

# Greedy Learning of Binary Latent Trees

Stefan Harmeling<sup>1</sup> and Chris Williams  
School of Informatics, University of Edinburgh

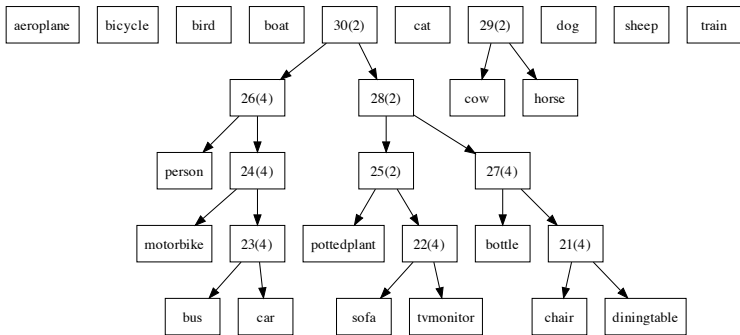
March 2010

---

<sup>1</sup>Now at Max Planck Institute for Biological Cybernetics, Tübingen

# The Goal: Learning Latent Trees

- ▶ Let  $\mathbf{x} = (x_1, \dots, x_D)^T$ . Model  $p(\mathbf{x})$  with the aid of latent variables
- ▶ Latent class model (LCM) has a single latent variable
- ▶ Latent tree (or hierarchical latent class, HLC) model has a tree structure, with visible variables as leaves
- ▶ Tree-structured network allows linear time inference
- ▶ Inspiration from parse-trees
- ▶ Zhang (2004), Zhang and Kočka (2004) search over HLCs, starting from LCM:  $O(D^3)$
- ▶ Our method: greedy, bottom-up determination of a tree/forest  $O(D^2)$
- ▶ Aim is to be fast, cf Bayesian MCMC over tree structures
- ▶ Note: Chow-Liu trees do not contain latent variables



- ▶ Incremental Learning of Binary Trees
- ▶ Runtime Complexity
- ▶ Agglomerative Hierarchical Clustering
- ▶ Related Work
- ▶ Experiments
- ▶ Conclusions

# Incremental Learning of Binary Trees

- ▶ Determining the *structure* of the latent tree: see BIN algorithm below
- ▶ Determining the cardinality of the latent variables and the CPTs given the structure. For each latent variable
  - ▶ Choose a range of cardinalities  $1, \dots, K_{max}$  and for each run EM (with restarts) to estimate a latent class model
  - ▶ Choose combination with the highest BIC score

- 1: **input:** a working set  $V$  of variables  $x_1, \dots, x_D$
- 2:  $G \leftarrow$  the graph with vertices  $V$  and no edges
- 3: calculate  $l(x_i; x_j)$  for all  $i \neq j$
- 4: **loop**
- 5:   pick pair  $(x_i, x_j)$  from  $V$  with maximum  $l(x_i; x_j)$
- 6:    $W \leftarrow \{x_i, x_j\}$  being the set of children
- 7:    $V \leftarrow V \setminus W$  /\* remove children from  $V$  \*/
- 8:    $z \leftarrow \text{LCM}(W)$  /\* find latent class model \*/
- 9:   **if**  $z$  has single state **then**
- 10:     **break** /\* outer loop \*/
- 11:   **end if**
- 12:   add vertex  $z$  to graph  $G$
- 13:   add edges from children  $W$  to  $z$  to graph  $G$
- 14:   **if**  $V$  is empty **then**
- 15:     **break** /\* outer loop \*/
- 16:   **end if**
- 17:   add latent variable  $z$  to set  $V$
- 18:   calculate  $l(x; z)$  for the new vertex in  $V$
- 19: **end loop**
- 20: refine the conditional probability tables using EM on structure  $G$
- 21: **output:** the graph  $G$  (being a forest)

- ▶ Note: MI for latent variables are computed using pseudo-frequencies based on belief propagation
- ▶ Algorithm could also be used to learn tree-structured Gaussian/linear models (i.e. non-recursive structural equation models)

# Motivation for selecting the pair with maximum MI

- ▶ Consider a distribution  $p(\mathbf{x})$  which is approximated by a distribution  $q(\mathbf{x})$ , where

$$q(\mathbf{x}) = p(x_i, x_j) \prod_{k \neq i, j} p(x_k) = \frac{p(x_i, x_j)}{p(x_i)p(x_j)} \prod_k p(x_k),$$

i.e.  $q(\mathbf{x})$  models the joint distribution of  $x_i$  and  $x_j$ , but only the marginal distributions of the other variables. Then

$$\begin{aligned} \text{KL}(p||q) &= \sum_{\mathbf{x}} p(\mathbf{x}) \log p(\mathbf{x}) - \sum_{\mathbf{x}} p(\mathbf{x}) \log q(\mathbf{x}) \\ &= -I(x_i, x_j) + \sum_{\mathbf{x}} p(\mathbf{x}) \log \frac{p(\mathbf{x})}{\prod_k p(x_k)}, \end{aligned}$$

- ▶ Thus in order to minimize the Kullback-Leibler divergence between  $p(\mathbf{x})$  and  $q(\mathbf{x})$  we should select the pair that has the highest mutual information.

# Runtime Complexity

$N$	number of data points
$D$	number of observed variables
$K$	maximal cardinality (actual)
$K_{\max}$	maximal cardinality (considered)
$I$	number of EM-iterations
$S$	number of EM-restarts

- ▶ Time complexity of BIN is  $O(D^2NK^2 + DSINK^2K_{\max})$
- ▶ Outer loop is executed at most  $D - 1$  times

# Agglomerative Hierarchical Clustering (Not)

- ▶ The greedy algorithm is reminiscent of Agglomerative Hierarchical Clustering (AHC). However, AHC forms the datapoints into a tree structure, not the variables
- ▶ Clustering of the variables only produces a tree structure, not a full graphical model
- ▶ Connolly (1993) builds structure by AHC on mutual information between variables. Then uses Fisher's (1987) COBWEB algorithm
- ▶ Kojadinovic (2004) builds structure by AHC on mutual information between variables. Does not construct a TSNB
- ▶ Wang et al (2008) also build structure by AHC on mutual information between variables. They determine the cardinality of all latent nodes globally

## Related Work

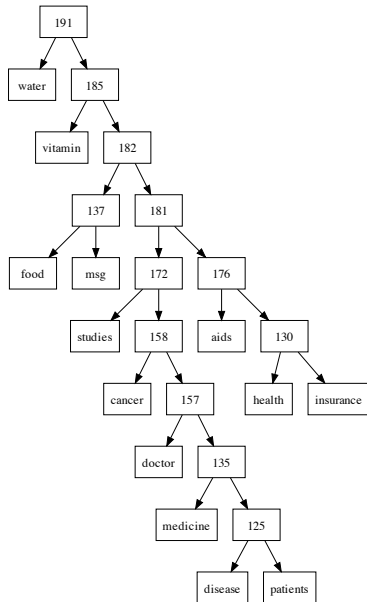
- ▶ Zhang (2004), Zhang and Kočka (2004)
- ▶ Pearl (1988) considers recovery of latent trees of binary and Gaussian variables if  $p(\mathbf{x})$  exactly decomposes. However, he does not address approximation of the joint, assumes exact statistics are available, and considers only *binary* discrete variables
- ▶ Hinton et al (2006): deep belief networks use a greedy layerwise learning procedure
- ▶ Branching tree models relating to clustering, e.g. Williams (2000), Neal (2003), Teh et al (2008), Kemp and Tenenbaum (2008), and phylogenetic trees (e.g. Felsenstein 2004). But in these models leaves are datapoints, not variables

- ▶ Probabilistic hierarchical clustering (Friedman, 2003; Heller and Ghahramani, 2005); neither of these methods constructs a generative model.
- ▶ H & G say: “[our model] is not in fact a hierarchical generative model of the data, but rather a hierarchical way of organizing nested clusters.”

- ▶ 20 newsgroups
- ▶ PASCAL Visual Object Classes (2007)
- ▶ COIL-42, COIL-86
- ▶ 10 small datasets
- ▶ Comparison with ZHANG algorithm by Zhang and Kočka (2004), implemented in JAVA

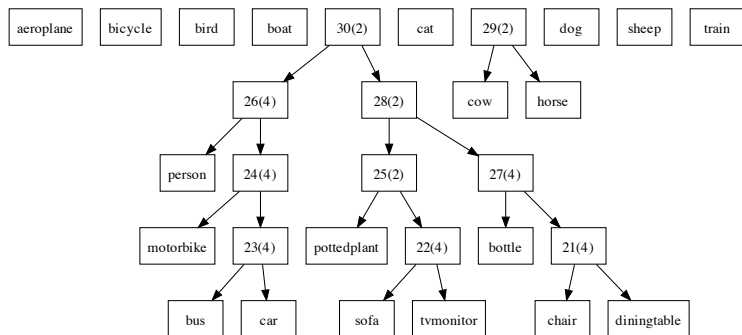
# 20 newsgroups

- ▶  $N = 16,242$ ,  $D = 100$  words (Roweis)
- ▶ Learned model can be interpreted as topics (cf Blei et al, 2003)
- ▶ Topics: medicine, sports, politics/religion, computer, spaceflight, car, others



# PASCAL VOC 2007 data

- ▶  $N = 20961$ ,  $D = 20$  objects
- ▶ Each variable can take on values  $0, \dots, 9$ , indicating where in the image each object is (0=absence)
- ▶ Note groupings of roadscenes and indoor objects



- ▶ COIL-86 data  $N = 5822$ ,  $D = 86$  concerning customers offered insurance (from COIL 2000 Challenge)
- ▶ COIL-42 prepared by Zhang and Kočka (2004)
- ▶ BIC score

	$D$	$N$	IND	LCM	BIN	ZHANG
COIL-42	42	5822	-79843.98	-67144.97	-53533.49	<b>-51465</b>
COIL-86	86	5822	-483255.54	-456899.96	<b>-375596.77</b>	—

- ▶ Running time (secs)

	$D$	$N$	IND	LCM	BIN	ZHANG
COIL-42	42	5822	0.04	<i>13155.06</i>	<b>487.73</b>	$\approx 435600$
COIL-86	86	5822	0.05	<b>13255.54</b>	<i>32260.01</i>	—

- ▶ BIC results from structure determination by hierarchical clustering are a bit worse than BIN, times are comparable

- ▶ 4 synthetic datasets, 4 used by Zhang (2004), plus HANNOVER-8 and CAR-EVALUATION
- ▶ BIN is competitive with ZHANG in terms of 10-fold cross-validated predictive log likelihood (CVPLL), while generally being faster
- ▶ CVPLL results for structure determination by hierarchical clustering (HC) are generally similar to BIN, with HC usually being faster.

# Conclusions

- ▶ Greedy algorithm for construction of latent tree model produces models that are generally competitive with ZHANG's algorithm, but with better scaling
- ▶ Latent tree models can be interpretable, and useful for data visualization/exploration
- ▶ We did consider e.g. non-binary trees, but preliminary experiments suggested that such approaches were not worth the extra computational cost
- ▶ MATLAB code is available

# References

- ▶ Connolly, D. (1993). Constructing Hidden Variables in Bayesian Networks via Conceptual Clustering. Proc. of the Tenth Int. Conf. on Machine Learning (pp. 6572).
- ▶ Kojadinovic, I. (2004). Agglomerative Hierarchical Clustering of Continuous Variables based on Mutual Information. Comp. Stat. and Data Analysis, 46, 269294.
- ▶ Wang, Y., Zhang, N., & Chen, T. (2008). Latent Tree Models and Approximate Inference in Bayesian Networks. Journal of Artificial Intelligence Research, 32, 879900.
- ▶ Zhang, N. L. (2004). Hierarchical Latent Class Models for Cluster Analysis. JMLR, 5, 697723.
- ▶ Zhang, N. L., & Kočka, T. (2004). Efficient Learning of Hierarchical Latent Class Models. Proc. of the 16th IEEE Int. Conf. on Tools with AI (ICTAI-2004).