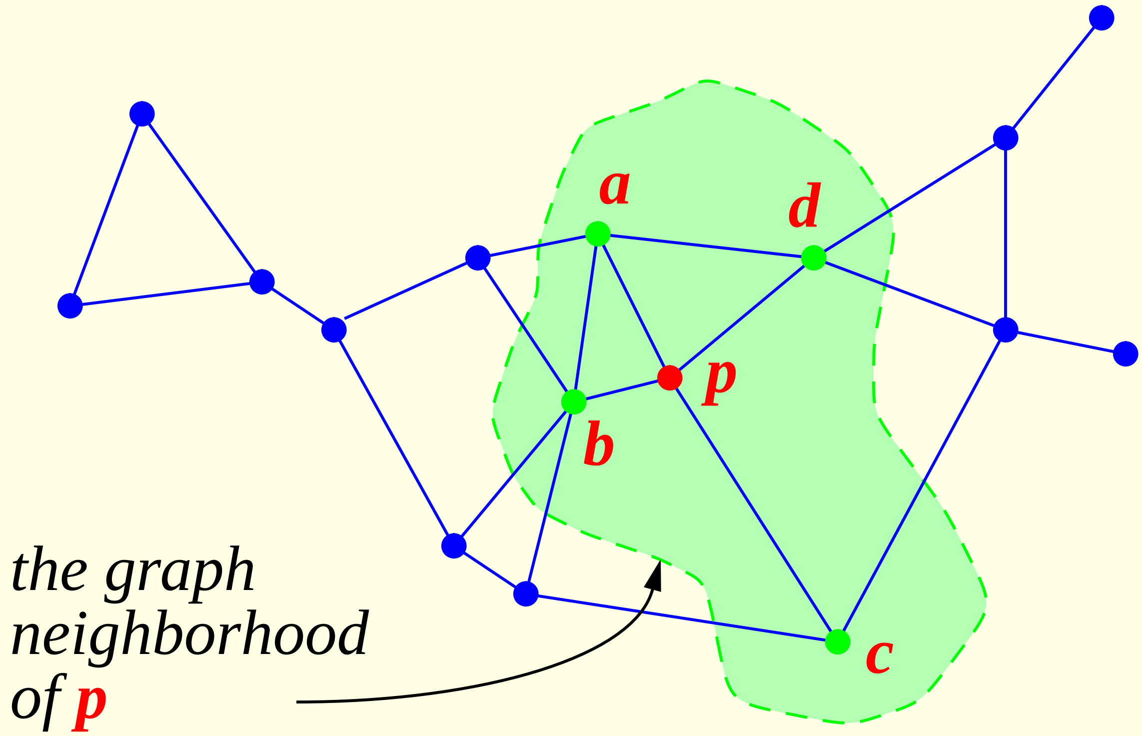


The *k nearest-centroid-neighbors* are not necessarily the *k neighbors with nearest centroid*.



Proximity-Graph-Neighbor Decision Rules

L. Devroye, L. Györfy and G. Lugosi - 1996
J. Sanchez, F. Pla and F. Ferri - 1997



Points ***a***,***b***,***c*** *and* ***d*** are ***graph neighbors*** of ***p.***

Proximity-graph-neighbor decision rules:

*Classify a new point **p** according to a majority vote of its graph neighbors.*

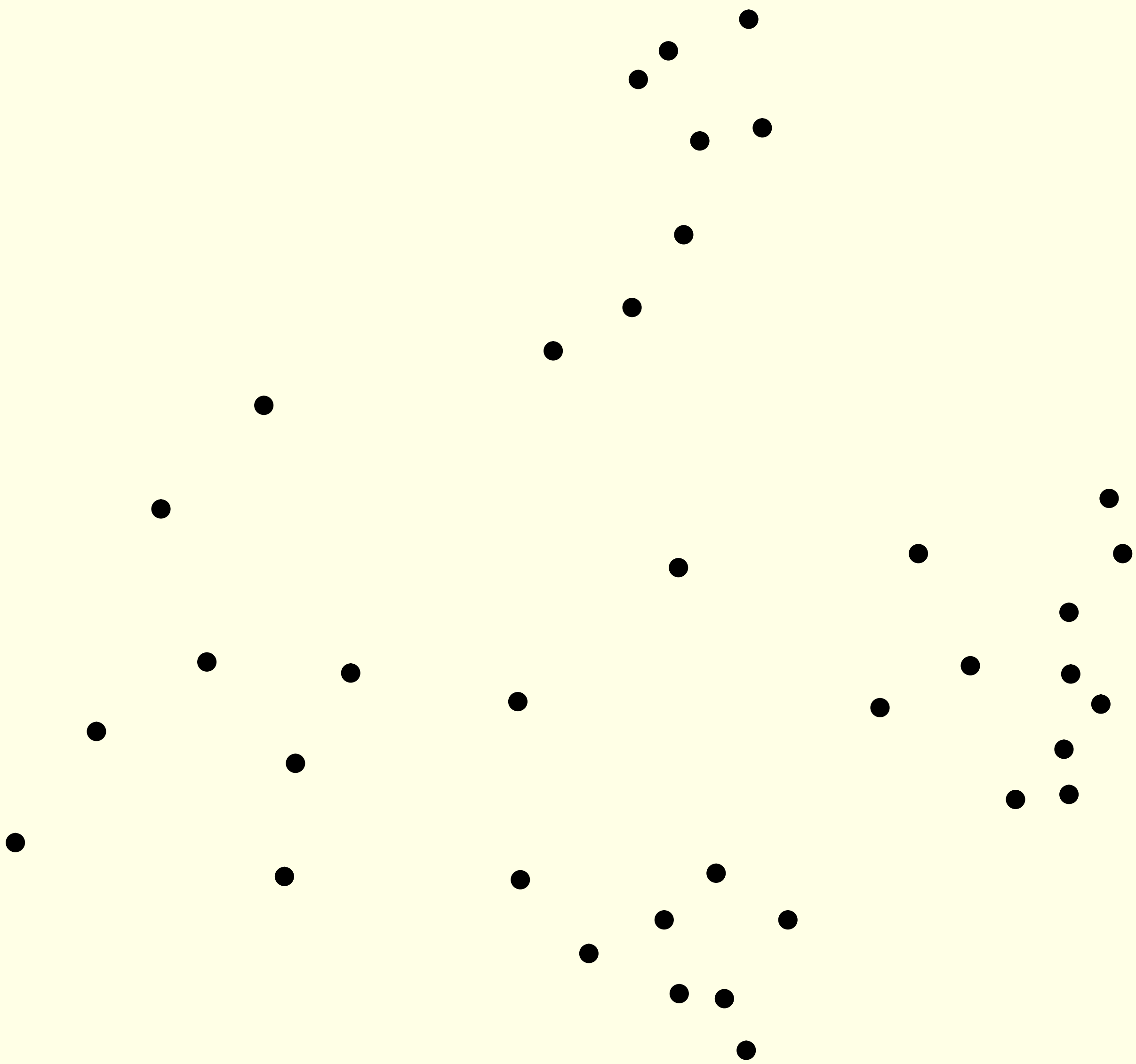
Identifying Competence-Critical Instances

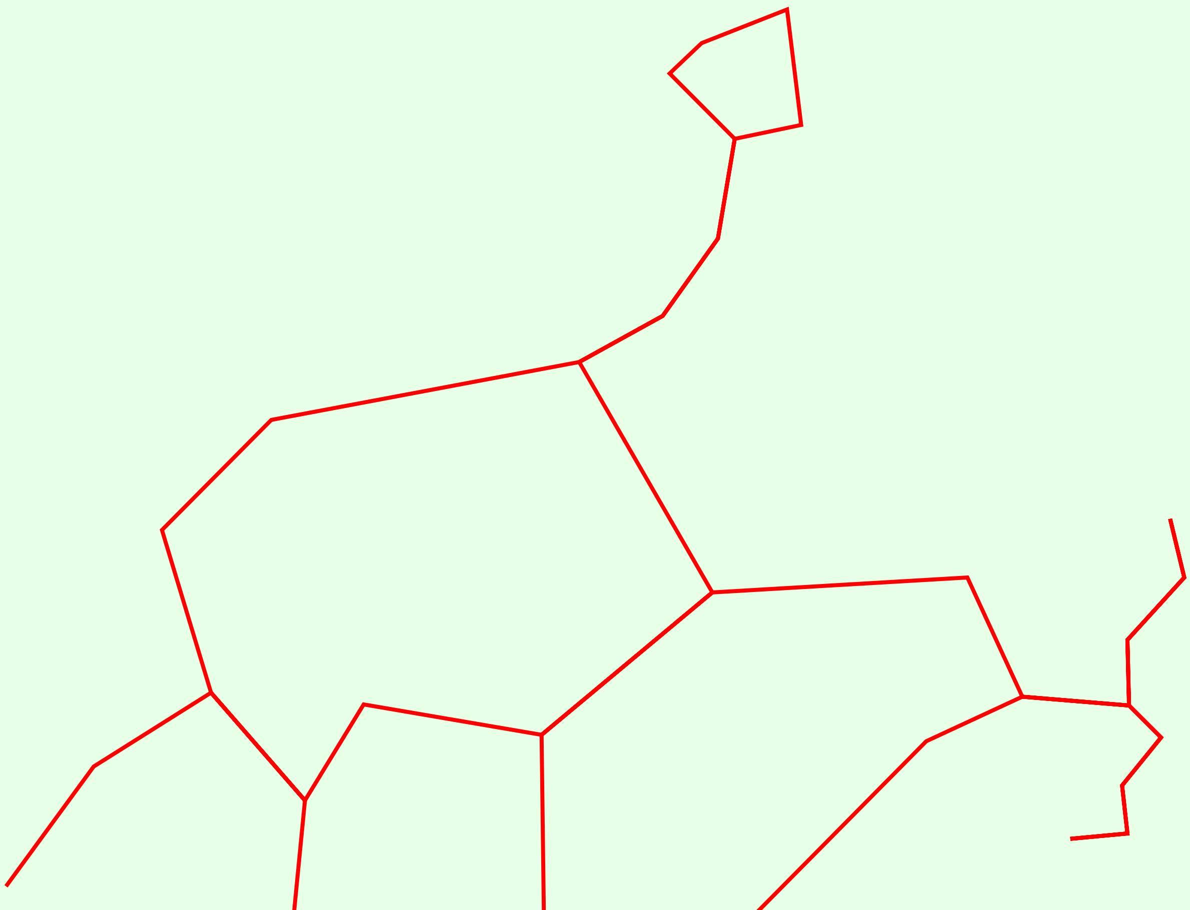
Henry Brighton and Chris Mellish - 2001

- ✓ *Review definitions of critical instances.*
- ✓ *Propose a new method.*
- ✓ *Perform an in-depth comparison of some of the best methods on 30 data sets.*
- ✓ Conclusion: *Methods work well for either homogeneous or non-homogeneous class structures, but NOT both.*
- ✓ *The best methods tuned to their class-structure can reduce the data sets by 80% without degradation in performance.*

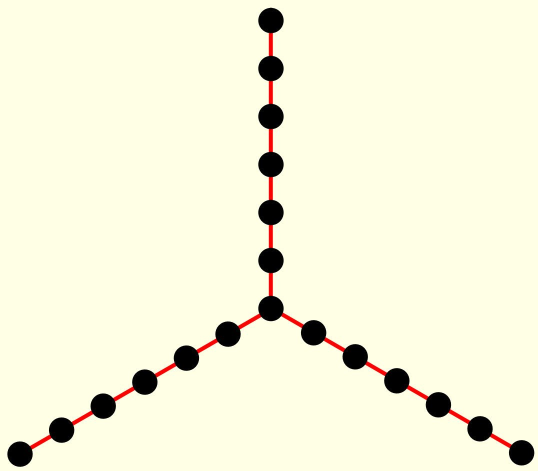
The Relative Neighborhood Graph

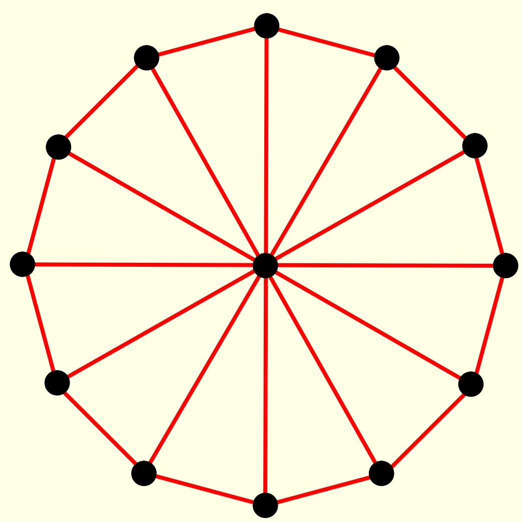
G. Toussaint, 1980





Two very different Relative Neighborhood Graphs



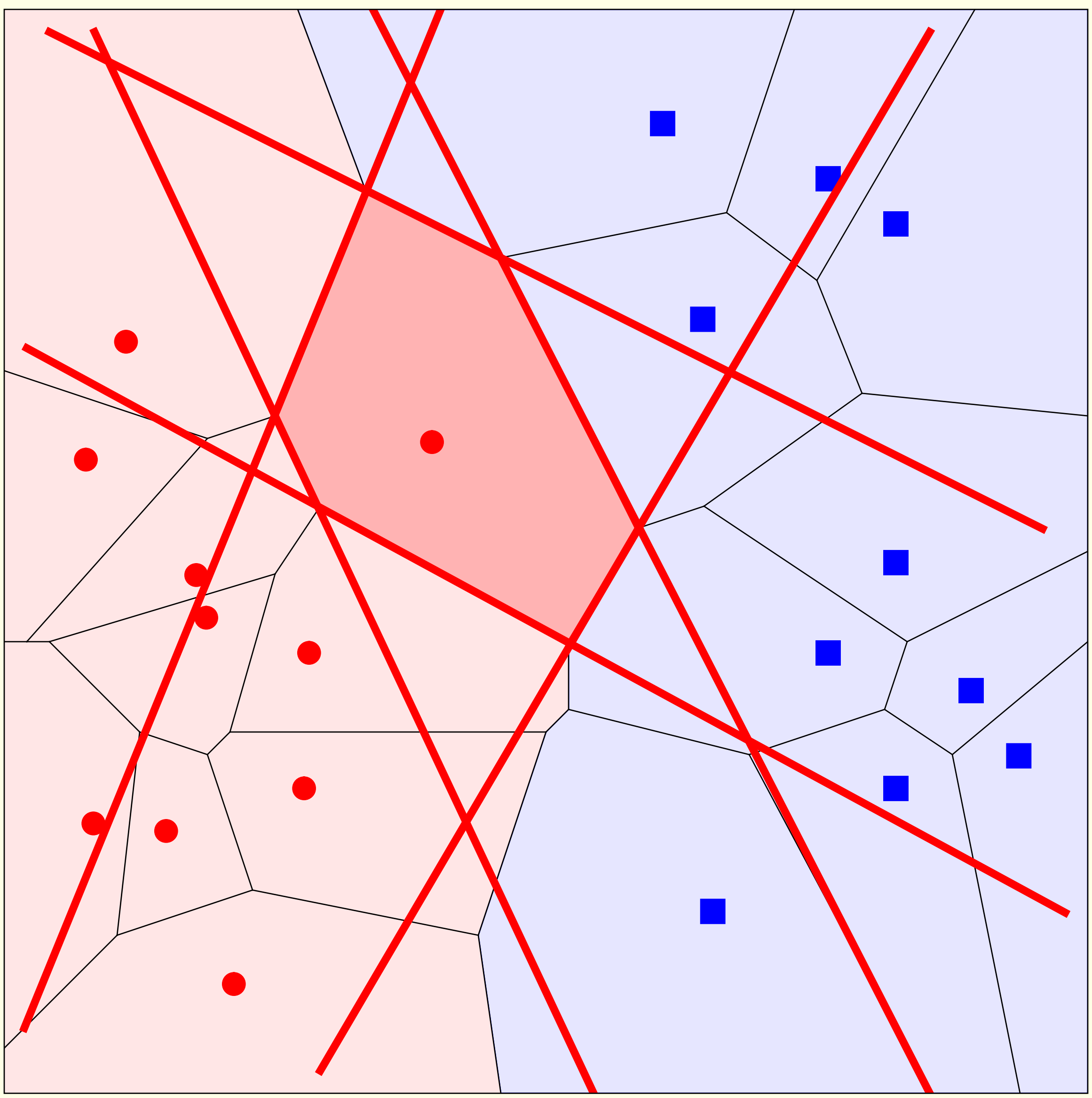


A Nearest Neighbor Pattern Classification Perceptron via explicit Voronoi diagrams

Owen Murphy - 1990

1. *Compute Voronoi diagram.*
2. *Use one McCulloch-Pitts neuron for each facet of each Voronoi cell in first layer.*

$O(n^2)$ neurons in $O(n^{\lceil (d+1)/2 \rceil})$ time & $O(n^{\lceil d/2 \rceil})$ space.

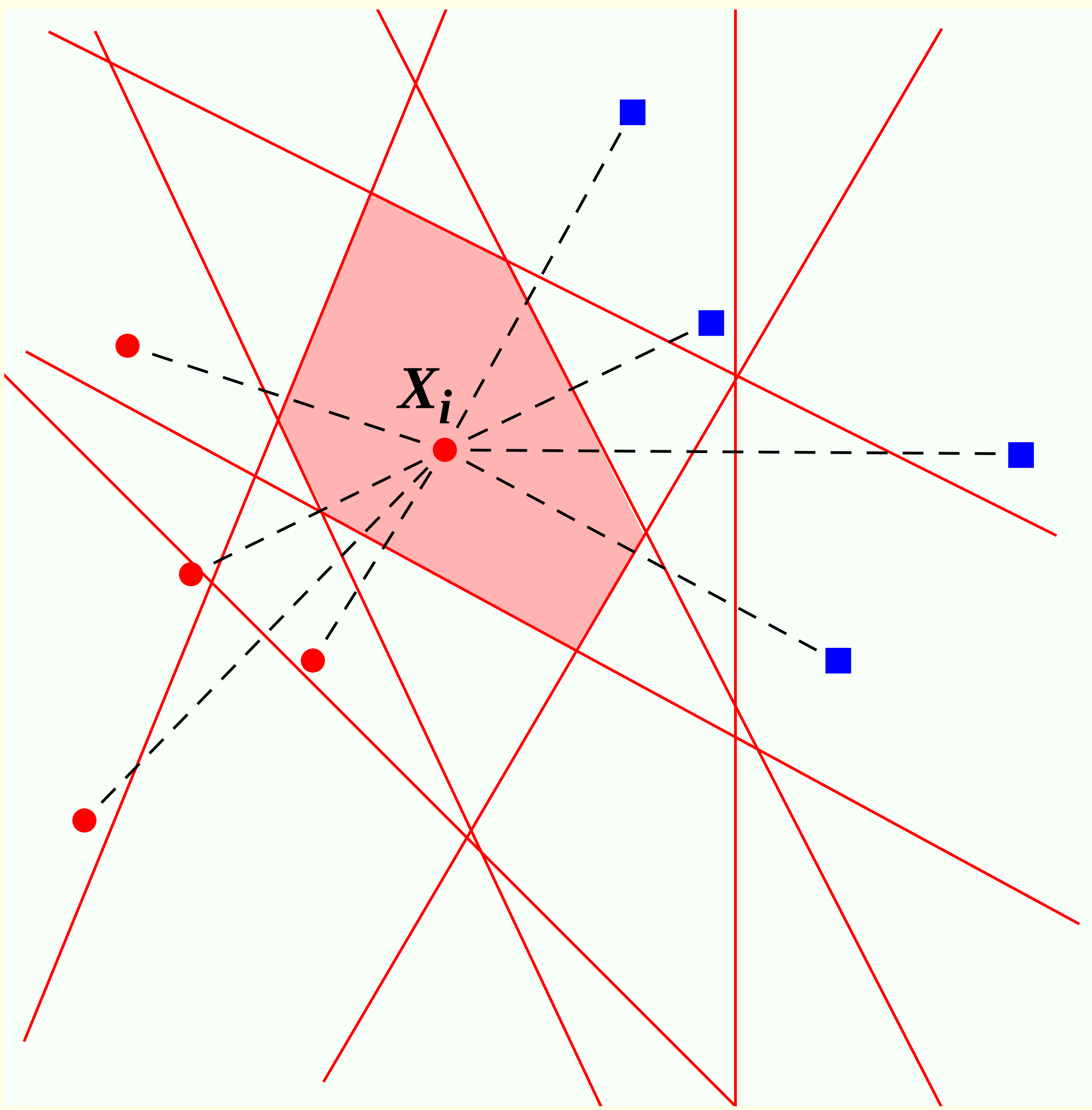


A Nearest Neighbor Pattern Classification
Perceptron via implicit Voronoi diagrams

Owen Murphy - 1990

1. *For each X_i Compute n-1 bisecting hyperplanes.*
2. *Use one McCulloch-Pitts neuron for each bisecting hyperplane in first layer.*

O(n^2) neurons in O(dn^2) time and O(dn) space.



Expected Size of Neural Networks for Nearest Neighbor Perceptrons

---

O. Murphy, B. Brooks and T. Kite - 1995

1. *Compute Voronoi diagram with fast expected time algorithms.*
2. *Use one McCulloch-Pitts neuron for each facet of each Voronoi cell in first layer.*

O(n) *expected neurons in* O(n) *expected time &* O(n) *expected space for fixed* d.

But:
1. *Hidden constant is large.*
2. *Worst case number of neurons is still* O(n^2).
3. *Worst case complexity of computing Voronoi diagram is exponential in* d.

Note: *They also rediscover Voronoi condensing proposed by* Toussaint and Poulsen in 1979.

Discarding Redundant Hyperplanes of Nearest Neighbor Perceptrons

C. Gentile and M. Aznaier - 2001

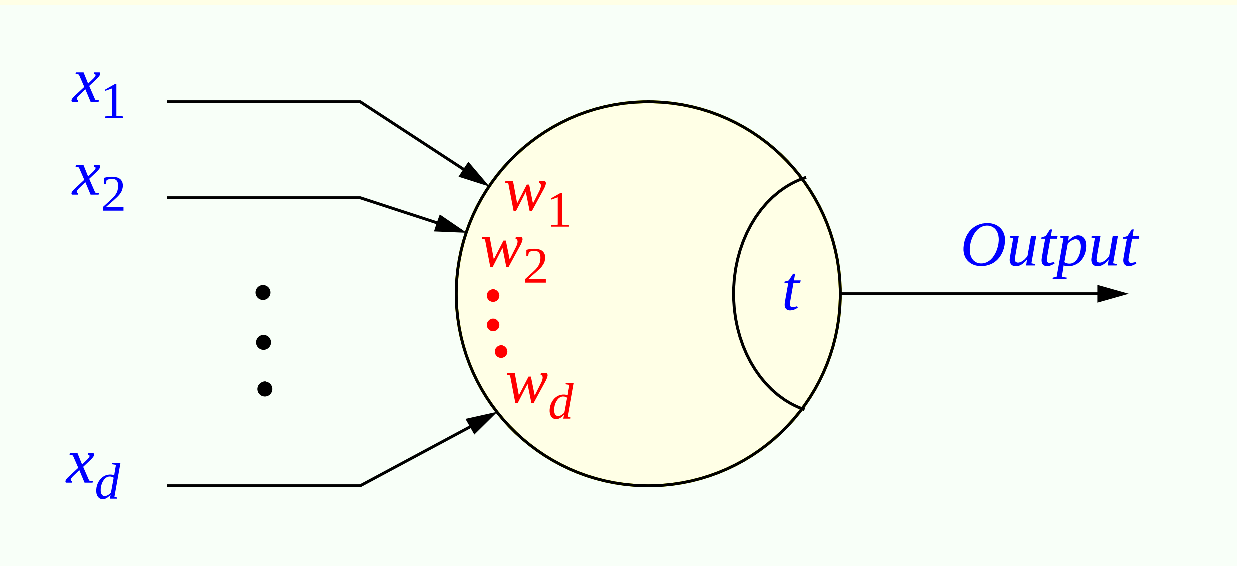
1. *Discards redundant hyperplanes. Computes Voronoi diagram.*
2. *Uses two layers instead of three, and fewer McCulloch-Pitts neurons than* Murphy et al.. (*largely duplicates work of* Murphy et al. *and does not reference them*)

But:

1. *Worst case number of neurons is still* O(n^2).
2. *Worst case complexity of computing Voronoi diagram is exponential in d.*

The One-Layer Two-Class Linear Neural Network

1. *Perceptron* - Rosenblatt - 1962
2. McCulloch-Pitts *neuron* - 1943
3. *Threshold Logic Unit* (TLU) - Dertouzos - 1965
4. *Linear discriminant function* - Fisher - 1936



if $\sum_{k=1}^{d} w_k x_k + w_{d+1} > 0$ *output* = 1

else *output* = 0

Solving Systems of Linear Inequalities via the Relaxation Method

S. Agmon - *Canadian J. of Mathematics* - 1954
T. Motzkin and I. Schoenberg - 1954

Given: training data of n d-dimensional vectors $\{\mathbf{X}\} = \{X_1, X_2, \ldots, X_i, X_{i+1}, \ldots, X_n\}$, *the weights of the perceptron can be determined by solving a system of linear inequalities.*

Class 1

$$\begin{cases} w_1 x_{11} + w_2 x_{12} + \ldots + w_d x_{1d} + w_{d+1} > 0 \\ w_1 x_{21} + w_2 x_{22} + \ldots + w_d x_{2d} + w_{d+1} > 0 \\ \vdots \\ w_1 x_{i1} + w_2 x_{i2} + \ldots + w_d x_{id} + w_{d+1} > 0 \end{cases}$$

Class 2

$$\begin{cases} w_1 x_{i+1,1} + w_2 x_{i+1,2} + \ldots + w_d x_{i+1,d} + w_{d+1} \leq 0 \\ \vdots \\ w_1 x_{n1} + w_2 x_{n2} + \ldots + w_d x_{nd} + w_{d+1} \leq 0 \end{cases}$$

Minimum-Distance Pattern Classification
Perceptron

Each class is represented by a prototype vector P_i.

Classify an unknown X *into class* C_i *if:*

$$d(X, P_i) \leq d(X, P_j) \text{ for all } j \neq i, \text{ or if:}$$

$$g_i(X) \equiv -d(X, P_i) > -d(X, P_j) \equiv g_j(X) \text{ for all } j \neq i.$$

$$g_i(X) = -(X - P_i) \bullet (X - P_i)$$

$$g_i(X) = -(X \bullet X - 2P_i \bullet X + P_i \bullet P_i)$$

$$g_i(X) = P_i \bullet X - (P_i \bullet P_i)/2$$

$$g_i(X) = \sum_{k=1}^{d} w_{ik} x_k + w_{d+1}$$

where $w_{ik} = p_{ik}$ *and* $w_{i,d+1} = -(P_i \bullet P_i)/2$.