

# Recurrent Bayesian Reasoning in Probabilistic Neural Networks

*Jiří Grim and Jan Hora*

**Institute of Information Theory and Automation  
Academy of Sciences of the Czech Republic, Prague**

**Department of Pattern Recognition**

<http://www.utia.cas.cz/RO>

**Conference ICANN'07, Porto, September 9-13, 2007**

# Outline

- 1 Framework of PNN: Statistical Pattern Recognition
  - Distribution Mixtures with Product Components
- 2 PNN Approach: Structural Mixture Model
  - Distribution Mixtures with Structural Parameters
  - EM Algorithm For Structural Mixtures
  - Product Mixture Components as Probabilistic Neurons
- 3 Recurrent Bayesian Reasoning
  - Recurrent Use of Bayes Formula
  - Recurrent Modification of Input Pattern
- 4 EXAMPLE
  - NIST SD19 Database of Hand-Written Numerals
  - Recognition of Numerals From the NIST SD19 Database
  - Basic Features of Recurrent Bayesian Reasoning
- 5 Concluding Remarks

# Statistical Approach to Pattern Recognition

$\mathbf{x} = (x_1, \dots, x_N) \in \mathcal{X}$ : N-dimensional binary data vectors

$\Omega = \{\omega_1, \omega_2, \dots, \omega_J\}$ : finite number of classes

$P(\mathbf{x}|\omega)p(\omega)$ ,  $\omega \in \Omega$ : conditional distributions of classes

**Bayes formula:** to classify any given  $\mathbf{x} \in \mathcal{X}$  uniquely

$$p(\omega|\mathbf{x}) = \frac{P(\mathbf{x}|\omega)p(\omega)}{P(\mathbf{x})}, \quad P(\mathbf{x}) = \sum_{\omega \in \Omega} P(\mathbf{x}|\omega)p(\omega)$$

**PNN solution:** approximation of  $P(\mathbf{x}|\omega)$  by mixtures of product components

$$P(\mathbf{x}|\omega) = \sum_{m \in \mathcal{M}_\omega} w_m F(\mathbf{x}|m), \quad F(\mathbf{x}|m) = \prod_{n \in \mathcal{N}} f_n(x_n|m), \quad \sum_{m \in \mathcal{M}_\omega} w_m = 1.$$

$$P(\mathbf{x}) = \sum_{m \in \mathcal{M}} f(m)F(\mathbf{x}|m), \quad f(m) = p(\omega)w_m, \quad \mathcal{M} = \bigcup_{\omega \in \Omega} \mathcal{M}_\omega$$

**PNN output layer:**  $p(\omega|\mathbf{x}) = \sum_{m \in \mathcal{M}_\omega} f(m|\mathbf{x}), \quad f(m|\mathbf{x}) = \frac{F(\mathbf{x}|m)f(m)}{\sum_{j \in \mathcal{M}} F(\mathbf{x}|j)f(j)}$

**PNN hidden layer:**  $y_m(\mathbf{x}) = T_m(\mathbf{x}) = \log(f(m|\mathbf{x}))$

# Structural Mixture Model (Grim et al. 1986, 1999, 2002)

**STRUCTURAL MODEL:** to avoid complete interconnection property

$$F(\mathbf{x}|m) = \prod_{n \in \mathcal{N}} f_n(x_n|m)^{\phi_{mn}} f_n(x_n|0)^{1-\phi_{mn}}, \quad \phi_{mn} \in \{0, 1\}$$

$\phi_{mn} = 0$ : distribution  $f_n(x_n|m)$  is replaced by a fixed “background”  $f_n(x_n|0)$

$$P(\mathbf{x}|\omega) = \sum_{m \in \mathcal{M}_\omega} F(\mathbf{x}|m) f(m) = \sum_{m \in \mathcal{M}_\omega} F(\mathbf{x}|0) G(\mathbf{x}|m, \phi_m) f(m)$$

$$F(\mathbf{x}|0) = \prod_{n \in \mathcal{N}} f_n(x_n|0), \quad G(\mathbf{x}|m, \phi_m) = \prod_{n \in \mathcal{N}} \left[ \frac{f_n(x_n|m)}{f_n(x_n|0)} \right]^{\phi_{mn}}$$

$G(\mathbf{x}|m, \phi_m) \approx$  depends on a subset of variables specified by  $\phi_{mn} = 1$

**“background distribution”  $F(\mathbf{x}|0)$  cancels in the Bayes formula:**

$$p(\omega|\mathbf{x}) = \frac{P(\mathbf{x}|\omega)p(\omega)}{P(\mathbf{x})} = \frac{\sum_{m \in \mathcal{M}_\omega} G(\mathbf{x}|m, \phi_m) f(m)}{\sum_{j \in \mathcal{M}} G(\mathbf{x}|j, \phi_j) f(j)} \approx \sum_{m \in \mathcal{M}_\omega} G(\mathbf{x}|m, \phi_m) f(m)$$

**Remark.** Unlike standard subspace approaches the structural mixture model enables statistically correct Bayesian decision-making.

# Structural Modification of EM Algorithm

**STRUCTURAL OPTIMIZATION:** can be included into EM algorithm

$$L = \frac{1}{|\mathcal{S}_\omega|} \sum_{\mathbf{x} \in \mathcal{S}_\omega} \log \left[ \sum_{m \in \mathcal{M}_\omega} F(\mathbf{x}|0) G(\mathbf{x}|m, \phi_m) w_m \right], \quad F(\mathbf{x}|0) = \prod_{n \in \mathcal{N}} f_n(x_n|0)$$

**EM Algorithm:** ( $m \in \mathcal{M}_\omega, n \in \mathcal{N}, \mathbf{x} \in \mathcal{S}_\omega$ )

$$q(m|\mathbf{x}) = q(m|\mathbf{x}, \omega) = \frac{G(\mathbf{x}|m, \phi_m) w_m}{\sum_{j \in \mathcal{M}_\omega} G(\mathbf{x}|j, \phi_j) w_j},$$

$$w'_m = \frac{1}{|\mathcal{S}_\omega|} \sum_{\mathbf{x} \in \mathcal{S}_\omega} q(m|\mathbf{x}), \quad \theta'_{mn} = \frac{1}{\sum_{\mathbf{x} \in \mathcal{S}_\omega} q(m|\mathbf{x})} \sum_{\mathbf{x} \in \mathcal{S}_\omega} x_n q(m|\mathbf{x})$$

**structural criterion:** (Kullback-Leibler  $\mathbb{I}$ -divergence)

$$\gamma'_{mn} = w'_m \left[ \theta'_{mn} \log \frac{\theta'_{mn}}{\theta_{0n}} + (1 - \theta'_{mn}) \log \frac{(1 - \theta'_{mn})}{(1 - \theta_{0n})} \right]$$

**structural parameter optimization:**  $\phi'_{mn} = 1$  for the  $r$  highest values  $\gamma'_{mn}$

**Remark.** The “structural” EM algorithm converges monotonically.

# Product Mixture Components as Probabilistic Neuron

**probabilistic neuron:**

$$y_m = \log f(m|\mathbf{x}) = \log f(m) + \sum_{n \in \mathcal{N}} \phi_{mn} \log \frac{f_n(x_n|m)}{f_n(x_n|0)} - \log \left[ \sum_{j \in \mathcal{M}} G(\mathbf{x}|j, \phi_j) f(j) \right]$$

$f(m|\mathbf{x}) \approx$  probability of “spike” given the input pattern  $\mathbf{x}$

$f(m) \approx$  spontaneous activity of the  $m$ -th neuron

$\log \frac{f_n(x_n|m)}{f_n(x_n|0)} \approx$  contribution of the input  $x_n$  to the activation of  $m$ -th neuron

$\log \left[ \sum_{j \in \mathcal{M}} G(\mathbf{x}|j, \phi_j) f(j) \right] \approx$  common “norming” term (lateral inhibition)

**”synaptical weight”:**  $\log \frac{f_n(x_n|m)}{f_n(x_n|0)} = \log \frac{f_n(x_n|m)}{P_n(x_n)} = \log \frac{f(m|x_n)}{f(m)}$

Hebb's postulate of learning (Hebb, 1949)

“When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic changes take place in one or both cells such that A's efficiency as one of the cells firing B, is increased.”

# Recurrent Use of Bayes Formula

$$\begin{aligned} \sum_{m \in \mathcal{M}} f(m) F(\mathbf{x}|m) &\approx \text{implicit "descriptive" decision problem} \\ f(m) F(\mathbf{x}|m) &\approx \text{"elementary" properties or situations} \\ f(m|\mathbf{x}) &\approx \text{conditional probabilities of situations given } \mathbf{x} \in \mathcal{X} \end{aligned}$$

**RECURRENT BAYES FORMULA:**  $t = 0, 1, 2, \dots$

$$f^{(t+1)}(m|\mathbf{x}) = \frac{G(\mathbf{x}|m, \phi_m) f^{(t)}(m|\mathbf{x})}{\sum_{j \in \mathcal{M}} G(\mathbf{x}|j, \phi_j) f^{(t)}(j|\mathbf{x})}, \quad f^{(0)}(m|\mathbf{x}) = f(m), \quad m \in \mathcal{M}$$

Convergence (Grim & Vejvalková, 1999)

Recurrent Bayes formula converges independently of the initial values  $f(m)$  to the limit weights  $f^*(m|\mathbf{x}) = \delta(m, m_0)$ ,  $m_0 = \arg \max_m \{G(\mathbf{x}|m, \phi_m)\}$ .

$$\mathcal{L}^{(t)} = \log \left[ \sum_{m \in \mathcal{M}} F(\mathbf{x}|0) G(\mathbf{x}|m, \phi_m) f^{(t)}(m) \right], \quad f^{(t)}(m) = f^{(t)}(m|\mathbf{x})$$

**Remark.** The recurrent computation of the conditional weights  $f^{(t)}(m|\mathbf{x})$  resembles natural process of cognition as iteratively improving understanding of input information.

# Recurrent Modification of Input Pattern

$$\mathcal{L}(\mathbf{x}) = \log P(\mathbf{x}) = \log \left[ \sum_{m \in \mathcal{M}} F(\mathbf{x}|0) G(\mathbf{x}|m, \phi_m) f(m) \right]$$

$\mathcal{L}(\mathbf{x})$  can be maximized as a function of  $\mathbf{x}$  by means of EM algorithm:

$$Q_n^{(t)} = \log \frac{\theta_{0n}}{1 - \theta_{0n}} + \sum_{m \in \mathcal{M}} \phi_{mn} f^{(t)}(m|\mathbf{x}) \log \frac{\theta_{mn}(1 - \theta_{0n})}{\theta_{0n}(1 - \theta_{mn})}$$

$$x_n^{(t+1)} = \begin{cases} 1, & Q_n^{(t)} \geq 0, \\ 0, & Q_n^{(t)} < 0, \end{cases}, \quad n \in \mathcal{N}, \quad t = 0, 1, 2, \dots$$

## Convergence property (Grim et al. 1998)

The sequence of iteratively modified input patterns  $\mathbf{x}^{(t)}$ ,  $t = 0, 1, 2, \dots$  converges to a local maximum or saddle point of  $P(\mathbf{x})$ , the corresponding sequence of log-likelihood values  $\mathcal{L}(\mathbf{x}^{(t)})$  is nondecreasing.

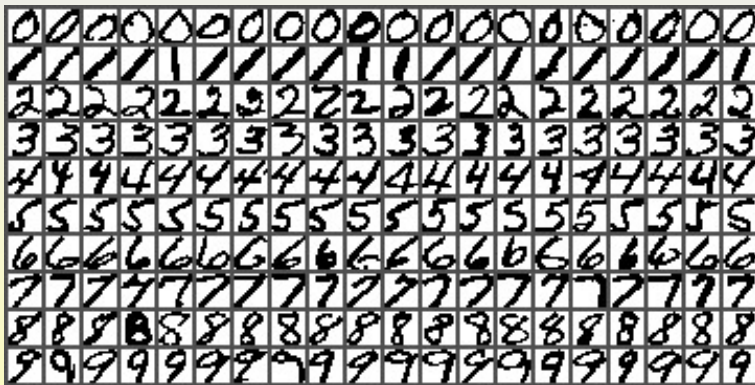
**Remark.** The modified data vectors  $\mathbf{x}^{(t)}$  are “adapted” according to  $P(\mathbf{x})$  and therefore more probable than the initial data vector  $\mathbf{x}^{(0)}$ . Similarly, human eye tends to “modify” visual information according to previous experience.



# Numerals From the NIST SD19 database

**NIST Special Database SD19: about 400000 handwritten numerals**

examples of numerals normalized to 16x16 binary raster



class-means ("mean images") of training numerals



# Recognition of Numerals From the NIST SD19 Database

**data split:** 200000 training numerals and 200000 testing numerals  
(odd data vectors for training, even data vectors for testing)

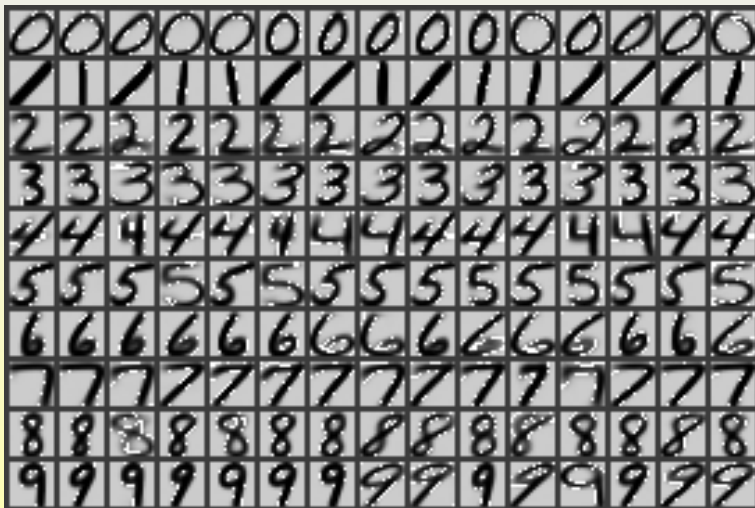
- all numerals normalized to 16x16 binary raster
- structural optimization controlled by simple thresholding
- initial number of components chosen identically in all classes
- random initialization of mixture parameters

**CLASSIFICATION ERROR** in % obtained by different methods and mixtures

Experiment No.	1	2	3	4	5
Number of Components	10	99	191	389	732
Number of Parameters	2005	22083	41986	94910	156819
Exact Bayes Formula	16.55	6.14	4.89	3.80	3.32
Modified Weights	22.63	9.22	7.50	4.93	5.24
Iterated Weights	16.50	6.17	4.95	3.83	3.39
Adapted Input Vector	16.62	6.15	4.89	3.80	3.32

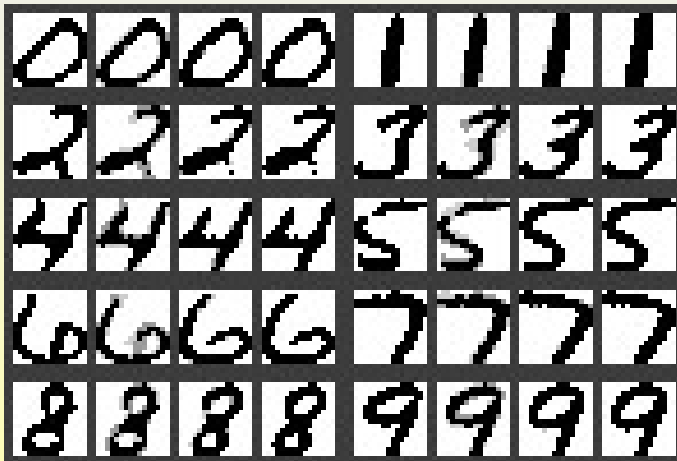
# Component Means of the Estimated Mixtures $P(\mathbf{x}|\omega)$

component parameters  $\theta_{mn} \in \langle 0, 1 \rangle$  displayed as grey levels in raster arrangement (the white fields denote unused variables with  $\phi_{mn} = 0$ )



# Examples of Iteratively Modified Input Patterns

iteratively modified input pattern  $x$  converges in several steps to a local extreme of  $P(x)$  which corresponds to a more probable form of the numeral



# Basic Features of Recurrent Bayesian Reasoning

## Recurrent Bayes Formula

- converges uniquely to asymptotic conditional weights - independently of the initial component weights
- $\Rightarrow$  mixture component weights can be involved into dynamic synaptic processes without destroying the final correct decision making
- resembles natural process of cognition as iteratively improving understanding of input information

## Recurrent Modification of Input Pattern

- converges in few steps to a more probable variant of the input pattern
- $\Rightarrow$  improves the recognition accuracy
- resembles the well known tendency of human eye to modify visual information according to previous experience






# Concluding Remarks

## Basic Features of Probabilistic Neural Networks






- design of one-layer-PNN consists in estimating class-conditional distribution mixtures by means of EM algorithm
- PNN can be trained by a sequential version of EM algorithm
- multilayer PNN can be designed sequentially layer-by-layer by using transformed training data
- hidden layers of PNN transform the classification problem without information loss and minimize the entropy of the output space
- interconnection structure of multilayer PNN may be incomplete
- structure of PNN can be optimized by means of EM algorithm in a statistically correct way
- product mixture components can be interpreted as probabilistic neurons in neurophysiological terms
- probabilistic model of synapse justifies Hebbian principle of learning
- independently trained PNN can be combined both horizontally and vertically



# References 1/3







-  [Grim, J.:](#) On numerical evaluation of maximum - likelihood estimates for finite mixtures of distributions, *Kybernetika* **18** (1982) 173-190
-  [Grim, J.:](#) Design of multilayer neural networks by information preserving transforms. In: *Third European Congress on Systems Science*. (E. Pessa et al. Eds.). Edizioni Kappa, Roma (1996) 977-982
-  [Grim, J.:](#) Information approach to structural optimization of probabilistic neural networks. In: *Fourth European Congress on Systems Science*. (Ferrer L., Caselles A. eds.). SESGE, Valencia (1999) 527-539
-  [Grim, J.:](#) A sequential modification of EM algorithm. In: *Classification in the Information Age*. Eds. Gaul W., Locarek-Junge H. Springer, Berlin (1999) 163-170
-  [Grim, J., Vejvalková, J.:](#) An iterative inference mechanism for the probabilistic expert system PES. *International Journal of General Systems* **27** (1999) 373-396

# References 2/3

-  Grim, J., Kittler, J., Pudil, P., Somol, P.: Multiple classifier fusion in probabilistic neural networks. *Pattern Anal. & Appl.* **5** (2002) 221-233
-  Grim, J., Pudil, P., Somol, P.: Recognition of handwritten numerals by structural probabilistic neural networks. In: *Proceedings of the Second ICSC Symposium on Neural Computation*. (Bothe, H., Rojas, R. eds.). ICSC, Wetaskiwin (2000) 528-534
-  Grim, J., Pudil P., Somol P.: Boosting in probabilistic neural networks. In: *Proc. 16th International Conference on Pattern Recognition*. (Kasturi, R. et al. Eds.). IEEE Comp. Soc., Los Alamitos (2002) 136-139
-  Grim J., Somol P., Novovičová J., Pudil P., Ferri F., (1998b): Initializing normal mixture of densities. In *Proc. 14th Int. Conf. on Pattern Recognition ICPR'98*, A.K. Jain et al. (Eds.), pp. 886-890, IEEE Computer Society: Los Alamitos, California, 1998
-  Vajda, I., Grim, J.: About the maximum information and maximum likelihood principles in neural networks. *Kybernetika*, **34** (1998) 485-494



# References 3/3

-  Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum-likelihood from incomplete data via the EM algorithm, Journal of the Royal Stat. Society **B 39** (1977) 1-38
-  Hebb, D.O. : The Organization of Behavior: A Neuropsychological Theory. New York: Wiley (1949)
-  McLachlan G.J. and Peel D. (2000): *Finite Mixture Models*, John Wiley & Sons, New York, Toronto: 2000
-  Schlesinger, M.I. (1968): Relation between learning and self-learning in pattern recognition." (in Russian), *Kibernetika*, (Kiev), No. 2, pp. 81-88
-  Specht, D.F.: Probabilistic neural networks for classification, mapping or associative memory. In: Proc. of the IEEE International Conference on Neural Networks, **I** (1988) 525-532
-  Streit, L.R., Luginbuhl, T.E.: Maximum-likelihood training of probabilistic neural networks, IEEE Trans. on Neural Networks (1994) 5, 764-783