Pattern Recognition by Probabilistic Neural Networks

Mixtures of Product Components versus Mixtures of Dependence Trees

Jiří Grim[1] and Pavel Pudil[2]

[1] *Institute of Information Theory and Automation*
Academy of Sciences of the Czech Republic, Prague
[2] *Faculty of Management, Prague University of Economics*
Jindřichův Hradec, Czech Republic

IJCCI - NCTA, October 22-25, 2014, Rome, Italy

ÚTIA

Outline

1. Probabilistic Neural Networks
   - Statistical Recognition by Product Mixtures
   - Subspace Mixture Model
   - EM Algorithm for Subspace Mixtures
   - Product Mixture Component as Probabilistic Neuron
2. Mixtures of Dependence Trees
   - Properties of Product Mixtures and Dependence Tree Mixtures
   - Dependence-Tree Concept (Chow & Liu, 1968)
   - Mixtures of Binary Dependence Trees
   - EM Algorithm for Dependence-Tree Mixtures
3. Product Mixtures versus Dependence Tree Mixtures
   - Comparison of Approximation Accuracy: Table Distribution
   - Recognition of Numerals by Mixtures of Dependence Trees
   - Comparison of Recognition Error: NIST Numerals
   - Information Contribution of the Dependence Structures
4. Conclusion

Statistical Approach to Pattern Recognition

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

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

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

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

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

Probabilistic Neural Networks (PNN):

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

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

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

Components $\approx$ Neurons $\approx$ all variables on input

Output Layer: $p(\omega|\mathbf{x}) = \sum_{m \in \mathcal{M}_\omega} q(m|\mathbf{x}), \quad q(m|\mathbf{x}) = \frac{F(\mathbf{x}|m) f(m)}{\sum_{j \in \mathcal{M}} F(\mathbf{x}|j) f(j)}$

ÚTIA

Subspace (Structural) Mixture Model

SUBSPACE MODEL: to avoid complete interconnection property

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

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

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

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

$G(\mathbf{x}|m,\phi_m) \approx$ defined on a subspace 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. The subspace mixture model provides statistically correct solution for Bayesian decision-making.

ÚTIA

Subspace Optimization by EM Algorithm (Grim, 1986)

subspace structure can be optimized by EM algorithm in full generality:

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

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

$$q(m|\mathbf{x}) = \frac{G(\mathbf{x}|m, \phi_m) w_m}{\sum_{j \in \mathcal{M}_\omega} G(\mathbf{x}|j, \phi_j) w_j}, \qquad f_n(x_n|m) = (\theta_{mn})^{x_n} (1 - \theta_{mn})^{1 - x_n},$$

$$w_m^{'} = \frac{1}{|\mathcal{S}_\omega|} \sum_{\mathbf{x} \in \mathcal{S}_\omega} q(m|\mathbf{x}), \qquad \theta_{mn}^{'} = \frac{1}{\sum_{\mathbf{x} \in \mathcal{S}_\omega} q(m|\mathbf{x})} \sum_{\mathbf{x} \in \mathcal{S}_\omega} x_n q(m|\mathbf{x})$$

structural criterion: Kullback-Leibler information divergence

$$\gamma_{mn}^{'} = w_m^{'} \left[\theta_{mn}^{'} \log \frac{\theta_{mn}^{'}}{\theta_{0n}} + (1 - \theta_{mn}^{'}) \log \frac{(1 - \theta_{mn}^{'})}{(1 - \theta_{0n})} \right] = \mathcal{I}(f_n^{'}(\cdot|m) || f_n(\cdot|0))$$

subspace structure optimization: $\phi_{mn}^{'} = 1$ for the r highest values $\gamma_{mn}^{'}$

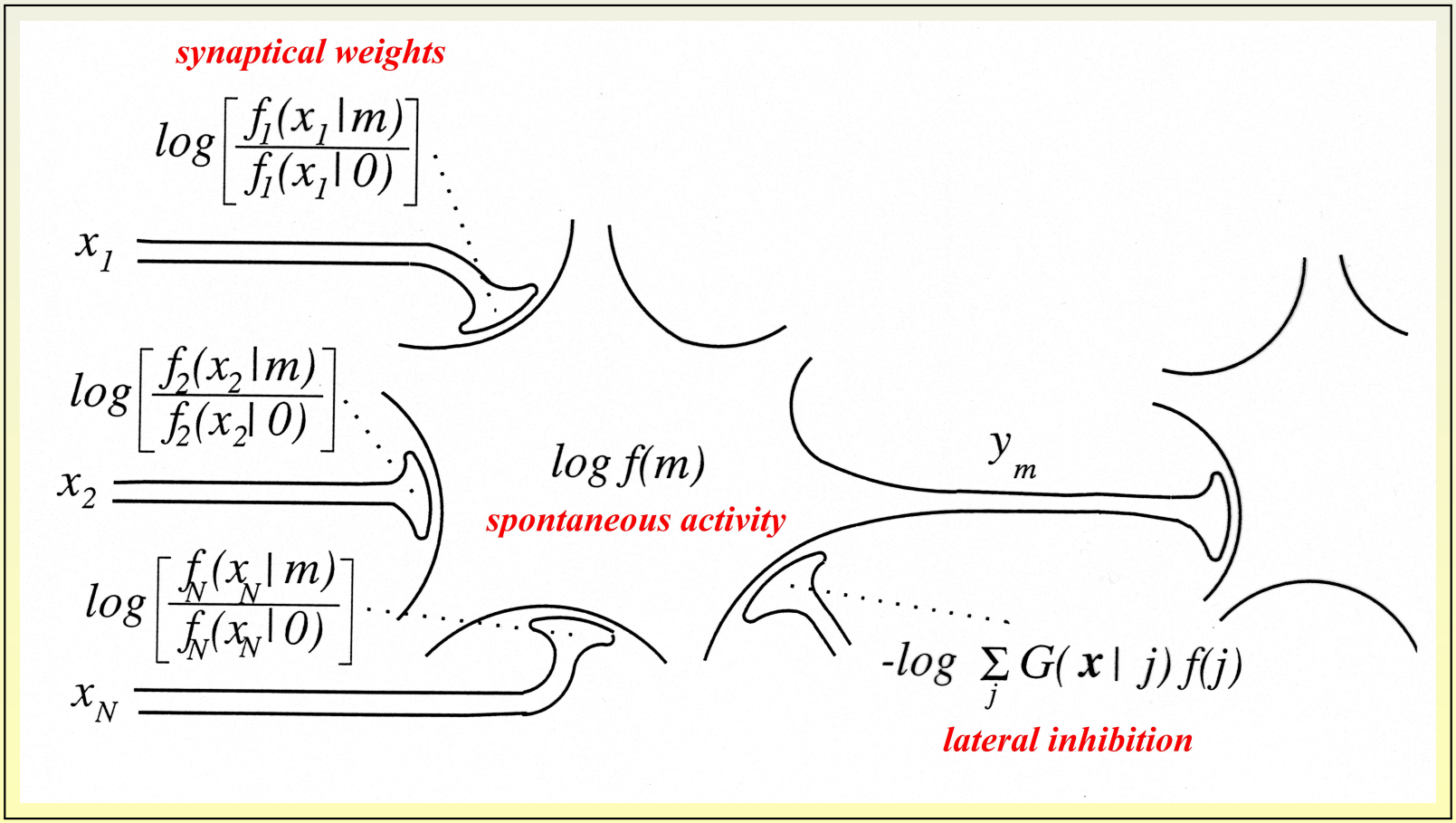
Remark. **The "subspace" EM algorithm converges monotonically.**

ÚTIA

Probabilistic Neural Networks (Grim et al. 1999-2012)

probabilistic neuron: **interpretation of mixture components**

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



▸ Hebb's postulate of learning

Product Mixtures and Mixtures of Dependence Tree

product mixtures:

- ⊕ marginals simply available by omitting superfluous product terms
- ⊕ computationally efficient implementation of EM algorithm
- ⊕ EM algorithm directly applicable to incomplete data
- ⊕ support "subspace" modification (component specific features)
- ⊖ restrictive assumption: conditional independence of variables

$$P(\mathbf{x}) = \sum_m w_m \prod_{n=1}^{N} f(x_n|m) \quad \otimes \quad P(\mathbf{x}) = \sum_m w_m f(x_1|m) \prod_{n=2}^{N} f(x_n|x_{k_n}, m)$$

mixtures of dependence trees

- ⊕ statistical relationship between two variables by a single component
- ⊕ structural optimization by maximum weight spanning tree
- ⊖ difficult evaluation of marginal distributions
- ⊖ computationally demanding implementation of EM algorithm

Dependence-Tree Concept (Chow & Liu, 1968)

chain expansion formula:

$$P(\mathbf{x}) = f(x_1) \prod_{n=2}^{N} f(x_n | x_{n-1}, \ldots, x_1), \quad \mathbf{x} = (x_1, x_2, \ldots, x_N) \in \mathbf{X},$$

dependence-tree expansion:

$\pi = (i_1, i_2, \ldots, i_N) \approx$ permutation of the index set $\mathcal{N} = \{1, 2, \ldots, N\}$

$$P(\mathbf{x}|\pi) = f(x_{i_1}) \prod_{n=2}^{N} f(x_{i_n} | x_{j_n}), \quad j_n \in \{i_1, \ldots, i_{n-1}\}$$

$$P(\mathbf{x}|\pi) = f(x_{i_1}) \prod_{n=2}^{N} \frac{f(x_{i_n}, x_{j_n})}{f(x_{j_n})} = \left[\prod_{n=1}^{N} f(x_{i_n})\right] \left[\prod_{n=2}^{N} \frac{f(x_{i_n}, x_{j_n})}{f(x_{i_n}) f(x_{j_n})}\right],$$

in natural ordering:

$$P(\mathbf{x}|\alpha, \theta) = \prod_{i=1}^{N} f(x_i) \prod_{n=2}^{N} \frac{f(x_n, x_{k_n})}{f(x_n) f(x_{k_n})} = f(x_1) \prod_{n=2}^{N} f(x_n | x_{k_n})$$

marginals: $\theta = \{f(x_n, x_{k_n}), n = 2, .., N\} \quad // \Rightarrow \quad \{f(x_n), n = 1, .., N\}$

dependence structure: $\alpha = (k_2, .., k_N) \quad \rightarrow$ **to be optimized**

ÚTIA

Mixtures of Binary Dependence Trees (Grim, 1984)

approximation of $P(\mathbf{x}|\omega), \omega \in \Omega$ by dependence-tree mixtures:

$$P(\mathbf{x}|\mathbf{w}, \boldsymbol{\alpha}, \boldsymbol{\Theta}, \omega) = \sum_{m \in \mathcal{M}_\omega} w_m F(\mathbf{x}|\boldsymbol{\alpha}_m, \boldsymbol{\theta}_m) = \sum_{m \in \mathcal{M}_\omega} w_m f(x_1|m) \prod_{n=2}^{N} f(x_n|x_{k_n}, m)$$

$\mathbf{w} = (w_m, m \in \mathcal{M}_\omega)$ $\approx$ weight vector
$\boldsymbol{\alpha} = \{\boldsymbol{\alpha}_m, m \in \mathcal{M}_\omega\}$ $\approx$ structural parameters
$\boldsymbol{\Theta} = \{\boldsymbol{\theta}_m, m \in \mathcal{M}_\omega\}$ $\approx$ two-dimensional marginals
$\boldsymbol{\theta}_m = \{f(x_n, x_{k_n}|m), n = 2, \ldots, N\}$

maximum likelihood criterion:

$$L(\mathbf{w}, \boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{|\mathcal{S}_\omega|} \sum_{x \in \mathcal{S}_\omega} \log \left[\sum_{m \in \mathcal{M}_\omega} w_m F(\mathbf{x}|\boldsymbol{\alpha}_m, \boldsymbol{\theta}_m) \right], \quad (\omega \in \Omega)$$

$$L(\mathbf{w}, \boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{|\mathcal{S}_\omega|} \sum_{x \in \mathcal{S}_\omega} \log \left[\sum_{m \in \mathcal{M}_\omega} w_m f(x_1|m) \prod_{n=2}^{N} f(x_n|x_{k_n}, m) \right]$$

Remark. **The dependence structure of mixture components α_m can be optimized by means of EM algorithm in full generality.**

ÚTIA

EM Algorithm for Binary Dependence-Tree Mixtures

EM algorithm: iterative maximization of $Q_m(\alpha_m, \theta_m)$ (see paper)

$$Q_m(\alpha_m, \theta_m) = \sum_{x \in \mathcal{S}_\omega} \frac{q(m|\mathbf{x})}{w_m' |\mathcal{S}_\omega|} \log F(\mathbf{x}|\alpha_m, \theta_m), \quad q(m|\mathbf{x}) = \frac{w_m F(\mathbf{x}|\alpha_m, \theta_m)}{P(\mathbf{x}|\mathbf{w}, \alpha, \Theta)},$$

denoting:

$$\hat{f}(\xi_n|m) = \sum_{x \in \mathcal{S}_\omega} \frac{q(m|\mathbf{x})}{w_m' |\mathcal{S}|} \delta(\xi_n, x_n), \quad \hat{f}(\xi_n, \xi_{k_n}|m) = \sum_{x \in \mathcal{S}_\omega} \frac{q(m|\mathbf{x})}{w_m' |\mathcal{S}|} \delta(\xi_n, x_n)\delta(\xi_{k_n}, x_{k_n})$$

we can write: (see paper)

$$Q_m(\alpha_m, \theta_m) = \sum_{x \in \mathcal{S}} \frac{q(m|\mathbf{x})}{w_m' |\mathcal{S}|} \left[\log f(x_1|m) + \sum_{n=2}^{N} \log f(x_n|x_{k_n}, m) \right] =$$

$$= \sum_{\xi_1=0}^{1} \hat{f}(\xi_1|m) \log f(\xi_1|m) + \sum_{n=2}^{N} \sum_{\xi_{k_n}=0}^{1} \hat{f}(\xi_{k_n}|m) \sum_{\xi_n=0}^{1} \frac{\hat{f}(\xi_n, \xi_{k_n}|m)}{\hat{f}(\xi_{k_n}|m)} \log f(\xi_n|\xi_{k_n}, m)$$

$\Rightarrow$ for any fixed α_m weighted likelihood $Q_m(\alpha_m, \theta_m)$ is maximized by

$$\theta_m' : \quad f(\xi_n|m) = \hat{f}(\xi_n|m), \quad f(\xi_n|\xi_{k_n}, m) = \hat{f}(\xi_n, \xi_{k_n}|m)/\hat{f}(\xi_{k_n}|m)$$

ÚTIA

EM Algorithm for Binary Dependence-Tree Mixtures

making substitutions: $f'(\xi_n|m) = \hat{f}(\xi_n|m), \quad f'(\xi_n|\xi_{k_n}, m) = \hat{f}(\xi_n|\xi_{k_n}, m)$

we can write:

$$Q_m(\alpha_m, \theta'_m) = \sum_{n=1}^{N} \sum_{\xi_n=0}^{1} f'(\xi_n|m) \log f'(\xi_n|m) +$$

$$+ \sum_{n=2}^{N} \sum_{\xi_n=0}^{1} \sum_{\xi_{k_n}=0}^{1} f'(\xi_n, \xi_{k_n}|m) \log \frac{f'(\xi_n, \xi_{k_n}|m)}{f'(\xi_n|m) f'(\xi_{k_n}|m)},$$

and, by using the Shannon information formula, we obtain:

$$Q_m(\alpha_m, \theta'_m) = \sum_{n=1}^{N} -H(f'_{n|m}) + \sum_{n=2}^{N} \mathcal{I}(f'_{n|m}, f'_{k_n|m}), \quad m \in \mathcal{M}_\omega$$

therefore $Q_m(\alpha_m, \theta'_m)$ as a function of α_m is maximized by the maximum-weight spanning-tree: ▸ MWST Algorithm

$$\alpha'_m = \arg\max_{\alpha_m} \left\{ \sum_{n=2}^{N} \mathcal{I}(f'_{n|m}, f'_{k_n|m}) \right\}, \qquad \mathcal{I}(f'_{n|m}, f'_{k_n|m}) \approx \textbf{edge weight}$$

ÚTIA

SUMMARY: Dependence-Tree EM Algorithm

CRITERION: $$L = \frac{1}{|\mathcal{S}_\omega|} \sum_{x \in \mathcal{S}_\omega} \log \Big[\sum_{m \in \mathcal{M}_\omega} w_m f(x_1|m) \prod_{n=2}^{N} f(x_n|x_{k_n}, m) \Big]$$

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

$$q(m|\mathbf{x}) = \frac{w_m F(\mathbf{x}|\boldsymbol{\alpha}_m, \boldsymbol{\theta}_m)}{P(\mathbf{x}|\mathbf{w}, \boldsymbol{\alpha}, \boldsymbol{\Theta}, \omega)}, \quad w_m^{'} = \frac{1}{|\mathcal{S}_\omega|} \sum_{\mathbf{x} \in \mathcal{S}_\omega} q(m|\mathbf{x})$$

for all pairs (n, k)**:**

$$\boldsymbol{\theta}_m^{'}: \quad f^{'}(\xi_n|\xi_k, m) = \hat{f}(\xi_n, \xi_k|m)/\hat{f}(\xi_k|m), \quad (f^{'}(\xi_n|m) = \hat{f}(\xi_n|m)),$$

for any fixed $\boldsymbol{\alpha}_m$**:**

$$\mathcal{I}(f_n^{'}, f_{k_n}^{'}) = \sum_{x_n=0}^{1} \sum_{x_{k_n}=0}^{1} f^{'}(x_n, x_{k_n}) \log \frac{f^{'}(x_n, x_{k_n})}{f^{'}(x_n) f^{'}(x_{k_n})} \quad \approx \textbf{edge weight}$$

optimal structure: **maximum weight spanning trees**

$$\boldsymbol{\alpha}_m^{'} = \arg\max_{\boldsymbol{\alpha}_m} \left\{ \sum_{n=2}^{N} \mathcal{I}(f_{n|m}^{'}, f_{k_n|m}^{'}) \right\}, \quad m \in \mathcal{M}_\omega$$

ÚTIA

Approximation Accuracy of a Binary Table Distribution

P^*: original; P_1: product of marginals; P_2: Chow & Liu; P_3: Ku & Kullback; product mixtures P_4: M=2; P_5, P_6: M=3; dependence tree mixture P_7: M=2;

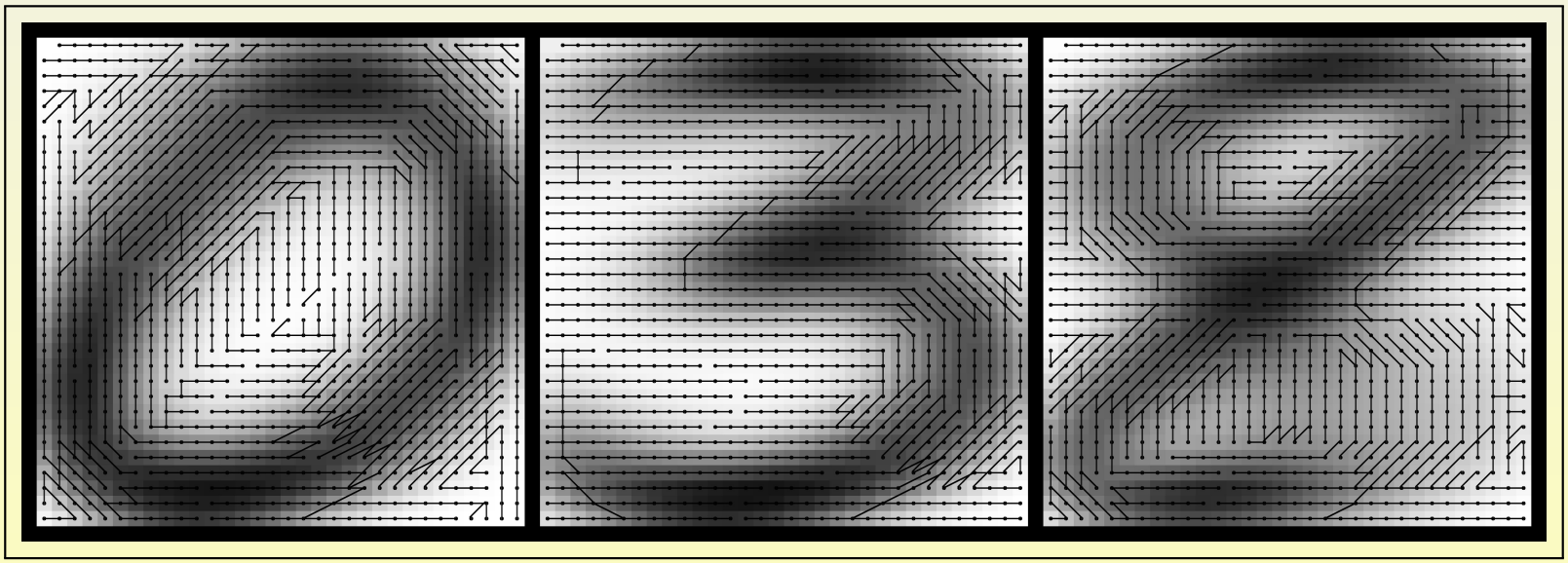
x_1	x_2	x_3	x_4	$P^*(x)$	$P_1(x)$	$P_2(x)$	$P_3(x)$	$P_4(x)$	$P_5(x)$	$P_6(x)$	$P_7(x)$
0	0	0	0	·1000	·0456	·1296	·09977	·1037	·1000	·0857	·1000
0	0	0	1	·1000	·0456	·1037	·10000	·1296	·1000	·1143	·1000
0	0	1	0	·0500	·0557	·0370	·04958	·0296	·0500	·0600	·0500
0	0	1	1	·0500	·0557	·0296	·04927	·0370	·0500	·0400	·0500
0	1	0	0	·0000	·0557	·0152	·00051	·0149	·0000	·0000	·0000
0	1	0	1	·0000	·0557	·0121	·00026	·0124	·0000	·0000	·0000
0	1	1	0	·1000	·0681	·0681	·10011	·0669	·0833	·0900	·1000
0	1	1	1	·0500	·0681	·0546	·05035	·0558	·0667	·0600	·0500
1	0	0	0	·0500	·0557	·0530	·05027	·0519	·0600	·0643	·0500
1	0	0	1	·1000	·0557	·0636	·09996	·0648	·0900	·0857	·1000
1	0	1	0	·0000	·0681	·0152	·00039	·0148	·0000	·0000	·0000
1	0	1	1	·0000	·0681	·0182	·00076	·0185	·0000	·0000	·0000
1	1	0	0	·0500	·0681	·0331	·04945	·0397	·0400	·0500	·0500
1	1	0	1	·0500	·0681	·0397	·04978	·0331	·0600	·0500	·0500
1	1	1	0	·1500	·0832	·1488	·14992	·1785	·1667	·1500	·1500
1	1	1	1	·1500	·0832	·1785	·14962	·1488	·1333	·1500	·1500
Number of param.				15	4	7	28	9	14	14	15
Number of comp.				—	1	1	1	2	3	3	2
$\mathcal{H}(P^*, P_i)$				·0000	·3687	·0952	·0098	·0952	·0092	·0084	·0000

(J. Grim, Kybernetika, Vol. 20, No. 1, pp. 1-17, 1984)

Recognition of Numerals by Mixtures of Dependence Trees

approximation of conditional distributions by mixtures of dependence-tree components:

$$P(\mathbf{x}|\omega) = P(\mathbf{x}|\mathbf{w}, \alpha, \mathbf{\Theta}, \omega) = \sum_{m \in \mathcal{M}_\omega} w_m f(x_1|m) \prod_{n=2}^{N} f(x_n|x_{k_n}, m)$$



examples of dependence-tree components (numerals: 0, 3, 8)

(number of components $|\mathcal{M}| = 400$, number of parameters: $|\mathbf{\Theta}| = 819200$)

Comparison of Recognition Error: NIST Numerals

Recognition of numerals by mixtures of product components:

$$P(\mathbf{x}|\omega) = \left[\prod_{n\in\mathcal{N}} f_n(x_n|0)\right] \sum_{m\in\mathcal{M}_\omega} w_m \prod_{n\in\mathcal{N}} \left[\frac{f_n(x_n|m)}{f_n(x_n|0)}\right]^{\phi_{mn}}, \quad \omega \in \Omega$$

Experiment No.	I	II	III	IV	V	VI	VII	VIII	IX
Components	10	40	100	299	858	1288	1370	1459	1571
Parameters	10240	38758	89973	290442	696537	1131246	1247156	1274099	1462373
Classif. error in %	11.93	4.81	4.28	2.93	2.40	1.95	1.91	1.86	1.84

Recognition of numerals by mixtures of dependence-tree components:

$$P(\mathbf{x}|\omega) = \sum_{m\in\mathcal{M}_\omega} w_m f(x_1|m) \prod_{n=2}^{N} f(x_n|x_{k_n}, m), \quad \omega \in \Omega$$

Experiment No.	I	II	III	IV	V	VI	VII	VIII	IX
Components	10	40	80	100	150	200	300	400	500
Parameters	20480	81920	163840	204800	307200	409600	614400	819200	1024000
Classif. error in %	6.69	4.13	2.86	2.64	2.53	2.22	2.13	1.97	2.01

Error Matrix: Product Mixture x Dependence Tree Mixture

CLASS	0	1	2	3	4	5	6	7	8	9	false n.
0	19950	8	43	19	39	32	36	0	38	17	1.1 %
1	2	22162	30	4	35	7	18	56	32	6	0.9 %
2	32	37	19742	43	30	9	8	29	90	16	1.5 %
3	20	17	62	20021	4	137	2	28	210	55	2.6 %
4	11	6	19	1	19170	11	31	51	30	247	2.1 %
5	25	11	9	154	4	17925	39	6	96	34	2.1 %
6	63	10	17	6	23	140	19652	1	54	3	1.6 %
7	7	12	73	10	73	4	0	20497	22	249	2.1 %
8	22	25	53	97	30	100	11	11	19369	72	2.1 %
9	15	13	25	62	114	22	3	146	93	19274	2.5 %
false p.	0.9%	0.7%	2.7%	2.0%	1.7%	2.3%	0.7%	1.6%	3.3%	3.5%	1.84%

Error matrix for a product mixture model ($|\mathcal{M}|$=1571, $|\Theta|$=1462373)

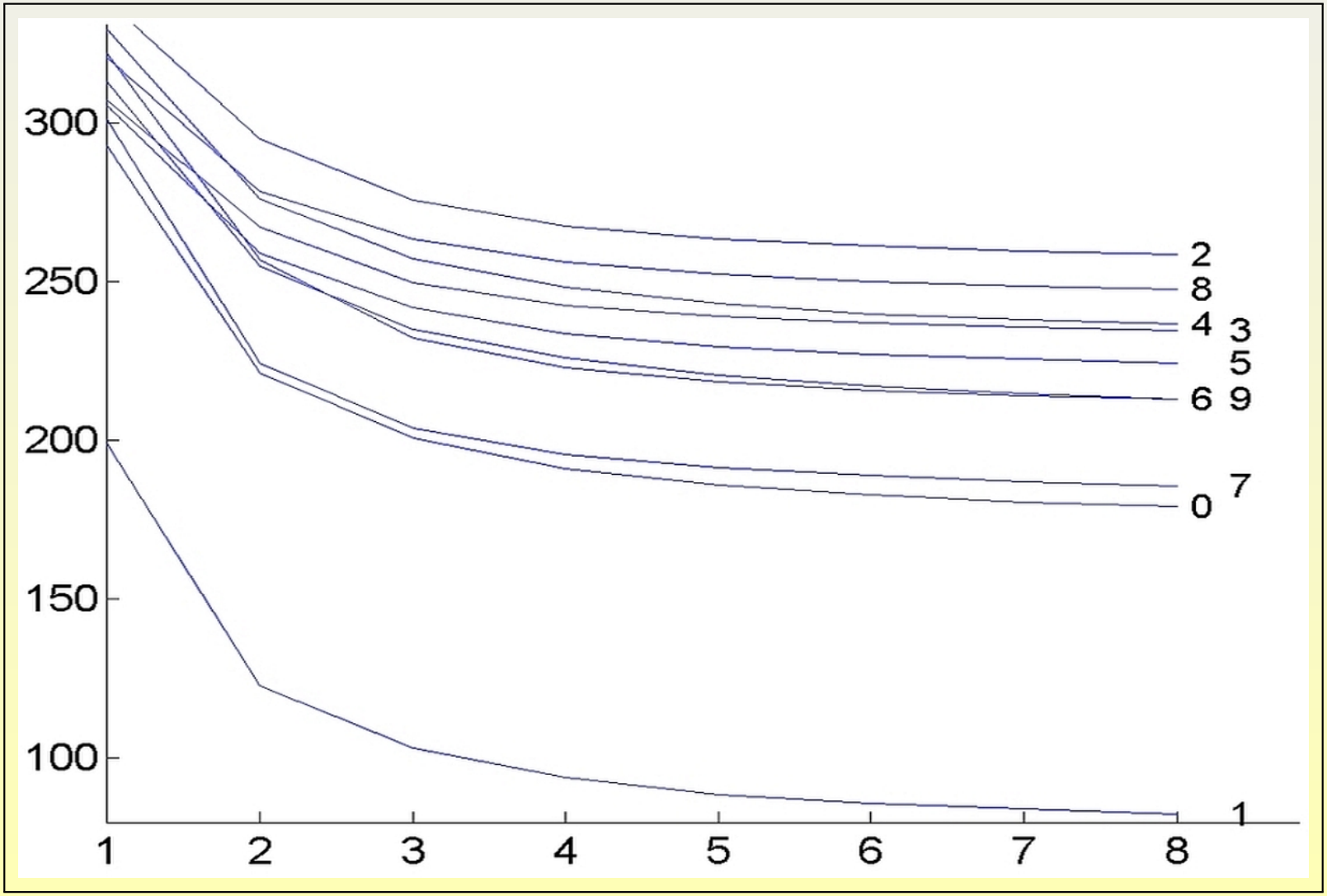
CLASS	0	1	2	3	4	5	6	7	8	9	false n.
0	19979	11	62	21	18	26	25	2	28	10	1.0 %
1	5	21981	78	13	74	1	20	155	21	4	1.7 %
2	22	15	19777	72	26	5	6	35	72	6	1.3 %
3	20	10	66	20169	1	120	1	20	122	27	1.9 %
4	12	16	13	4	19245	1	13	52	44	177	1.7 %
5	25	5	15	157	8	17874	45	9	129	36	2.3 %
6	100	19	38	25	43	90	19575	1	75	3	2.0 %
7	17	33	108	24	71	0	0	20367	28	299	2.8 %
8	18	30	47	167	27	55	22	17	19337	70	2.3 %
9	12	20	62	74	89	33	3	144	134	19196	2.9 %
false p.	1.4%	0.7%	2.4%	2.7%	1.8%	1.8%	0.7%	1.6%	3.1%	3.2%	1.97%

Error matrix for a dependence-tree mixture model ($|\mathcal{M}|$=400, $|\Theta|$=819200)

ÚTIA

Information Contribution of the Dependence Structures

overall spanning-tree weight in iterations $1 \div 8$ for the numerals $0 \div 9$



Conclusion

small number of multidimensional components:

- a single dependence tree can describe statistical relations of variables
- dependence trees may essentially increase approximation power of a small number of product components
- information contribution of a small number of dependence tree components can increase in the course of EM iterations

large number of components:

- mixture of many dependence trees is similar to nonparametric Parzen estimate, the form of the kernels is less relevant
- dependence structure of components does not increase the approximation power of large product mixtures essentially
- in large mixtures the information contribution of dependence structures usually decreases in final EM iterations
- optimal estimate of a large dependence tree mixture tends to approach a simple product mixture model

Literatura 1/3

- C. Chow and C. Liu, "Approximating discrete probability distributions with dependence trees", *IEEE Trans. on Information Theory*, Vol. IT-14, No.3, pp. 462- 467, 1968.
- J. Grim, "On structural approximating multivariate discrete probability distributions", *Kybernetika*, Vol. 20, No. 1, pp. 1-17, 1984. *http://dml.cz/dmlcz/125676*
- M. Meila and M.I. Jordan, "Learning with mixtures of trees", *Journal of Machine Learning Research*, Vol. 1, No. 9, pp. 1-48, 2001.
- I.J. Kim and J.H. Kim, "Statistical Character Structure Modeling and Its Application to Handwritten Chinese Character Recognition", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 11, pp. 1422-1436, 2003.
- S. Kirshner and P. Smyth, "Infinite mixtures of trees", *Proc. of the 24th International Conference on Machine Learning (ICML'07)*, Ed. Zoubin Ghahramani, ACM, New York, USA, pp. 417-423, 2007.

Back

Literatura 2/3

- B. Behsaz and M. Rahmati. "Estimation of Probability Density Function by Dependence Tree Methods for Pattern Recognition Systems." *Tech. Rep. U. Alberta*, 1, 2006.
- A.P. Dempster, N.M. Laird and D.B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm", *J. Roy. Statist. Soc., B*, Vol. 39, pp. I-38, 1977.
- J. Grim, "On numerical evaluation of maximum - likelihood estimates for finite mixtures of distributions", *Kybernetika*, Vol. I8, No.3, pp.173-190, 1982. *http://dml.cz/dmlcz/124132*
- J. Grim, "Multivariate statistical pattern recognition with nonreduced dimensionality", *Kybernetika*, Vol. 22, No. 2, pp. 142-157, 1986. *http://dml.cz/dmlcz/125022*
- J. Grim, "Preprocessing of Screening Mammograms Based on Local Statistical Models", *Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies, ISABEL 2011*, Barcelona, ACM, pp. 1-5, 2011

Back

Literatura 3/3

- O. Borůvka, "On a minimal problem", *Transaction of the Moravian Society for Natural Sciences* (in Czech), No. 3, 1926.
- V. Jarník: "About a certain minimal problem", *Transaction of the Moravian Society for Natural Sciences* (in Czech), No. 6, 1930.
- J.B Kruskal, "On the shortest spanning sub-tree of a graph", *Proc. Amer. Math. Soc.*, No. 7, pp. 48-50, 1956.
- R.C. Prim, "Shortest connection networks and some generalizations", *Bell System Tech. J.*, Vol. 36 , pp. 1389-1401, 1957.
- E. Parzen, "On estimation of a probability density function and its mode," *Annals of Mathematical Statistics*, Vol. 33., pp. 1065-1076, 1962.

Remark: Both J.B. Kruskal and R.C. Prim refer to an "... obscure Czech paper of O. Borůvka ..." describing construction of the minimum-weight spanning tree and the corresponding proof of uniqueness. The Prim's algorithm was developed in 1930 by Czech mathematician Vojtěch Jarník

◂ Back (Outline)

Appendix: Maximum-Weight Spanning-Tree Construction

```
//*************************************************************************
// Maximum-weight spanning tree construction (Jarník, 1930, Prim, 1957)
//*************************************************************************
//  NN........ number of nodes, N=1,2,...,NN
//  T[N]...... characteristic function of the known part of spanning tree
//  E[N][K]... positive weight of the edge <N,K>
//  A[K]...... index of the heaviest neighbor of node K in the known subtree
//  GE[K]..... greatest edge weight between the node K and the known subtree
//  K0........ index of the most heavy neighbor of the defined part of tree
//  SUM....... total weight of the spanning tree: {<2,A[2]>,...,<NN,A[NN]>}
//*************************************************************************
    for(N=1; N<=NN; N++) {GE[N]=-1; T[N]=0; A[N]=0;}  //  initial values
    N0=1; T[N0]=1; K0=0;
    for(I=2; I<=NN; I++)
    {  FMAX=-1E0;
        for(N=2; N<=NN; N++) if(T[N]<1)
        {  F=E[N0][N];
        if(F>GE[N]) {GE[N]=F; A[N]=N0;}  else F=GE[N];
        if(F>FMAX)  {FMAX=F; K0=N;}
        }  //  end of N-loop
        N0=K0;  T[N0]=1;  SUM+=FMAX;
    }  // end of spanning tree construction
```

Back

Product Mixture Component as Probabilistic Neuron

probabilistic neuron: **interpretation of mixture components**

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

$q(m\|\mathbf{x})$	$\approx$	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
$\log [\sum_{j\in\mathcal{M}} G(\mathbf{x}\|j,\phi_j) f(j)]$	$\approx$	common "norming" term (lateral inhibition)

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

Hebb's postulate of learning (Hebb, 1949)

"When an axon of cell A ($\approx n$) is near enough to excite a cell B ($\approx m$) 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."

Back