Natural Discriminant Analysis Using Interactive Potts Models

Neural computation 2002, April

Jiann-Ming Wu
Jmwu@mail.ndhu.edu.tw
Department of Applied Mathematics, National Donghwa University

Outlines

- Discriminate analysis of paired data
- Generative models of predictors
- PottsNDA
  - Discriminate function
  - Learning network
- A mixed integer and linear programming
- Free energy approaches
  - Free energy function
  - Interactive Dynamics
- Incremental learning
- Numerical simulations and discussion
- Conclusions

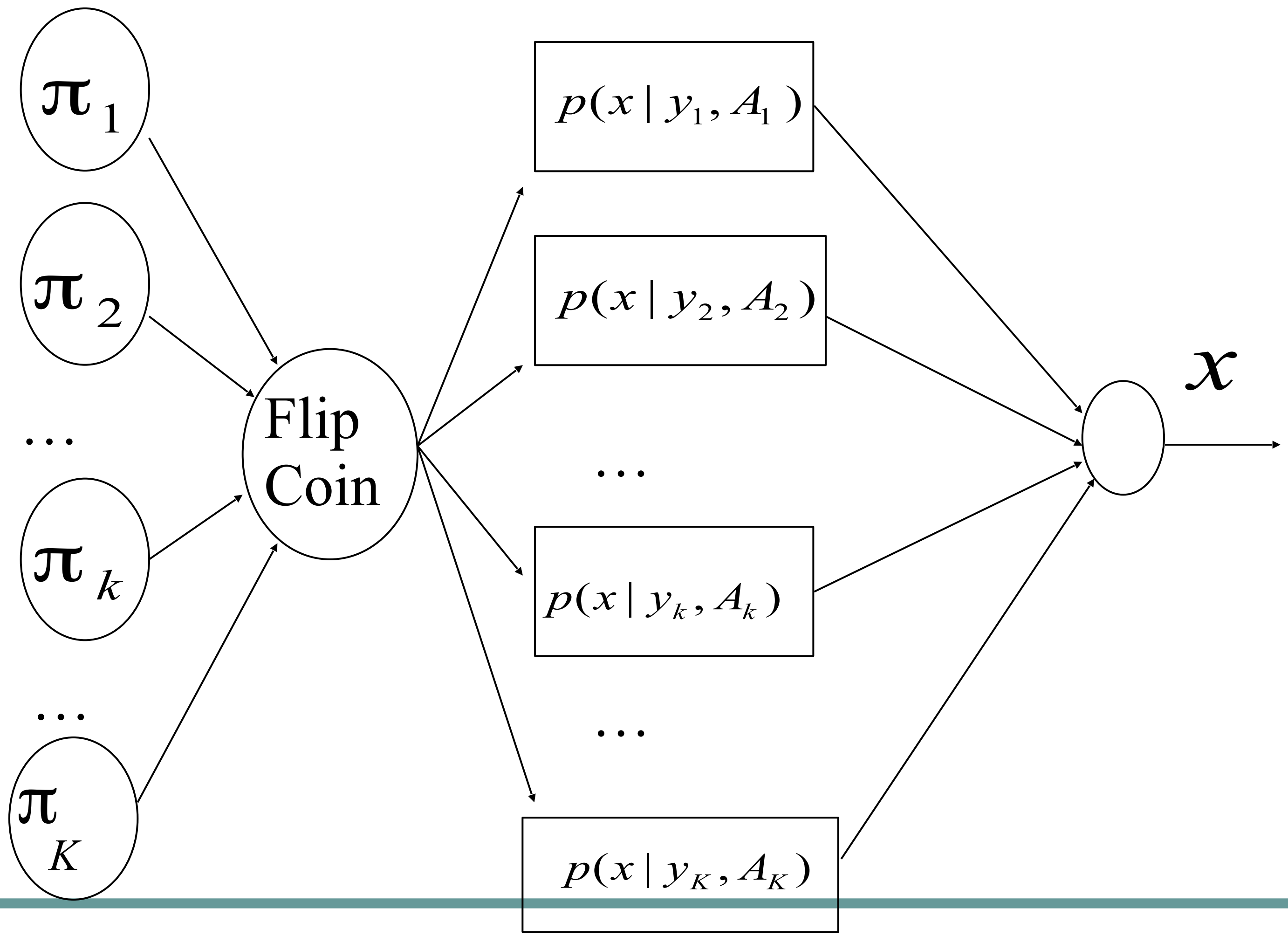
Paired data

$$\mathrm{D} = \{(\mathbf{x}_i, q_i)\}_i$$

$\mathbf{x}_i \in R^d$ denotes a predictor

q_i represents the category of $\mathbf{x}_i$

A generative model for predictors



Prior probabilities

$$\{\pi_m\}_m$$

Unitary condition

$$\sum_m \pi_m = 1$$

Generation of predictors:

According to prior probabilities, each time one of joined sub-models is selected and triggered to generate a predictor

Sub-models

- Multivariate Gaussians
- pdf

$$p_k(x) = \frac{1}{(2\pi)^{\frac{d}{2}}\sqrt{|A^{-1}|}} \exp\left(-\frac{1}{2}(x-y_k)'A(x-y_k)\right)$$

- A common weight matrix, A

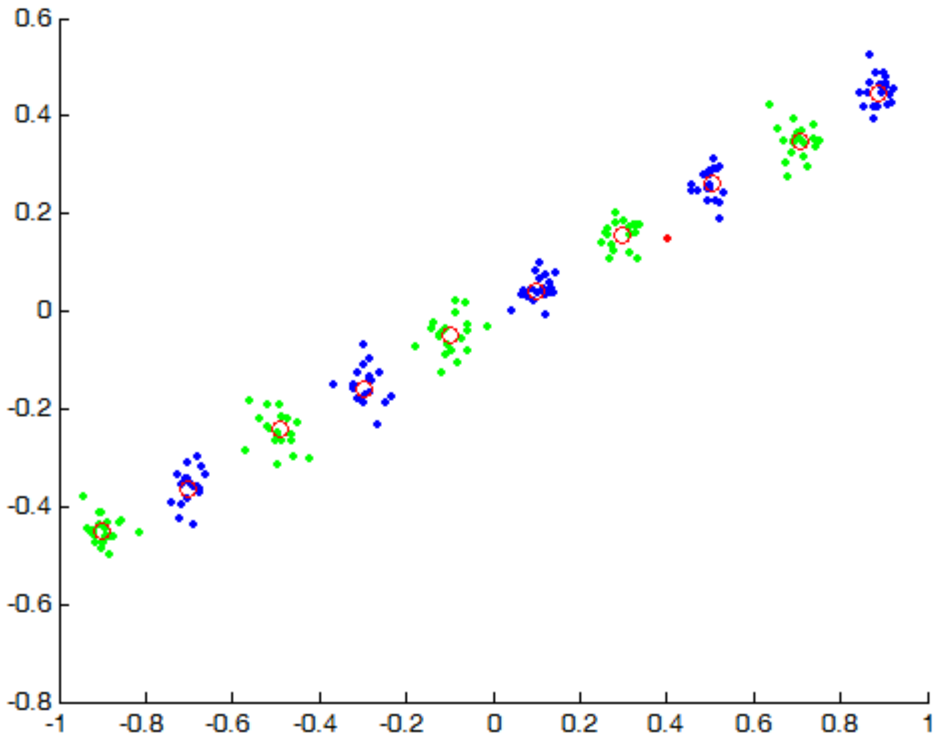
Gaussian mixtures

- Gaussian mixture assumption: given predictors are sampled from Gaussian mixtures
- pdf

$$p(\mathbf{x}) = \sum_k p_k(\mathbf{x})$$

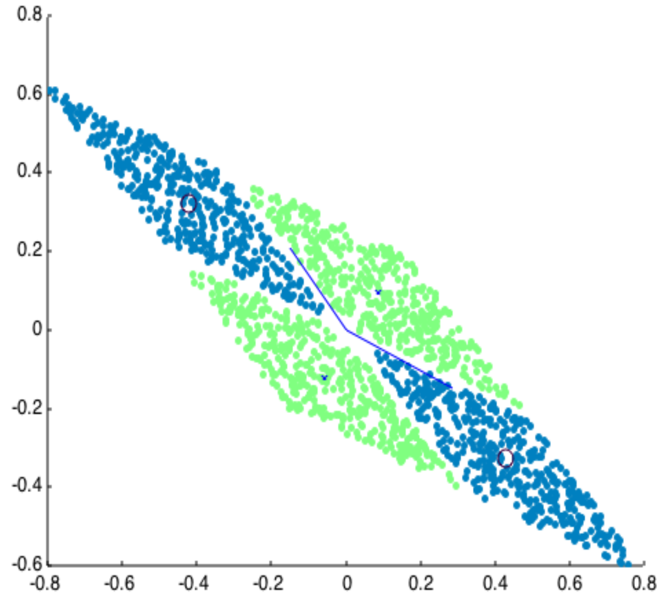
Examples: Gaussian Mixtures

- Linear local means



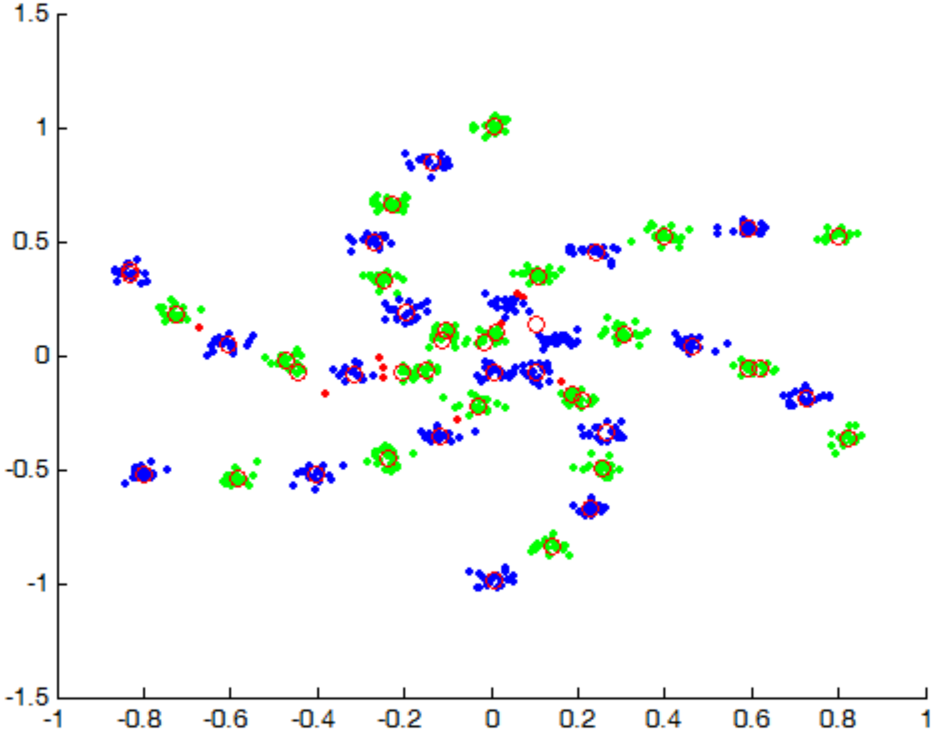
Examples: Gaussian mixture

- Four local means
- Non-overlapping distributions
- A common weight matrix for rotation



Examples: Gaussian mixtures

- Spiral data



Unitary vectors for category representations

- Example: two categories

$$q_i \in \{(1,0),(0,1)\}$$

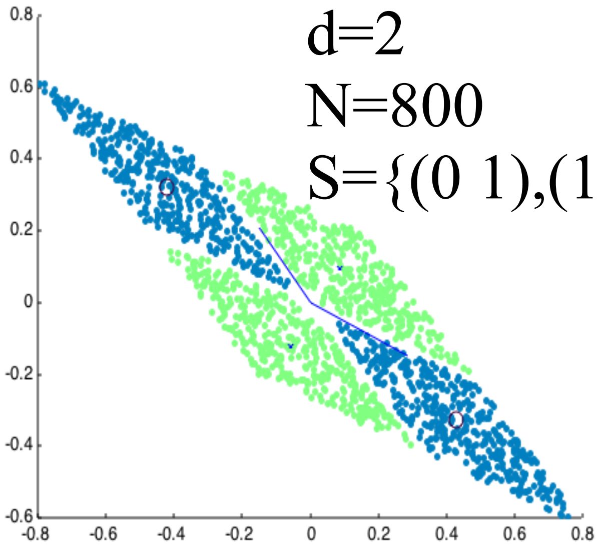
Unitary vectors for category representations

- Example: three categories

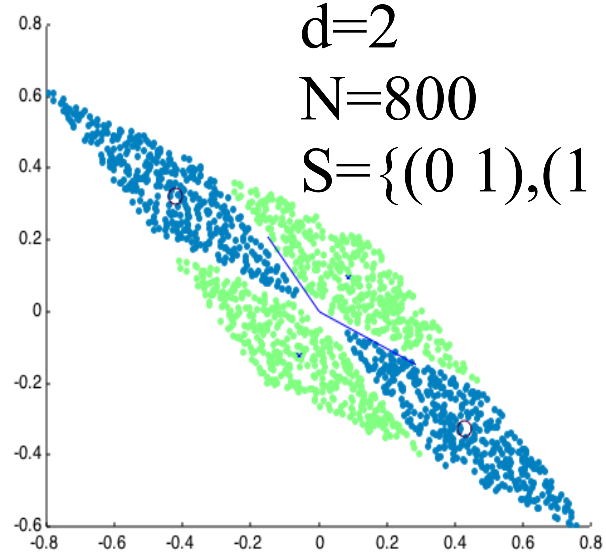
$$q_i \in \{(1,0,0),(0,1,0),(0,0,1)\}$$

Discriminate analysis of paired data

Training set
d=2
N=800
S={(0 1),(1 0)}



Testing set
d=2
N=800
S={(0 1),(1 0)}



Training set= $\{(x_i, q_i), 1 \leq i \leq N, x_i \in R^d, q_i \in S\}.$

Training set → PottsDA(θ) → q=g(x; θ) ← Testing set

q=g(x; θ) → Correct rate

Voronoi partition

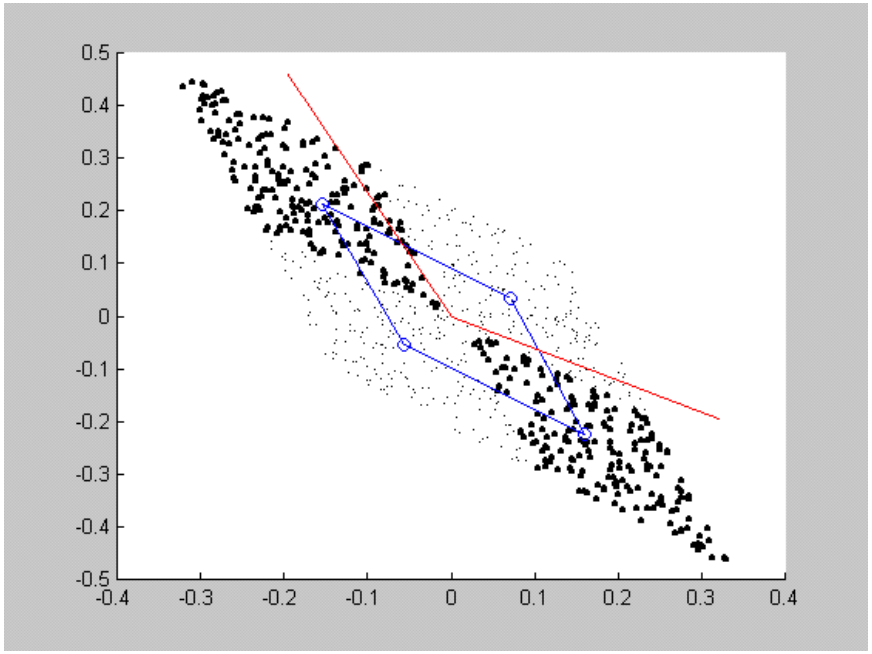
Manhalanobis distance

$$\|\mathbf{x}-\mathbf{y}\|_A = \sqrt{(\mathbf{x}-\mathbf{y})^T A(\mathbf{x}-\mathbf{y})}$$

Voronoi Partition defined by A and all $\mathbf{y}_\mathrm{i}$ in θ

$$\Omega_\mathrm{k} = \{x \mid k = \arg\min_j \|\boldsymbol{x} - \boldsymbol{y}_j\|_A\}$$

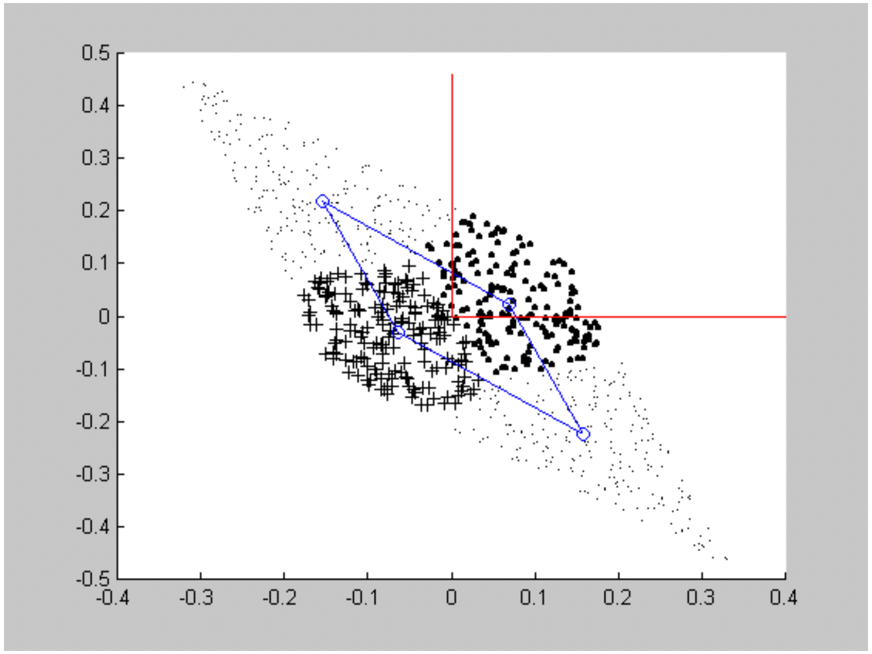
Partition based on Mahalanobis distances



Voronoi partition

Partition based on Euclidean distances

A=I



Memberships

- Unitary vectors for membership representations

$\mathbf{e}_k$ denotes a unitary vector with the kth bit one and others zeros

$\Xi_{\mathrm{K}} = \{\mathbf{e}_k\}_{k=1}^{K}$ denotes collection of possible memberships

Exclusive Memberships

$\boldsymbol{\delta}_i$ denotes the exclusive membership of $\mathbf{x}_\mathbf{i}$ to regions defined by θ

$$\boldsymbol{\delta}_i = F(\mathbf{x}_i; \theta) = \mathbf{e}_k \quad \text{if } \mathbf{x}_i \in \Omega_k$$

Category labels

- Let each region possess its own category label, denoted by $\boldsymbol{\xi}_m$
- $\boldsymbol{\xi}$ denotes collection of all category labels

Discriminating function

- θ and ξ define a discriminate function

$$g(\mathbf{x}_i;\boldsymbol{\theta},\boldsymbol{\xi})$$

$$= \sum_k \xi_k F(\mathbf{x}_i;\theta)\mathbf{e}_k$$

$$= \sum_k \xi_k \boldsymbol{\delta}_i^T \mathbf{e}_k$$

$$= \sum_k \sum_m \xi_k \delta_{im}$$

Discriminate function

$$g(x) = \xi_{k^*},$$

$$k^* = \arg\min_{k} \left\| x - y_k \right\|_A,$$

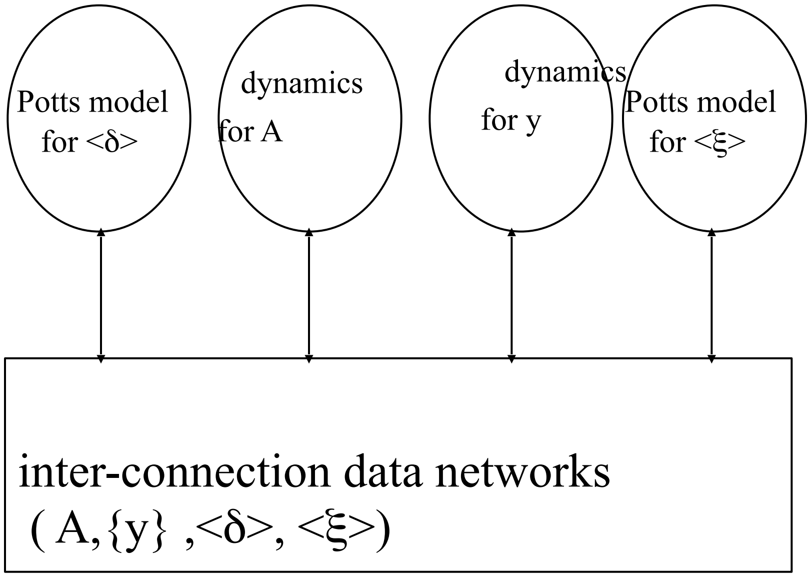
Discriminate functions

- Overlapping memberships

$$G_k^A(x) = \frac{\exp(-\beta(x-y_k)'A(x-y_k))}{\sum_j \exp(-\beta(x-y_j)'A(x-y_j))},$$

$$g(\mathbf{x}_i;\boldsymbol{\theta},\boldsymbol{\xi}) = \sum_k \sum_m \boldsymbol{\xi}_k \delta_{im} \quad \Rightarrow \quad g_\beta(\mathbf{x}_i;\boldsymbol{\theta},\boldsymbol{\xi}) = \sum_k \sum_m \boldsymbol{\xi}_k G_k^A(\mathbf{x}_i)$$

Learning Network of PottsDA



Fitting Gaussian mixtures

- Translate fitting a generative model to tasks of fitting joined individual sub-models

$$l = \sum_k l_k$$

Fitting a submodel

- Maximal likelihood

$$l_k = \log \prod_{x_i \in \Omega_k} p_k(x_i).$$

$$= \sum_{x_i \in \Omega_k} \log p_k(x_i)$$

$$= \sum_i \delta_{ik} \log p_k(x_i)$$

Fitting criteria

$$l = \sum_i \sum_k \delta_{ik} \log p_k(x_i)$$
$$= -\frac{1}{2} \sum_i \sum_k \delta_{ik} (x_i - y_k)' A (x_i - y_k)$$
$$- \frac{N}{2} \log \det(A^{-1}) - \frac{Nd}{2} \log(2\pi),$$

- *Setting det(A^{-1}) = - det(A)* and neglecting the last constant term

$$E_1 = \frac{1}{2} \sum_i \sum_k \delta_{ik} (x_i - y_k)' A (x_i - y_k) - \frac{N}{2} \log \det(A)$$

- Maximizing the function *l* is equivalent to minimizing the function E_1

Discriminating errors

$$E_2 = \frac{1}{2}\sum_i ||q_i - \sum_k \delta_{ik}\xi_k||^2$$

$$= \frac{1}{2}\sum_i ||q_i - \Lambda\delta_i||^2,$$

$$\Lambda = [\xi_1, \ldots, \xi_k, \ldots, \xi_K]$$

MINP: A mixed integer nonlinear programming

- Objectives

$$
\begin{aligned}
E(\delta, \xi, y, A) &= E_1 + cE_2 \\
&= \frac{1}{2} \sum_i \sum_k \delta_{ik} (x_i - y_k)' A (x_i - y_k) \\
&\quad - \frac{N}{2} \log \det(A) + \frac{c}{2} \sum_i ||q_i - \Lambda \delta_i||^2,
\end{aligned}
$$

Constraints

$$\delta_{ik} \in \{0, 1\}, \text{ for all } i, k$$

$$\sum_k \delta_{ik} = 1, \text{ for all } i$$

$$\xi_{km} \in \{0, 1\}, \text{ for all } k, m$$

$$\sum_m \delta_{km} = 1, \text{ for all } k,$$

MINP

- Mixed integer nonlinear programming
- Minimize E subject to unitary constraints of Potts variables

A mixed energy function for MINP

$$
\begin{aligned}
E(\delta, \xi, y, A) &= E_1 + cE_2 \\
&= \frac{1}{2}\sum_i \sum_k \delta_{ik}(x_i - y_k)' A (x_i - y_k) \\
&\quad - \frac{N}{2}\log\det(A) + \frac{c}{2}\sum_i ||q_i - \Lambda\delta_i||^2,
\end{aligned}
$$

Boltzmann assumption

- The system obeys the Boltzmann distribution

$$\Pr(\delta, \xi) \propto \exp(-\beta E(\delta, \xi)).$$

Physical annealing

- Physical annealing schedules the parameter *K* gradually from sufficiently low to high values
- At sufficiently large *K* value, the Boltzmann distribution will be dominated by optimal configurations.

$$\lim_{\beta \to \infty} \Pr(\delta^*, \xi^*) = 1,$$

where

$$E(\delta^*, \xi^*) = \min_{\delta, \xi} E(\delta, \xi)$$

A free energy

- A free energy measures the sum of the mean energy and the negative system entropy
- Independent assumption
  - All individuals are statistically independent
  - The mean energy can be approximated by substituting individual means to E
  - The system entropy equals the sum of individual entropies

A tractable free energy

$$\Psi(y, A, \langle\delta\rangle, \langle\xi\rangle, v, u)$$
$$= E(y, A, \langle\delta\rangle, \langle\xi\rangle) + \sum_i \sum_k \langle\delta_{ik}\rangle v_{ik} + \sum_k \sum_m \langle\xi_{km}\rangle u_{km}$$
$$-\frac{1}{\beta}\sum_i \ln\left(\sum_k \exp(\beta v_{ik})\right) - \frac{1}{\beta}\sum_k \ln\left(\sum_m \exp(\beta u_{km})\right)$$

where $\langle N\rangle$, $\langle Y\rangle$, u , and v denote $\{N_i\}$, $\{Y_k\}$, $\{u_{km}\}$, and $\{v_{ik}\}$, respectively, and u_i and v_k are auxiliary vectors.

Multiple sets of interactive dynamics

- A tractable free energy function is differentiable with respect to all of its dependent variables
- Setting zeros to derivatives of a tractable free energy function leads to multiple sets of interactive dynamics

A hybrid of mean field annealing and gradient descent methods

- The gradient descent method can not be directly applied to binary variables
- MFE for binary variables and GD for continuous variables
- $\{\delta_i\}$ and $\{\xi_k\}$ are associated with Potts neural variables or Potts spins in statistical mechanism

Mean field equations

$$\frac{\partial \Psi}{\partial \langle \delta_i \rangle} = 0, \quad \frac{\partial \Psi}{\partial v_i} = 0, \text{ for all } i$$

$$\frac{\partial \Psi}{\partial \langle \xi_k \rangle} = 0, \quad \frac{\partial \Psi}{\partial u_k} = 0, \text{ for all } k$$

Two sets of Mean field equations

$$v_i = -\frac{\partial E(y, A, \langle\delta\rangle, \langle\xi\rangle)}{\partial \langle\delta_i\rangle}$$

$$= -\frac{1}{2}(x_i - y_k)' A (x_i - y_k) - c\Lambda'(q_i - \Lambda\delta_i)$$

$$\langle\delta_i\rangle = \left[\frac{\exp(\beta v_{i1})}{\sum_h \exp(\beta v_{ih})}, \ldots, \frac{\exp(\beta v_{iK})}{\sum_h \exp(\beta v_{ih})}\right]'$$

$$u_k = -\frac{\partial E(y, A, \langle\delta\rangle, \langle\xi\rangle)}{\partial \langle\xi_k\rangle}$$

$$= c\sum_i \langle\delta_{ik}\rangle (q_i - \Lambda \langle\delta_i\rangle)$$

$$\langle\xi_k\rangle = \left[\frac{\exp(\beta u_{k1})}{\sum_m \exp(\beta u_{km})}, \ldots, \frac{\exp(\beta u_{kM})}{\sum_m \exp(\beta u_{km})}\right]'$$

Updating rule of weight matrix A

$$\triangle A_{mn} \propto -\frac{\partial \Psi}{\partial A_{mn}}$$

$$= -\frac{\partial E}{\partial A_{mn}}$$

$$= -\frac{1}{2}\sum_i \sum_k \langle \delta_{ik} \rangle (x_{im} - y_{km})(x_{in} - y_{kn}) + \frac{N}{2}[(A')^{-1}]_{mn}$$

When all $\triangle A_{mn} = 0$, we have

$$A = (W^{-1})',$$

$$W_{mn} = \frac{1}{N}\sum_i \sum_k \langle \delta_{ik} \rangle (x_{im} - y_{km})(x_{in} - y_{kn}).$$

Update rule of local means

- Gradient

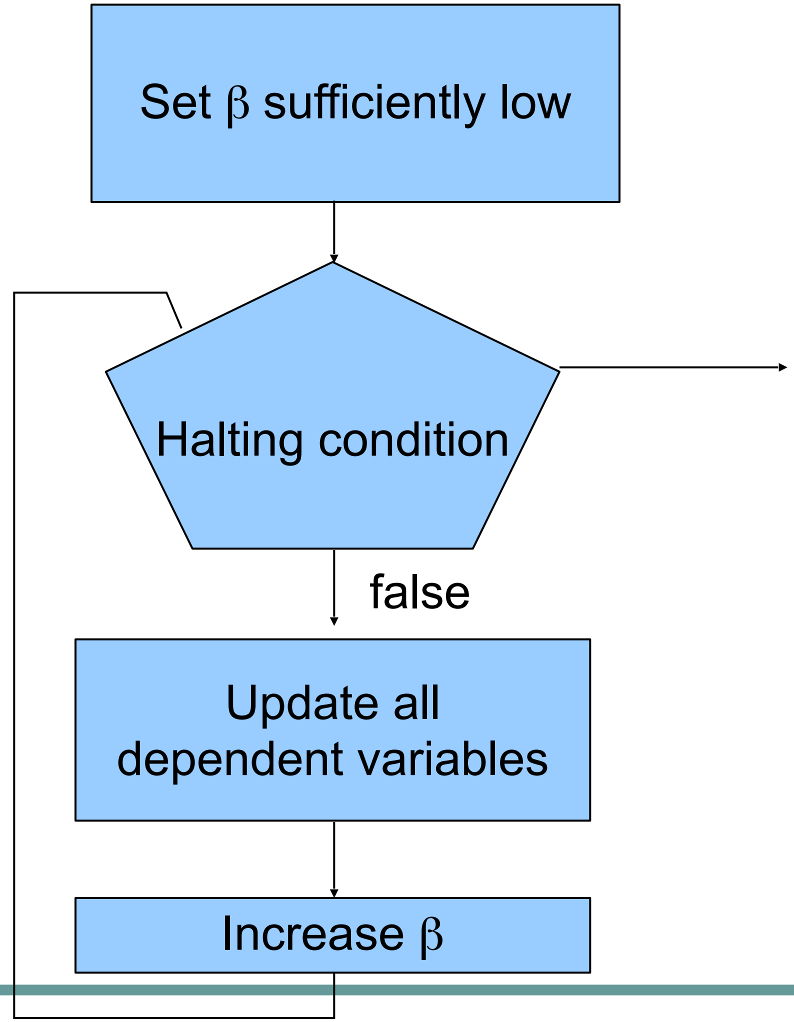
$$\Delta y_k \propto -\frac{\partial \Psi}{\partial y_k}$$

$$= \frac{1}{2} \sum_i \langle \delta_{ik} \rangle (A + A')(x_i - y_k)$$

- Again when *Ay = 0* , we have

$$y_k = \frac{\sum_i \langle \delta_{ik} \rangle x_i}{\sum_i \langle \delta_{ik} \rangle}$$

Annealing


Set β sufficiently low
Halting condition
false
Update all
dependent variables
Increase β

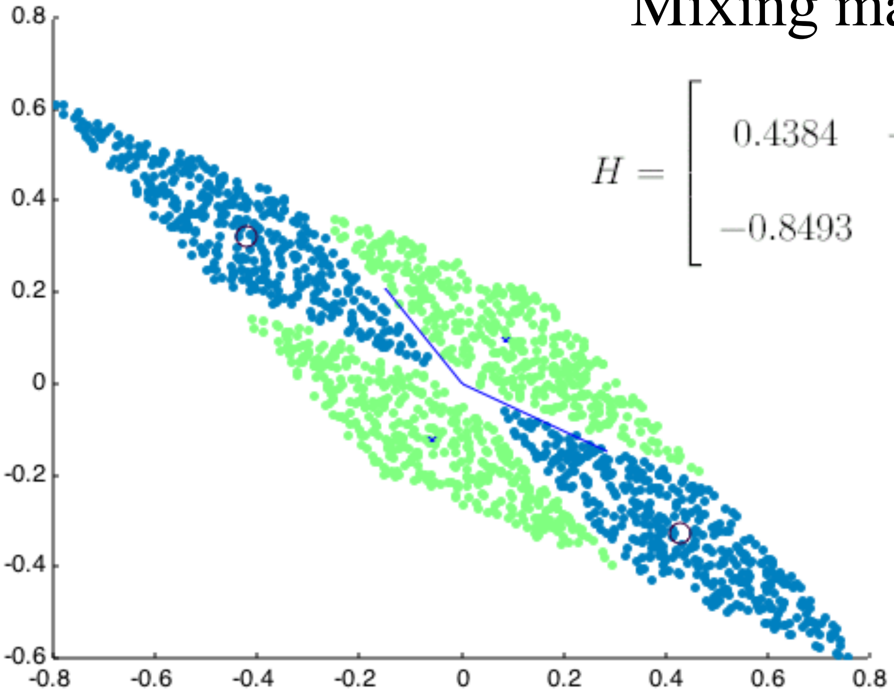
An annealing process for learning PottsDA

1. Set a sufficiently low β value, each kernel y_k near the mean of all predictors and each $\langle\delta_{ik}\rangle$ near $\frac{1}{K}$ and $\langle\xi_{km}\rangle$ near $\frac{1}{M}$.
2. Iteratively update all $\langle\delta_{ik}\rangle$ and v_{ik} by equations 3.4 and 3.5, respectively, to a stationary point.
3. Iteratively update each $\langle\xi_{km}\rangle$ and u_{km} by equations 3.6 and 3.7, respectively, to a stationary point.
4. Update each $\mathbf{y}_i$ by equation 3.12.
5. Update $\mathbf{A}$ by equations 3.9 and 3.10.
6. If $\sum_{ik}\langle\delta_{ik}\rangle^2$ and $\sum_{km}\langle\xi_{km}\rangle^2$ are larger than a prior threshold, then halt; otherwise increase β by an annealing schedule and go to step 2.

Numerical Simulations

- Performance evaluation:
  1. PottsDA
  2. Radial basis function(RBF) method
  3. Support vector machine(SVM) method (Vapnik 1995)

Artificial data: Example 1



Mixing matrix

$$H = \begin{bmatrix} 0.4384 & -0.8988 \\ -0.8493 & 0.5279 \end{bmatrix}$$

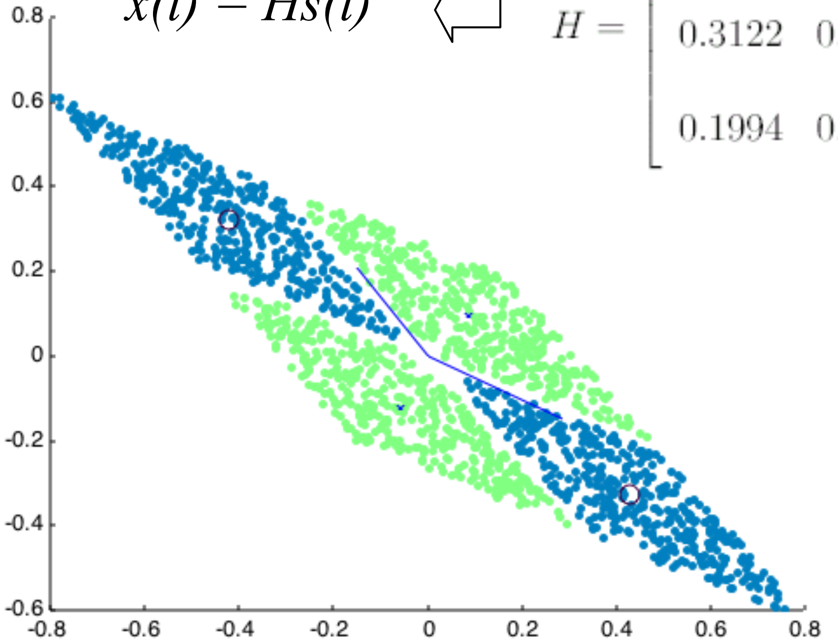
PottsDA: two columns of the inverse of A, four kernels, and category labels.

Table 1 The performance of the three methods for the first example

	RBF(4)	RBF(8)	RBF(12)	RBF(24)	SVM	PottsDA(4)
Training	14.1%	12.0%	8.6%	3.9%	13.2%	0%
Testing	13.0%	12.1%	8.3%	4.5%	14.3%	0%

Artificial data: Example 2

$x(t) = Hs(t)$ ⇦ $H = \begin{bmatrix} 0.9288 & 0.2803 & 0.3770 \\ 0.3122 & 0.9366 & 0.2572 \\ 0.1994 & 0.2098 & 0.8897 \end{bmatrix}$ ⇦ $s(t) = [s^1(t)\ s^2(t)\ s^3(t)]'$

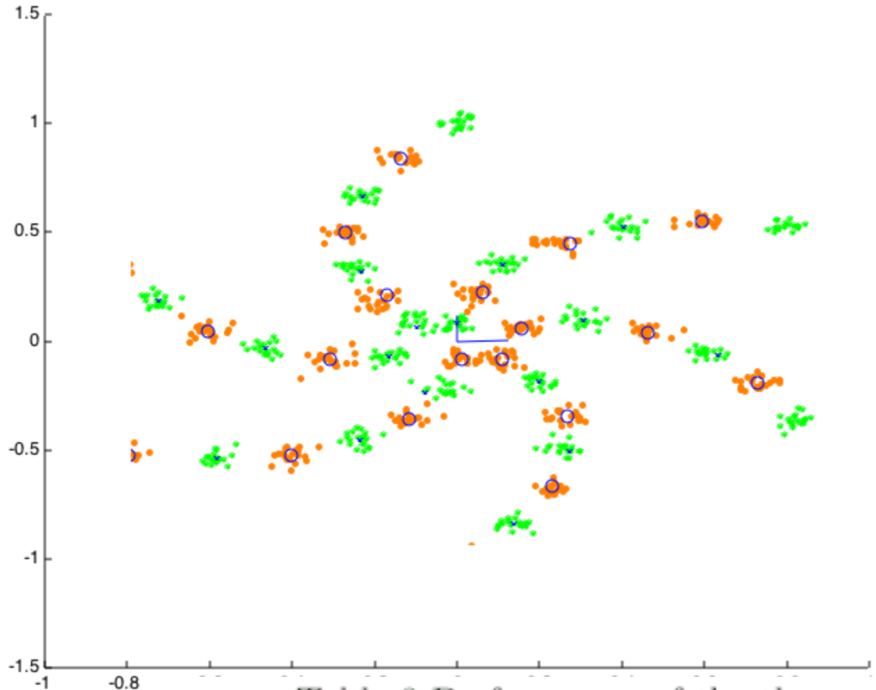


- $s^1(t)$ and $s^2(t)$, are uniform random variables within $[-0.5, 0.5]$
- $s^3(t)$ is a Gaussian noise of $N(0,\sqrt{2})$
- Discriminate rule: $sign(s^1(t)) * sign(s^2(t))$ the third source as a noise for prediction.

Table 2 Performance of the three methods for the second example

	RBF(4)	RBF(8)	RBF(12)	RBF(24)	SVM	PottsDA(4)
Training	45.3%	31.2%	22.6%	10.9%	3.2%	0.2%
Test	44%	31.1%	24.6%	13.9%	5.9%	0%

Artificial data: Example 3



PottsDA:

Two columns of the inverse of A

40 local means,

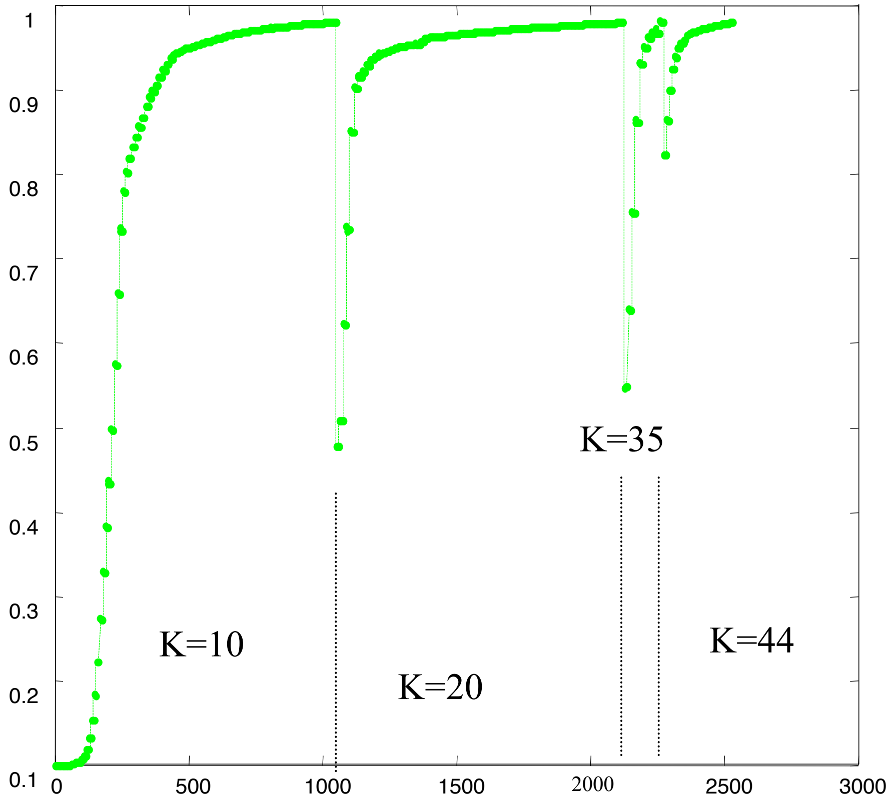
and category labels

Table 3 Performance of the three methods for the third example

	RBF(40)	RBF(50)	RBF(60)	RBF(80)	SVM	PottsDA(40)
Training	14.6%	10.4%	7.8%	3.3%	45.5%	0.8%
Test	15.7%	12.3%	9.5%	4.1%	45.6%	0.4%

Incremental learning for Example 3

- $\sum \langle \delta_{ik} \rangle^2$



.

The horizontal coordinate is the time index for varying the beta value

Discriminate analysis of Wisconsin Breast Cancer Database

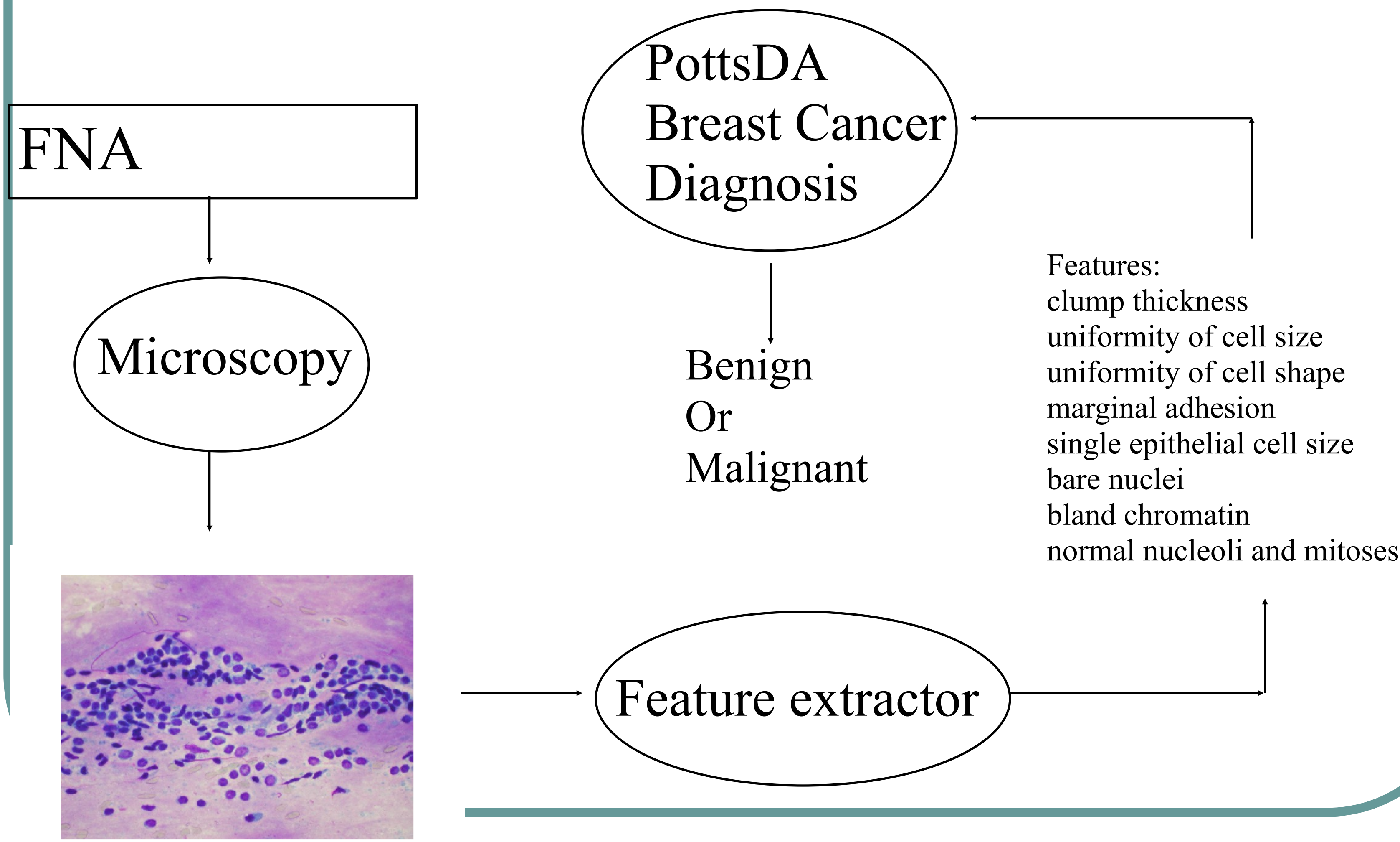
- Walberg and Mangasarian 1990
- 699 instances,

  each containing 9 features for predicting one of

  benign and malignant categories.
- 458 instances in the benign category

  241 instances in the malignant category

Wisconsin Breast Cancer Database



Simulation Results

- Walberg and Mangasarian 1990
  error rate for testing > 6%
- 683 instances of the database by Malini Lamego(2001)

	PottsDA(42)	Neural Net with algebraic loops
Train(483)	1.4%	2.3%
Test(200)	1%	4.5%

- For the 219-case test set, the RBF method with 80 kernels and the SVM method result in error rates, 4.17% and 4.63%, for testing.

Conclusions

- PottsDA
  - A discriminant network
  - An annealed learning approach
- Translate discriminate analysis to minimization of fitting criteria and approximating errors
- PottsDA learning is realized by a hybrid of mean field annealing and gradient descent methods
- Incremental learning for PottsDA is effective for determining the optimal model size.
- Encouraging learning results of PottsDA discriminate analysis.

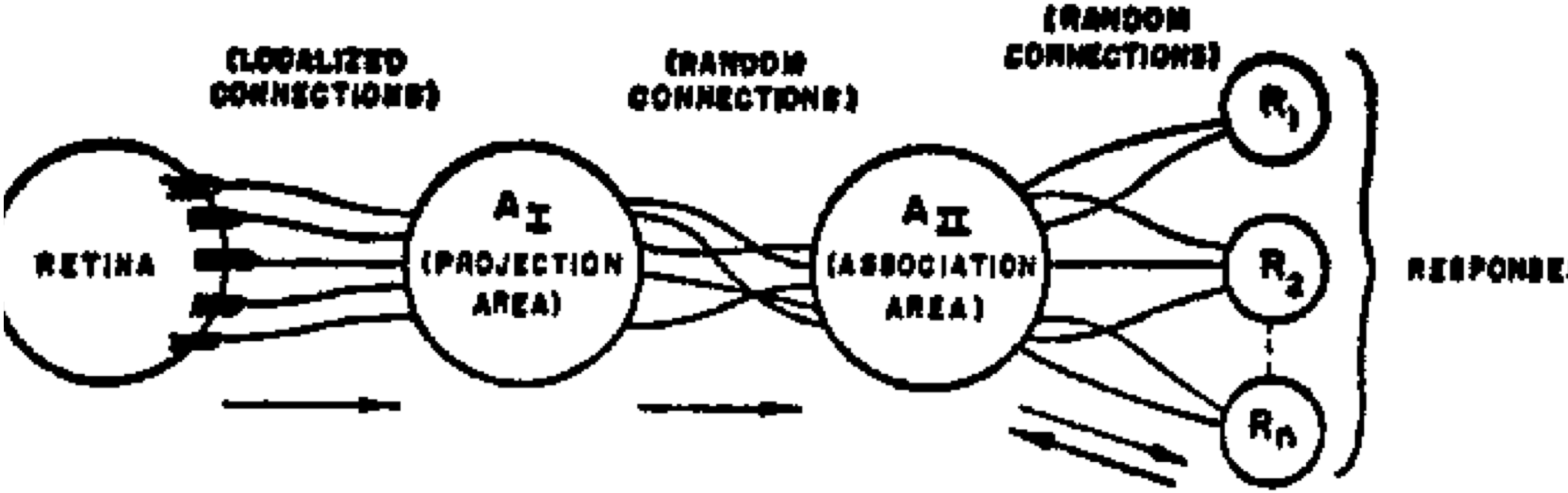


FIG. 1. Organization of a perceptron.