Generative Modeling of Tree-Structured Data

Davide Bacciu

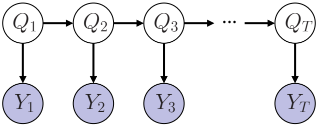
Dipartimento di Informatica
Università di Pisa
bacciu@di.unipi.it

Machine Learning: Neural Networks and Advanced Models (AA2)

Outline of the Talk

- Refresher on hidden Markov Models for sequences
- Hidden Markov Models for trees
- Discriminative approaches within the generative paradigm
  - Input-driven Markov models
- Application examples
  - Tree data mapping and visualization

Hidden Markov Model (HMM)



- Elements of an observed sequence $\mathbf{y} = y_1, \ldots, y_T$ are generated by an hidden process regulated by corresponding state variables $\{Q_1, \ldots, Q_T\}$
- Past independent of the future given the present (Markov Assumption)

$$P(Q_t | Q_{t-1}, \ldots, Q_1) = P(Q_t | Q_{t-1})$$

- Currently observed element of the sequence is generated based only on current hidden state

$$P(Y_t | Q_T, \ldots, Q_1, Y_T, \ldots, Y_1) = P(Y_t | Q_t)$$

HMM Parameters

Stationariety assumption $\rightarrow$ time-independent parameterization

1. State transition distribution

$$A_{ij} = P(Q_t = i | Q_{t-1} = j), \quad \sum_{i=1}^{C} A_{ij} = 1$$

2. Prior distribution (for $t = 1$)

$$\pi_i = P(Q_1 = i), \quad \sum_{i=1}^{C} \pi_i = 1$$

3. Label emission distribution

$$B_{ki} = P(Y_t = k | Q_t = i), \quad \sum_{k=1}^{K} B_{ki} = 1$$

Inference and Learning in HMM

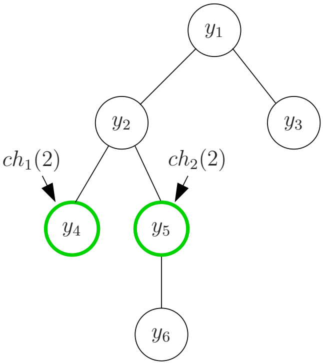
The three classic problems of HMMs

Smoothing Given the observed sequence $\mathbf{y}$ and a model $\lambda = \{A, B, \pi\}$ compute $P(Q_t|\mathbf{y}, \lambda)$

Learning Adjust the model parameters $\lambda = \{A, B, \pi, \phi\}$ to maximize $P(\mathbf{y}|\lambda)$

Decoding Given the observed sequence $\mathbf{y}$ and a model $\lambda = \{A, B, \pi\}$ select the *optimal* hidden state assignment $\mathbf{Q}$

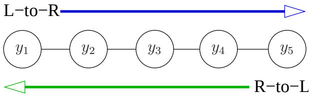
Tree Structured Data



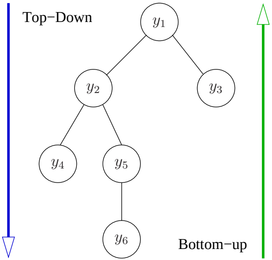
- Labeled rooted trees with finite outdegree L
- $u \rightarrow$ node index
- $y_u \rightarrow$ observation (label)
- $ch_l(u) \rightarrow$ l-th child of u
- Node position with respect to its siblings is relevant for information representation (positional trees)

Generative Process

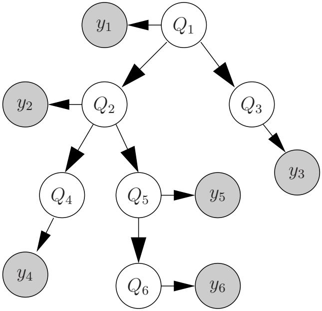
Sequences can be parsed (generated) left-to-right or right-to-left



Do we have generation directions also in trees?



Top-down Hidden Tree Markov Model (THTMM)



Generative model of all the paths from the root to the leaves

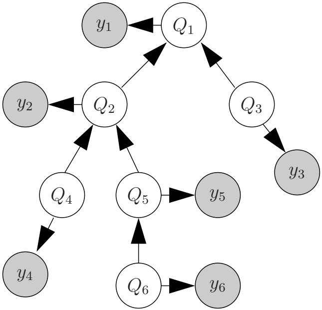
- Generative process from the root to the leaves
  - Q_u hidden state at node u
  - Label emission governed by $P(y_u|Q_u)$
- Markov assumption (conditional dependence)

$$Q_u \rightarrow Q_{ch_l(u)} \quad l = 1, \ldots, L$$

- Parent to children hidden state transition

$$P(Q_{ch_l(u)}|Q_u)$$

Bottom-up Hidden Tree Markov Model (BHTMM)



- Generative process from the leaves to the root
- Markov assumption (conditional dependence)

$$Q_{ch_1(u)}, \ldots, Q_{ch_L(u)} \to Q_u$$

- Children to parent hidden state transition

$$P(Q_u | Q_{ch_1(u)}, \ldots, Q_{ch_L(u)})$$

Generative model of all substructure compositions in the tree

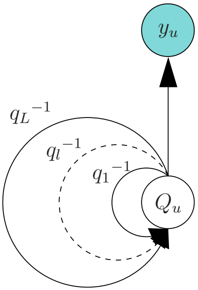
Bottom-up Vs Top-Down

- Direction of the generative process matters when dealing with trees
  - Top-down and bottom-up automata have different expressive power
- Modeling paths (TD) versus modeling substructures (BU)
  - BU allows recursive processing (compositionality)
  - TD cannot model dependence between sibling nodes
- Conditional independence assumptions change drastically

$$P(Q_u | Q_{ch_1(u)}, \ldots, Q_{ch_L(u)}) \text{ vs } P(Q_{ch_l(u)} | Q_u)$$

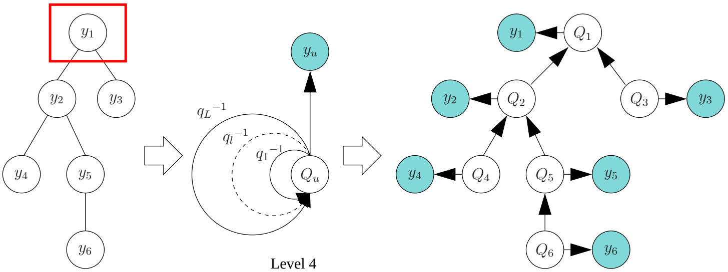
BHTMM Recursive Model

Trees are generated by an hidden probabilistic process from the leaves to the root



- Regulated by the hidden state random variables Q_u
  - Represent information on the substructure rooted in node u
- Q_u depends on the context from the l-th child q_l^{-1}
  - *Simpler* structures are processed first
  - Exploit substructure information to process compound entities

BHTMM Encoding Example



Combinatorial Problem

$$P(Q_u | Q_{ch_1(u)}, \ldots, Q_{ch_L(u)})$$

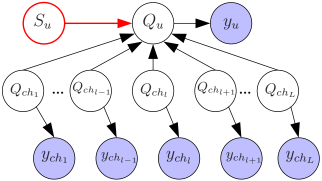
State transition distribution is $O(C^{L+1})$ for a C-dimensional hidden state space

Switching-Parent BHTMM (SP-BHTMM)

Key Idea

Approximate the joint state transition distribution as a mixture of pairwise transition matrices

- Introduce a child selector variable for each parent u
- Switching Parent $S_u \in \{1, \ldots, L\}$
- $P(S_u = l)$ measures the influence of the l-th child on the state transition to Q_u



$$P(Q_u | Q_{ch_1(u)}, \ldots, Q_{ch_L(u)}) = \sum_{l=1}^{L} P(S_u = l) P(Q_u | Q_{ch_l(u)})$$

Summary of SP-BHTMM Parameters

1. State transition distribution (child-parent)

$$A_{ij} = P(Q_u = i | Q_{ch_l(u)} = j), \quad \sum_{i=1}^{C} A_{ij} = 1$$

2. Prior distribution (for leaves states)

$$\pi_i = P(Q_u = i), \quad \sum_{i=1}^{C} \pi_i = 1$$

3. Switching-parents distribution

$$\phi_l = P(S_u = l), \quad \sum_{l=1}^{L} \phi_l = 1$$

4. Label emission distribution

$$B_{ki} = P(Y_u = k | Q_u = i), \quad \sum_{k=1}^{K} B_{ki} = 1$$

The three basic problems in HTMM...

Same problems as in HMMs for sequences

Smoothing Given the observed tree $\mathbf{y}$ and a model $\lambda = \{A, B, \pi, \phi\}$ compute $P(Q_u|\mathbf{y}, \lambda)$

Learning Adjust the model parameters $\lambda = \{A, B, \pi, \phi\}$ to maximize $P(\mathbf{y}|\lambda)$

Decoding Given the observed tree $\mathbf{y}$ and a model $\lambda = \{A, B, \pi, \phi\}$ select the *optimal* hidden state assignment $\mathbf{Q}$

...and their three solutions

Exploit a smart factorization on the structure of the tree to make solutions computationally tractable

Smoothing (Upward-Downward Algorithm) Introduce, by marginalization, the hidden states for each $ch_l(u)$ and estimate a factorization of

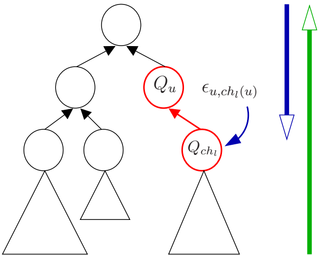
$$P(Q_u|\mathbf{y},\lambda) = \sum_{\mathbf{q}_{\mathbf{ch}(u)}} P(\mathbf{q}_{\mathbf{ch}(u)}, Q_u|\mathbf{y},\lambda)$$

Learning (EM Algorithm) Perform a double upward-downward recursion to compute posteriors $P(Q_u, Q_{ch_l(u)}, S_u|\mathbf{y})$ and use them to update λ

Decoding (Viterbi Algorithm) Maximize a factorization of

$$P(\mathbf{y},\mathbf{q})$$

Upwards-Downwards Algorithm



- Message passing based on the Bayesian factorization

$$P(Q_u = i|\mathbf{y}) = \frac{P(\mathbf{y}_{1\backslash u}|Q_u = i)}{P(\mathbf{y}_{1\backslash u}|\mathbf{y}_u)} P(Q_u = i|\mathbf{y}_u)$$

- Upwards pass estimates

$$\beta_u(i) = P(Q_u = i|\mathbf{y}_u)$$

- $\beta_u(i)$ serves to compute $P(\mathbf{y}|\lambda)$

- Downwards pass uses $\beta_u(i)$ to estimate the posterior

$$\epsilon_{u,ch_l(u)}(i,j) = P(Q_u = i, Q_{ch_l(u)} = j, S_u = l|\mathbf{y})$$

Viterbi Decoding

State inference problem

Determines the most likely joint hidden states assignment $\mathbf{Q} = \mathbf{x}$ for a given observed tree $\mathbf{y}$

Based on the recursive formulation

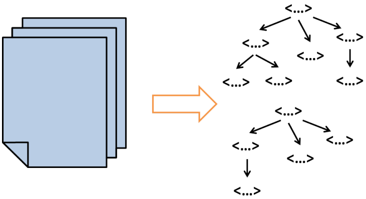
$$\max_{\mathbf{x}} P(\mathbf{y}, \mathbf{Q} = \mathbf{x}) =$$

$$= \max_{i} \left\{ \delta_u(i) \max_{\mathbf{x}_{1\setminus u}} \left\{ P(\mathbf{y}_{1\setminus \mathbf{CH}(u)}, \mathbf{Q}_{1\setminus u} = \mathbf{x}_{1\setminus u} | Q_u = i) \right\} \right\}$$

Exact Viterbi inference in BHTMM is $O(C^L)$ but can be approximated by an $O(LC)$ procedure

Application Example

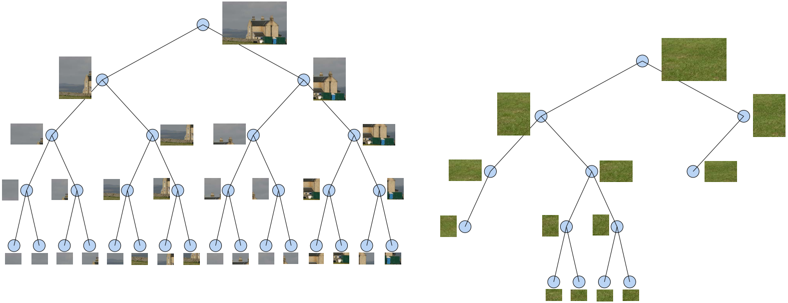
- Challenging tasks with tree-structured data arise in documental analysis
  - Parse trees
  - Tagging languages often provide structured document representation, e.g. XML
- INEX 2005 Competition
  - 9361 XML documents from 11 thematic categories
  - 366 XML labels and outdegree 32



Hidden States	BHTMM	THTMM
$C = 2$	32.20 (7.17)	34.28 (5.66)
$C = 4$	24.98 (5.89)	23.40 (4.89)
$C = 6$	22.91 (3.64)	30.50 (9.33)
$C = 8$	18.11 (3.02)	27.36 (6.53)
$C = 10$	18.93 (3.18)	28.92 (4.53)

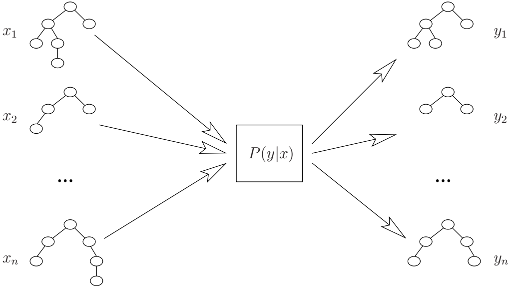
What about different document types?

- Image parse trees
  - Hierarchical segmentation of the image yields tree structure
  - Visual content in image segments determine node labels



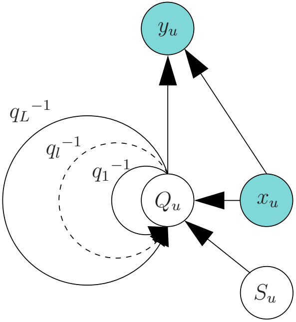
Amounts to learning an image grammar (Thesis Advertisement)

An Input-Driven Generative Model for Trees



- So far, we have focused on learning a generative process $P(\mathbf{x})$ for a tree $\mathbf{x}$
- What about learning an input-conditional generative process $P(\mathbf{y}|\mathbf{x})$ between Input-Output structures $(\mathbf{x}, \mathbf{y})$?
- Learn an isomorph transduction τ from $\mathbf{x}$ to $\mathbf{y}$

Input-Output BHTMM (IO-BTHMM)



- An input label x_u acting as observable context
- An output label y_u generated by the input-conditional emission $P(y_u|Q_u, x_u)$
- The input-conditional state transition is approximated by a finite mixture

$$P(Q_u|Q_{ch_1(u)}, \ldots, Q_{ch_L(u)}, x_u) = \sum_{l=1}^{L} P(S_u = l) P(Q_u|Q_{ch_l(u)}, x_u)$$

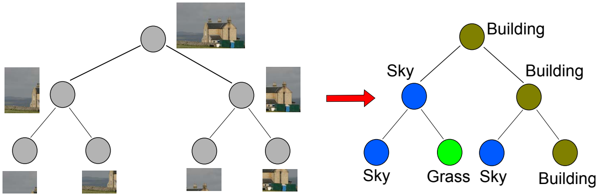
IO-BHTMM in INEX 2005

Classification is a tree-transduction task to a single node labeled with the class

Hidden States	IO-BHTMM root	IO-BHTMM vote	BHTMM	THTMM
INEX 2005				
$C = 2$	38.09 (1.24)	32.60 (2.24)	32.20 (7.17)	34.28 (5.66)
$C = 4$	27.09 (4.86)	19.66 (2.17)	24.98 (5.89)	23.40 (4.89)
$C = 6$	16.45 (2.84)	15.10 (2.77)	22.91 (3.64)	30.50 (9.33)
$C = 8$	16.43 (3.88)	13.31 (2.75)	18.11 (3.02)	27.36 (6.53)
$C = 10$	12.18 (3.57)	11.43 (2.93)	18.93 (3.18)	28.92 (4.53)

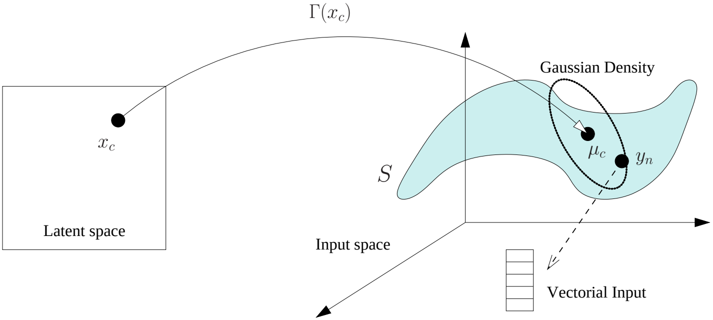
Learning Image Transductions

- Supervised hierarchical topic model for image processing
  - Learning transductions from segmentation trees to visual theme hierarchies
  - IO-isomorph transduction with multinomial labels
  - Thesis advertisement



Generative Topographic Mapping for Flat data

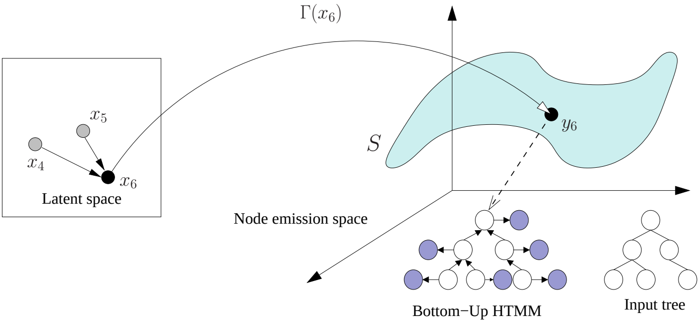
Visualization of high-dimensional vectors on a 2D map (latent space) that preserves vector similarities



Vectors generated by Gaussian distributions with means μ_c constrained on manifold S induced by the smooth mapping Γ

Generative Topographic Mapping for Trees (GTM-SD)

- Create a 2D map to visualize trees
    - Use SP-BHTMM to generate trees instead of Gaussians
- Since BHTMM is compositional we obtain a projection of all the substructures *for free*

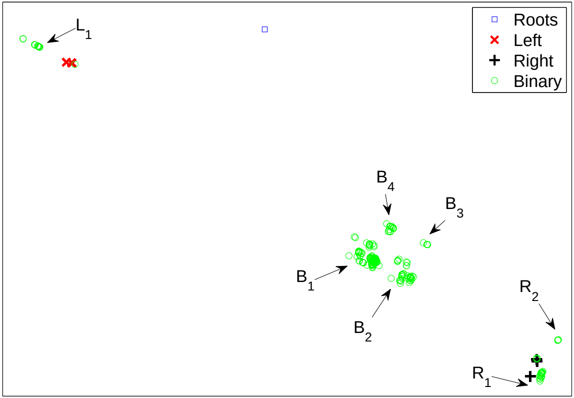


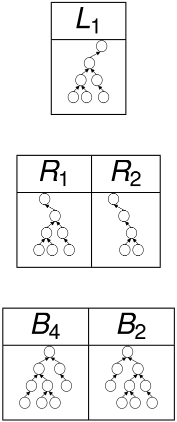
Tree Projection

- Train a constrained SP-BHTMM model using the EM algorithm
- Visualization of a tree $\mathbf{y}$ is based on projecting its root onto the lattice by using its hidden state assignment Q_1
  - Mean projection $\longrightarrow X_{mean}(\mathbf{y}) = \sum_{i=1}^{C} P(Q_1 = i|\mathbf{y})x_i$
  - Mode projection $\longrightarrow X_{mode}(\mathbf{y}) = \arg\max_{x_i} P(Q_1 = i|\mathbf{y})$
- Distribution $P(Q_1 = x_i|\mathbf{y})$ is obtained as a by-product of Upwards-Downwards algorithm
  - Alternatively, Viterbi inference can be used

Topographic Mapping with Structure Only

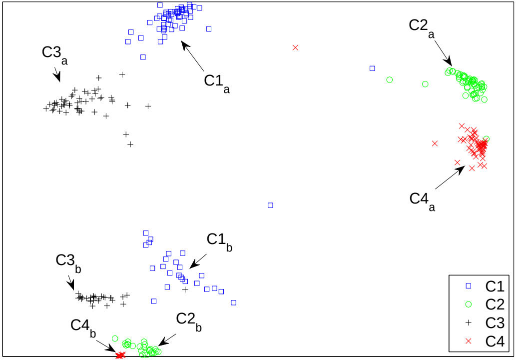
Left/Right sequences and binary trees with identical label for all nodes and trees



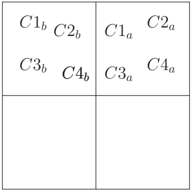


Four Gaussian

Complete binary trees from four 2-state TD-HTMMs



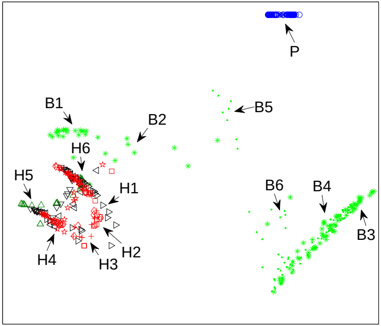
Root Position

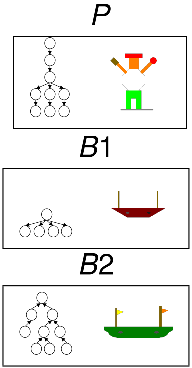


Quadrants

Policeman Dataset

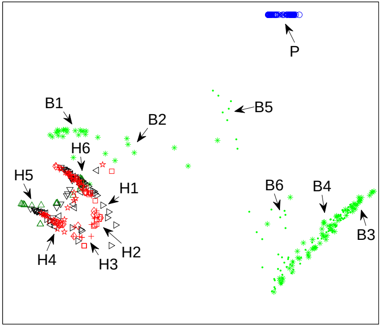
12 classes representing Policemen, Boat and House images

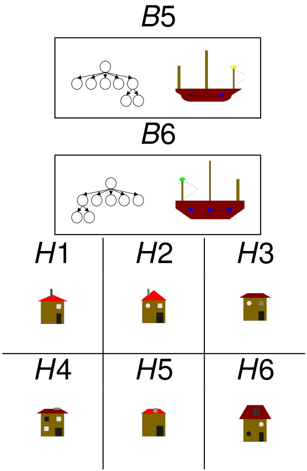




Policeman Dataset

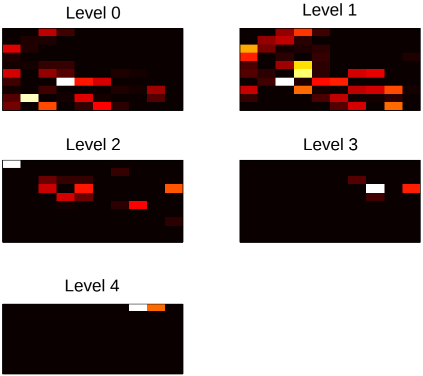
12 classes representing Policemen, Boat and House images





Sub-tree Projection

Compositionality allows topographic organization and visualization of all the substructures in the dataset



Take Home Messages

- Generative models provide an interesting approach to structured data
  - Can generate and explain data
  - Can learn transductions
  - Can be computationally expensive
  - Generative vs Discriminative
- Room for improvement on models and applications (a.k.a. thesis)
  - Structured image processing
  - Learning non-isomorphic transductions
  - Generative models for graphs
- Preview of upcoming lessons
  - Building blocks for generative kernels for trees
  - Discriminative approaches on top of generative models

Bibliography (I)

Hidden Tree Markov Models

- D. Bacciu, A. Micheli and A. Sperduti, "Compositional Generative Mapping for Tree-Structured Data - Part I: Bottom-Up Probabilistic Modeling of Trees", IEEE Transactions on Neural Networks and Learning Systems, vol. 23, no. 12, pp. 1987-2002, 2012
- P. Frasconi , M. Gori and A. Sperduti, "A general framework for adaptive processing in data structures", IEEE Transactions on Neural Networks, vol. 9, no. 5, pp.768-785, 1998
- M. Diligenti, P. Frasconi, M. Gori, "Hidden tree Markov models for document image classification", IEEE Transactions. Pattern Analysis and Machine Intelligence, Vol. 25, pp. 519-523, 2003

Bibliography (II)

Input/Output Generative Models

- D. Bacciu, A. Micheli and A. Sperduti, "An Input-Output Hidden Markov Model for Tree Transductions", Neurocomputing, Elsevier, Vol. 112, pp. 34-46, Jul, 2013
- Y. Bengio, P. Frasconi, "Input-Output HMMs for sequence processing", IEEE Transactions on Neural Networks, Vol. 7, pp. 1231-1249, 1996

Bibliography (III)

Topographic Mapping

- D. Bacciu, A. Micheli and A. Sperduti, "Compositional Generative Mapping for Tree-Structured Data - Part II: Topographic Projection Model", IEEE Transactions on Neural Networks and Learning Systems, vol. 24, no. 2, pp. 231-247, Feb 2013
- M. Hagenbuchner , A. Sperduti and A. Tsoi "A self-organizing map for adaptive processing of structured data", IEEE Transactions on Neural Networks, vol. 14, no. 3, pp. 491-505, 2003
- N. Gianniotis, P. Tino, "Visualization of tree-structured data through generative topographic mapping", IEEE Transactions on Neural Networks, vol. 19, pp. 1468-1493, 2008