Statistical Learning Theory with Chow-Liu Trees: Tree-aided Classifiers

Timo Koski

Umeå 2010-11-09

November 8, 2010

KTH Matematik

Introduction

A target class of probabilities $\mathcal{P}(\mathcal{X})$ on a finite but high dimensional discrete set $\mathcal{X}$ and concept class $\mathcal{P}(\mathcal{X};\mathcal{S})$ of probabilities on $\mathcal{X}$ depending on some approximating product structures $\mathcal{S}$. One fixes $\mathbf{p} \in \mathcal{P}(\mathcal{X})$ and samples N independent configurations in $\mathcal{X}$. The learning task is to find a distribution $\mathbf{p}_{\mathcal{S}}$ in $\mathcal{P}(\mathcal{X};\mathcal{S})$ such that the Kullback distance between the empirical distribution of the N samples and $\mathbf{p}$ is minimized. The question is not to find the true structure of $\mathbf{p}$, but to find the best distribution in $\mathcal{P}(\mathcal{X};\mathcal{S})$ that permits tractable inference, source coding and classification.

KTH Matematik

Product Approximation

P.M. Lewis II (1959):

> *A product approximation is defined to be an approximation to a higher order distribution made up of a product of several of its lower order component distributions such that the product is an extension of the lower order distributions.*

By extension Lewis means that the lower order component distributions can be obtained by marginalization from the product, and that the product is a probability distribution. A product of an arbitrary set of lower dimensional probability distributions will not satisfy these requirements.

Acknowledgements

This is joint work with Jukka Corander, Peter Dawyndt, Magnus Ekdahl, Mats Gyllenberg, John Noble e.t.c.. The slides to follow do not assign references on each topic to the prior sources.

KTH Matematik

Notations

X_i a generic random variable defined assuming values $x_i \in \mathcal{X}_i$, an alphabet with the cardinality $\mid \mathcal{X}_i \mid < \infty$. Let $\mathbf{X} = (X_i)_{i=1}^d$, i.e., $\mathbf{X}$ is d-dimensional ($d < \infty$) block of random variables with values in $\mathcal{X} = \times_{i=1}^d \mathcal{X}_i$. A *configuration* is $\mathbf{x} = (x_i)_{i=1}^d \in \mathcal{X}$. The probability of $\mathbf{X}$ in the configuration $\mathbf{x}$ is

$$p(\mathbf{x}) = P(\mathbf{X} = \mathbf{x}).$$

Then

$$\mathbf{p} = \{p(\mathbf{x})\}_{\mathbf{x} \in \mathcal{X}}$$

designates the probability distribution on $\mathcal{X}$ given by the probabilities $p(\mathbf{x})$.

KTH Matematik

Notations

$$\mathbf{I} = \{1, 2, \ldots, d\}.$$

For any $A \subseteq \mathbf{I}$ we have $\mathcal{X}_A = \times_{i \in A} \mathcal{X}_i$

$$\mathbf{x}_A = (x_i)_{i \in A} \in \mathcal{X}_A.$$

and the probability table

$$\mathbf{p}_A = \{p_A(\mathbf{x}_A)\}_{\mathbf{x}_A \in \mathcal{X}_A}.$$

Dependence Structure

$$\mathbf{I} = \{1, 2, \ldots, d\}.$$

Let $(A_i)_{i=1}^k$ be a partition of $\mathbf{I}$, and let $\mathcal{S}$ be a sequence of pairs of subsets of $\mathbf{I}$

$$\mathcal{S} = (A_i, B_i)_{i=1}^k \tag{1}$$

such that

$$B_1 = \emptyset, B_r \subset \cup_{i=1}^{r-1} A_i \subseteq \mathbf{I}, \quad r = 2, \ldots, k. \tag{2}$$

Then $\mathcal{S}$ is a *dependence structure*

Product Approximation & Dependence Structure

Let $\mathcal{S}$ be a dependence structure. The probability distribution defined by

$$p\left(\mathbf{x} \mid \mathcal{S}\right) = p_{A_1}\left(\mathbf{x}_{A_1}\right) \prod_{i=2}^{k} p\left(\mathbf{x}_{A_i} \mid \mathbf{x}_{B_i}\right), \mathbf{x} \in \mathcal{X}, \tag{3}$$

is called a *product approximation* of the distribution $\mathbf{p}$.

$$\mathbf{p}_{\mathcal{S}} = \left\{p\left(\mathbf{x} \mid \mathcal{S}\right)\right\}_{\mathbf{x} \in \mathcal{X}}. \tag{4}$$

KTH Matematik

Product Approximation in Supervised Bayesian Classification

C is an alphabet of class indices c. Then $(\mathbf{x}, c) \in \mathcal{X} \times C$. Let $\mathbf{e} = \{\mathbf{X} = \mathbf{x}\}$. Then the maximum a posterior estimate of c, or the supervised classification of $\mathbf{e}$, $\widehat{c}\,(\mathbf{e})$ is

$$\widehat{c}\,(\mathbf{e}) = \operatorname{argmax}_{c \in C} p\,(c \mid \mathbf{e}) = \operatorname{argmax}_{c \in C} p\,(\mathbf{e} \mid c)\, q\,(c)\,.$$

Then we approximate with a product approximation and get

$$\widehat{c}\,(\mathbf{e}; \mathcal{S}) = \operatorname{argmax}_{c \in C} p\,(\mathbf{e} \mid \mathcal{S}, c)\, q\,(c)\,.$$

M. Ekdahl & T. Koski: Bounds for the Loss in Probability of Correct Classification Under Model Based Approximation. *The Journal of Machine Learning Research*, 7, 2006.

KTH Matematik

Directed and Acyclic Graphs

Let $\mathcal{G} = (\mathbf{I}, E)$ be a directed and acyclic graph. The set of *nodes* is $\mathbf{I} = \{1, \ldots, d\}$ and the *edges* are ordered pairs of $\mathbf{I} \times \mathbf{I}$, i.e.,

$$E = \{(j, k) | j \in \mathbf{I}, k \in \mathbf{I}, j \neq k\}.$$

If $(j, k) \in E$, the node j is said to be a parent of the node k. We can always enumerate the nodes of a directed and acyclic graph so that the parents of the node i are included in $\{1, 2, \ldots, i-1\}$. The node 1 has no parents. We let $B_i \stackrel{\text{def}}{=}$ parents of i. Then

$$\mathcal{S} = (\{i\}, B_i)_{i=1}^{k} \tag{5}$$

is a dependence structure.

Bayesian Networks

$$p\left(\mathbf{x} \mid \mathcal{S}\right) = p_1\left(x_1\right) \prod_{i=2}^{d} p\left(x_i \mid \mathbf{x}_{B_{i-1}}\right), \mathbf{x} \in \mathcal{X}. \tag{6}$$

The pair consisting of a directed and acyclic graph and a probability distribution factorized as in (6) is known as a *Bayesian network* for the random variables $X_1, \ldots, X_d$.

Kullback-Leibler Distance

The *Kullback distance* $D(\mathbf{p} \parallel \mathbf{p}_{\mathcal{S}})$ between $\mathbf{p}$ and $\mathbf{p}_{\mathcal{S}}$ in (4) is defined by

$$D(\mathbf{p} \parallel \mathbf{p}_{\mathcal{S}}) \stackrel{\text{def}}{=} \sum_{\mathbf{x} \in \mathcal{X}} p(\mathbf{x}) \ln \frac{p(\mathbf{x})}{p(\mathbf{x} \mid \mathcal{S})}. \tag{7}$$

We have that

$$D(\mathbf{p} \parallel \mathbf{p}_{\mathcal{S}}) \geq 0.$$

Kullback-Leibler Distance

$$H(\mathbf{I}) + D(\mathbf{p} \parallel \mathbf{p}_{\mathcal{S}}) \leq E_{\mathbf{p}}\left[\lceil -\log_2 p_{\mathcal{S}}(\mathbf{X})\rceil\right] \leq H(\mathbf{I}) + D(\mathbf{p} \parallel \mathbf{p}_{\mathcal{S}}) + 1.$$

This shows the Kullback distance (in bits) as the expected source coding length redundancy, within one bit, when coding a configuration $\mathbf{x}$, generated by the distribution $\mathbf{p}$, using $\lceil -\log_2 p_{\mathcal{S}}(\mathbf{x})\rceil$ bits

Entropy and Mutual Information

$$H(A) \stackrel{\text{def}}{=} -\sum_{\mathbf{x}_A \in \mathcal{X}_A} p_A(\mathbf{x}_A) \ln p_A(\mathbf{x}_A). \tag{8}$$

$$\mathbf{I}(A,B) \stackrel{\text{def}}{=} \sum_{\mathbf{x}_{A\cup B} \in \mathcal{X}_{A\cup B}} p_{A\cup B}(\mathbf{x}_{A\cup B}) \ln \frac{p_{A\cup B}(\mathbf{x}_{A\cup B})}{p_A(\mathbf{x}_A)\, p_B(\mathbf{x}_B)}. \tag{9}$$

KTH Matematik

Lewis' Formula

$$D\left(\mathbf{p} \parallel \mathbf{p}_{\mathcal{S}}\right) = -H\left(\mathbf{I}\right) + \sum_{i=1}^{k} H\left(A_i\right) - \sum_{i=1}^{k} \mathbf{I}(A_i, B_i). \tag{10}$$

We observe here that $H\left(\mathbf{I}\right)$ is independent of the product approximation. Hence we maximize

$$Q\left(\mathcal{S}\right) = -\sum_{i=1}^{k} H\left(A_i\right) + \sum_{i=1}^{k} \mathbf{I}(A_i, B_i) \tag{11}$$

Chow-Liu Dependence Tree

$$Q(\mathcal{S}) = -\sum_{i=1}^{k} H(A_i) + \sum_{i=1}^{k} \mathbf{I}(A_i, B_i)$$

One way for fast and feasible maximization is to restrict the cardinality of B_is. With $|B_i| = 1$ this gives Chow-Liu dependence structures (trees). The problem is reduced to optimizing maximal mutual information and then to a undirected maximum spanning tree problem (MST).

Chow-Liu Dependence Structure

Let $(i_r)_{r=1}^d$ be an arbitrary permutation of $\mathbf{I} = \{1, 2, \ldots, d\}$. The singleton sets $A_r = \{i_r\}$, $r = 1, \ldots, d$, are a partition of $\mathbf{I}$. Let σ be a sequence of pairs of singletons of $\mathbf{I}$

$$\sigma = (i_r, j_r)_{r=1}^d \,, \tag{12}$$

where

$$j_1 = \emptyset, j_r \in \{i_1, \ldots, i_{r-1}\} \subseteq \mathbf{I}, \quad r = 2, \ldots, d. \tag{13}$$

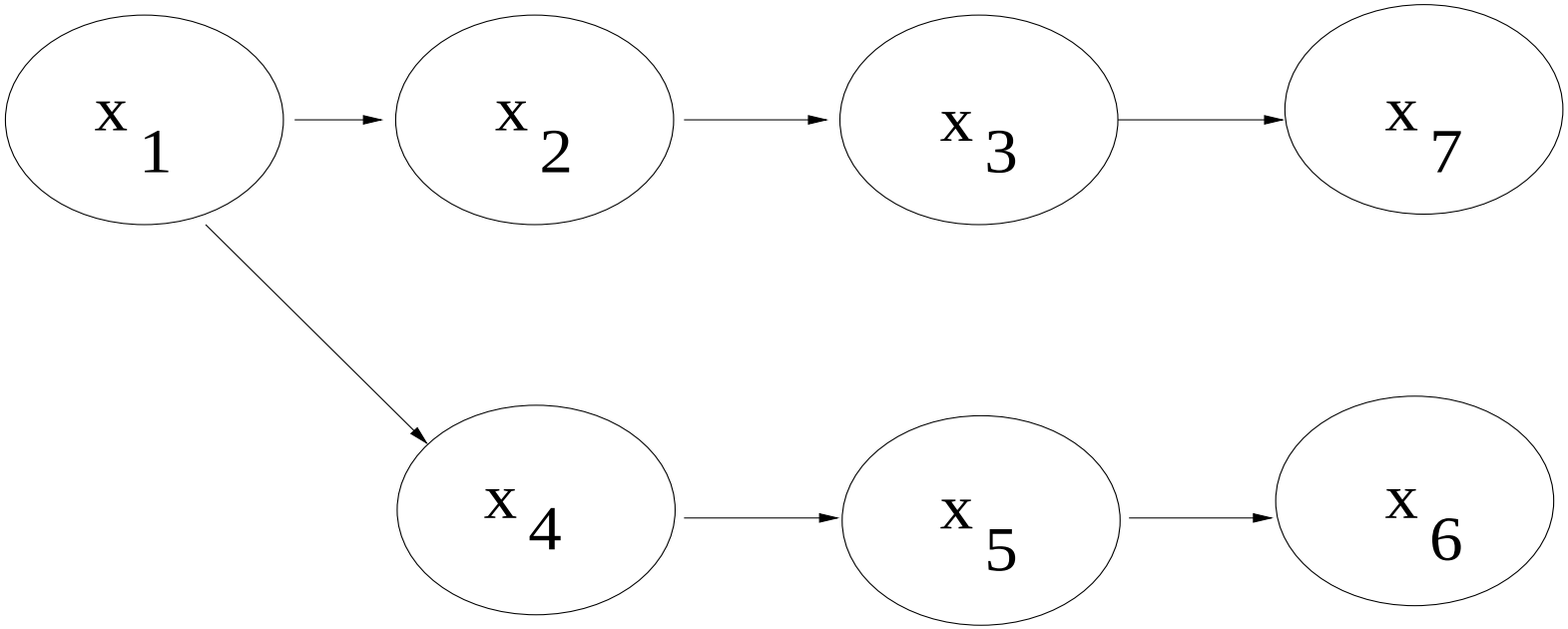
Then σ is a *Chow-Liu dependence structure*.

Chow-Liu dependence structure

$$
\begin{aligned}
p(\mathbf{x} \mid \sigma) &= p_{i_1}(x_{i_1}) \prod_{r=2}^{d} p(x_{i_r} \mid x_{j_r}) \\
&= \prod_{r=1}^{d} p_{i_r}(x_{i_r}) \prod_{r=2}^{d} \frac{p_{i_r \cup j_r}(x_{i_r}, x_{j_r})}{p_{i_r}(x_{i_r})\, p_{j_r}(x_{j_r})}, \quad \mathbf{x} = (x_i)_{i=1}^{d} \in \mathcal{X}.
\end{aligned}
\tag{14}
$$

KTH Matematik

Chow-Liu dependence structure



Lewis' formula under the Chow-Liu dependence structure

$$Q(\sigma) = -\sum_{r=1}^{d} H(i_r) + \sum_{r=1}^{d} \mathbf{I}(i_r, j_r).$$

The first term $\sum_{r=1}^{d} H(i_r)$ is independent of σ, hence we need only to maximize the term $\sum_{r=1}^{d} \mathbf{I}(i_r, j_r)$.

Finding the optimal Chow-Liu Dependence Structure

The task of maximizing the term $\sum_{r=1}^{d} \mathbf{I}(i_r, j_r)$ is that of finding the maximum weight spanning tree in a weighted graph.
There are well known standard algorithms for finding the maximum weight spanning tree, e.g., the *Kruskal algorithm* and/or the *Prim algorithm*

Learning of a Chow-Liu Dependence Structure from Data

Data $\mathbf{D}$, which consists a sample of N independent observations (passively observed instantiations of configurations in $\mathcal{X}$) of $\mathbf{X} = (X_i)_{i=1}^d$. We set

$$\mathbf{D} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)}\},$$

where

$$\mathbf{x}^{(n)} = \left(x_i^{(n)}\right)_{i=1}^d \in \mathcal{X}.$$

The empirical distribution

We set for any $\mathbf{x} \in \mathcal{X}$

$$I_{\mathbf{x}}\left(\mathbf{x}^{(n)}\right) = \begin{cases} 1 & \text{if } \mathbf{x} = \mathbf{x}^{(n)} \\ 0 & \text{otherwise.} \end{cases}$$

Then we define the empirical probability on $\mathcal{X}$ based on D as

$$\widehat{p}_N(\mathbf{x}) = \frac{1}{N} \sum_{n=1}^{N} I_{\mathbf{x}}\left(\mathbf{x}^{(n)}\right). \tag{15}$$

The maximum likelihood

$$D\left(\widehat{p}_N \parallel \mathbf{p}\right) = -\widehat{H}_N\left(\mathbf{l}\right) - \sum_{\mathbf{x}\in\mathcal{X}} \widehat{p}_N\left(\mathbf{x}\right) \ln p\left(\mathbf{x}\right).$$

where

$$\sum_{\mathbf{x}\in\mathcal{X}} \widehat{p}_N\left(\mathbf{x}\right) \ln p\left(\mathbf{x}\right) = \frac{1}{N}\sum_{n=1}^{N} \ln p\left(\mathbf{x}^{(n)}\right)$$

Learning of a Chow-Liu Dependence Structure from Data: Plug-in

For any $i \in \mathbf{I}$ and $\xi \in \mathcal{X}_i$

$$\widehat{p}_i(\xi) = \frac{1}{N} \sum_{n=1}^{N} I_{\xi,i}\left(\mathbf{x}^{(n)}\right), \tag{16}$$

is the relative frequency of $\xi \in \mathcal{X}_i$ in our data set $\mathbf{D}$, or, the *plug-in estimate* of p_i.

$$\widehat{p}_{ij}(\xi, \eta) = \frac{1}{N} \sum_{n=1}^{N} I_{\xi,i}\left(\mathbf{x}^{(n)}\right) I_{\eta,j}\left(\mathbf{x}^{(n)}\right), \tag{17}$$

which is the relative frequency of the pair $(\xi, \eta) \in \mathcal{X}_i \times \mathcal{X}_j$ in $\mathbf{D}$.

KTH Matematik

The loglikelihood

$$l(\sigma, \mathcal{P}) = \frac{1}{N} \sum_{n=1}^{N} \ln p\left(\mathbf{x}^{(n)} \mid \sigma, \mathcal{P}\right).$$

$$l(\sigma, \mathcal{P}) = \sum_{\xi \in \mathcal{X}_{i_1}} \widehat{p}_{i_1}(\xi) \ln p_{i_1}(\xi)$$

$$+ \sum_{r=2}^{d} \sum_{\eta \in \mathcal{X}_{j_r}} \widehat{p}_{j_r}(\eta) \sum_{\xi \in \mathcal{X}_{i_r}} \frac{\widehat{p}_{i_r j_r}(\xi, \eta)}{\widehat{p}_{j_r}(\eta)} \ln p_{i_r | j_r}(\xi \mid \eta),$$

Plug-in Estimate of the Mutual Information

$$I\left(\sigma, \mathcal{P}^{\mathrm{ML}}\right) = \sum_{r=1}^{d} \sum_{\xi \in \mathcal{X}_{i_r}} \widehat{p}_{i_r}\left(\xi\right) \ln \widehat{p}_{i_r}\left(\xi\right) + \sum_{r=2}^{d} \widehat{\mathbf{I}}(i_r, j_r), \tag{18}$$

where we have introduced the plug-in estimate of mutual information

$$\widehat{\mathbf{I}}(i,j) = \sum_{x_i \in \mathcal{X}_i} \sum_{x_j \in \mathcal{X}_j} \widehat{p}_{i \cup j}\left(x_i, x_j\right) \ln \frac{\widehat{p}_{i \cup j}\left(x_i, x_j\right)}{\widehat{p}_i\left(x_i\right) \widehat{p}_j\left(x_j\right)}.$$

The ML-Estimate of a Chow -Liu Tree

We find the maximum likelihood estimate σ^{ML} of the structure σ by

$$\sigma^{\mathrm{ML}} = \mathrm{argmax}_{\sigma} \left\{ \sum_{r=2}^{d} \widehat{\mathbf{I}}(i_r, j_r) \right\}. \tag{19}$$

The number of trees with d nodes is finite. Hence in principle we could find σ^{ML} by exhaustive search and evaluation of $I\left(\sigma, \mathcal{P}^{\mathrm{ML}}\right)$. Nevertheless, since the number of spanning trees with d nodes is d^{d-2} (Cayley$'$s formula), exhaustive search is infeasible in practice. But the Kruskal algorithm finds σ^{ML}.

Error Rate for The ML-Estimate of a Chow -Liu Tree

$$\mathbf{I}^{(o)}(e) > \mathbf{I}^{(o)}(e'). \tag{20}$$

The *crossover event* (w.r.t. 20) is

$$\mathcal{C}_{e,e'} = \left\{ \widehat{\mathbf{I}}(e') \geq \widehat{\mathbf{I}}(e) \right\}. \tag{21}$$

Error Rate for The ML-Estimate of a Chow -Liu Tree

$$J_{e,e'} = \min_{Q \in \mathcal{P}(\mathcal{X}^4) \cap \Lambda} D\left(Q \parallel \mathbf{p}_{e,e'}\right), \tag{22}$$

where the constraint set is

$$\Lambda = \{Q \in \mathcal{P}\left(\mathcal{X}^4\right) \mid \mathbf{I}(Q_e) = \mathbf{I}(Q_{e'})\}, \tag{23}$$

Error Rate for The ML-Estimate of a Chow -Liu Tree

For each edge e' not in $\sigma^{(o)}$, its *dominant replacement edge* $r(e') \in \sigma^{(o)}$ is defined as the edge in the unique path along $\sigma^{(o)}$ connecting the nodes in e' with the minimum crossover rate

$$r(e') \stackrel{\text{def}}{=} \arg \min_{e \in \mathrm{Path}\left(e', \sigma^{(o)}\right)} J_{e,e'}, \tag{24}$$

Error Rate for The ML-Estimate of a Chow -Liu Tree

the maximum likelihood tree estimation error exponent is given by

$$K_{\mathbf{p}^{(o)}} = J_{r(e^*),e^*} = \min_{e' \notin \mathcal{T}_d^{(o)}} \min_{e \in \mathrm{Path}\left(e',\sigma^{(o)}\right)} J_{e,e'}, \tag{25}$$

where $r\left(e^*\right)$ is the dominant replacement edge, defined in (24), associated to $e^* \in \sigma^{(o)}$ and e^* is the optimizing non-neighbor node pair

$$e^* = \arg \min_{e' \notin \sigma^{(o)}} J_{r(e'),e'}. \tag{26}$$

Error Rate for The ML-Estimate of a Chow -Liu Tree

The dominant error tree $\mathcal{T}_d^* = (\mathbf{I}, \sigma^*)$ has the edge set

$$\sigma^* = \sigma^{(o)} \cup \{e^*\} \setminus \{r(e^*)\}. \tag{27}$$

Gamla KTH

KTH Matematik

Stochastic Complexity

$$\mathbf{p}_\theta^N = \left\{ \prod_{i=1}^{N} p_\theta\left(\mathbf{x}^{(i)}\right) \right\}_{\mathbf{x}^{(1)}\in\mathcal{X},\ldots,\mathbf{x}^{(N)}\in\mathcal{X}}.$$

$$\mathbf{q} \in \mathcal{P}\left(\mathcal{X}^N\right).$$

$$D\left(\mathbf{p}_\theta^N \parallel \mathbf{q}\right).$$

Let now $w(\theta)$ be a prior probability density on Θ.

$$R_w(\mathbf{q}) \stackrel{\text{def}}{=} \int_\Theta D\left(\mathbf{p}_\theta^N \parallel \mathbf{q}\right) dw(\theta). \tag{28}$$

We shall now find the $\mathbf{q}$ minimizing $R_w(\mathbf{q})$.

Stochastic Complexity

$$\mathbf{m}^* = \arg \min_{\mathbf{q} \in \mathcal{P}(\mathcal{X}^N)} R_w(\mathbf{q}). \tag{29}$$

Then

$$m^*\left(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\right) = \int_\Theta \prod_{i=1}^N p_\theta\left(\mathbf{x}^{(i)}\right) dw(\theta). \tag{30}$$

In fact

$$\min_{\mathbf{q}} \max_{\Theta} D\left(\mathbf{p}_\theta^N \parallel \mathbf{q}\right) = \max_w \int_\Theta D\left(\mathbf{p}_\theta^N \parallel \mathbf{m}^*\right) dw(\theta).$$

Furthermore, the least favorable w can often be taken as a *Jeffrey's prior*.

Stochastic Complexity

The distribution $\mathbf{m}^*$ is called the *predictive distribution* of $\mathbf{X}_1, \ldots, \mathbf{X}_N$. Furthermore, if $X^N = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$,

$$SC\left(X^N; \mathcal{M}\right) \stackrel{\text{def}}{=} -\ln m^*\left(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\right)$$

used to be called the *stochastic complexity*' of the family of models

$$\mathcal{M} = \{\mathbf{p}_\theta \mid \theta \in \Theta\}.$$

This does not necessarily presuppose an underlying true distribution in $\mathcal{M}$ or elsewhere.

Stochastic Complexity

Stochastic complexity can be understood as a generalization of Shannon's information, $-\log \prod_{i=1}^{N} p_\theta\left(\mathbf{x}^{(i)}\right)$, which depends on a single θ, whereas $SC\left(X^N; \mathcal{M}\right)$ is defined with respect to the whole $\mathcal{M}$. Rissanen has shown, that SC plays an essential role in statistical learning, as SC is a tight lower bound on the total predictive code length. The best model in $\mathcal{M}$ explaining X^N is the one which can best compress the data X^N adding to the length the bits needed for the compression of the model.

KTH Matematik

Chow Representation of Likelihood

$\mathcal{X}_i$ at each node of the tree, i.e., $x_i \in \{0,1\}$. B^d is the binary hypercube and let $\mathbf{x}$ denote its elements with d binary components, i.e.,

$$\mathbf{x} \in B^d = \left\{ \mathbf{x} \mid \mathbf{x} = (x_i)_{i=1}^d, x_i \in \{0,1\} \right\}.$$

$$p(\mathbf{x}) = p_1(x_1) \cdot p(x_2|x_{j_2}) \ldots p\left(x_{d-1}|x_{j_{d-1}}\right) \cdot p(x_d|x_{j_d}). \qquad (31)$$

Chow Representation of Likelihood

The conditional probabilities $p\left(x_i|x_{j_i}\right)$ are written as

$$p\left(x_i|x_{j_i}\right) = \left(\theta_i^{x_i}\left(1-\theta_i\right)^{(1-x_i)}\right)^{x_{j_i}} \cdot \left(\phi_i^{x_i}\left(1-\phi_i\right)^{(1-x_i)}\right)^{1-x_{j_i}}, \quad (32)$$

where for $i = 2, \ldots, d$ we have the (unknown) parameters

$$\theta_i = p\left(x_i = 1|x_{j_i} = 1\right), \quad (33)$$

and

$$\phi_i = p\left(x_i = 1|x_{j_i} = 0\right), \quad (34)$$

and

$$p\left(x_1\right) = \theta_1^{x_1}\left(1-\theta_1\right)^{(1-x_1)}.$$

Chow Representation of Likelihood

We introduce

$$p_{\theta_i,\phi_i}\left(x_i|x_{j_i}\right) \equiv p\left(x_i|x_{j_i}\right) \tag{35}$$

and

$$p_{\theta_1}\left(x_1\right) \equiv p\left(x_1\right)$$

We set from (33) and (34)

$$\underline{\theta} = \left(\theta_1, \ldots, \theta_d\right), \quad \underline{\phi} = \left(\phi_2, \ldots, \phi_d\right).$$

Chow Representation of Likelihood

The probability of the configuration $\mathbf{x}$ given the tree $\mathcal{T}_d$ and the parameters $\underline{\theta}$ and $\underline{\phi}$ is now represented as

$$p_{\underline{\theta},\underline{\phi}}(\mathbf{x}) = p_{\theta_1}(x_1) \cdot \prod_{i=2}^{d} p_{\theta_i,\phi_i}(x_i|x_{j_i}).$$

Chow Representation of Likelihood

The *Chow representation*

$$\prod_{l=1}^{N} p_{\underline{\theta},\underline{\phi}}(\mathbf{x}^{(l)}) =$$

$$\theta_1^{n_1}(1-\theta_1)^{N-n_1}\prod_{i=2}^{d}\theta_i^{n_i(1,1)}(1-\theta_i)^{n_i(0,1)}\cdot\phi_i^{n_i(1,0)}(1-\phi_i)^{n_i(0,0)},$$

Chow Representation of Likelihood

where, as all $\mathbf{x}$ are instantiations of the same dependence tree,

$$n_i(1,1) = \sum_{l=1}^{N} x_i^{(l)} x_{j_i}^{(l)}, \quad n_i(1,0) = \sum_{l=1}^{N} x_i^{(l)} \left(1 - x_{j_i}^{(l)}\right),$$

$$n_i(0,1) = \sum_{l=1}^{N} \left(1 - x_i^{(l)}\right) x_{j_i}^{(l)}, \quad n_i(0,0) = \sum_{l=1}^{N} \left(1 - x_i^{(l)}\right) \left(1 - x_{j_i}^{(l)}\right),$$

for $i = 2, \ldots, d$, and

$$n_1 = \sum_{l=1}^{N} x_1^{(l)}.$$

$n_i(1,1)$ counts the number of times we have simultaneously $x_i^{(l)} = 1$ and $x_{j_i}^{(l)} = 1$ in $\mathbf{D} = \left\{\mathbf{x}^{(l)}\right\}_{l=1}^{N}$ and e.t.c.. We set also

$$n_i(1), n_{j_i}(1), n_i(0), n_{j_i}(0)$$

as the count of ones (1) in $\mathbf{D}$ at position i e.t.c..

Chow Representation of Likelihood

Let next Θ and Φ denote respective copies of the d-fold and $d-1$-fold product of the unit interval, then $\underline{\theta} \in \Theta$ and $\underline{\phi} \in \Phi$. Given a prior probability density $w\left(\underline{\theta}, \underline{\phi}\right)$ on $\Theta \times \Phi$, we obtain the predictive likelihood corresponding to the Chow expansion as

$$m\left(\mathbf{D}|\mathcal{T}_d\right) = \int_{\Theta} \int_{\Phi} \prod_{l=1}^{N} p_{\underline{\theta},\underline{\phi}}\left(\mathbf{x}^{(l)}\right) w\left(\underline{\theta}, \underline{\phi}\right) d\underline{\theta} d\underline{\phi}. \tag{36}$$

The predictive likelihood corresponding to the Chow expansion

- We take as $w(\theta)$ the Jeffreys' prior.
- Then $w(\theta)$ is a product of Beta densities with parameter $\alpha = 1/2$.

The predictive likelihood

Theorem

If $m(D|\mathcal{T}_d)$ is evaluated assuming both local meta independence and Jeffreys' prior or the Beta distribution $\mathrm{Be}(1/2, 1/2)$ for each of the parameters, then

$$
\begin{aligned}
-\log m(\mathbf{D}|\mathcal{T}_d) = \log \frac{t!}{\Gamma(n_1 + 1/2) \cdot \Gamma(N - n_1 + 1/2)} \\
+ N \cdot \sum_{i=2}^{d} \mathbf{h}(n_i(1)/t) - N \cdot \sum_{i=2}^{d} \widehat{\mathbf{I}}_{i,j_i} \\
+ \frac{1}{2} \sum_{i=2}^{d} \log(n_{j_i}(1)) + \log(n_{j_i}(0)) + C, \qquad (37)
\end{aligned}
$$

where $\mathbf{h}$ is the binary entropy function, and $\widehat{\mathbf{I}}_{i,j_i}$ is the empirical mutual information on the edge (i, j_i) in $\mathcal{T}_d$ and C is bounded in N.

Classification

- $$\mathbf{D} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)}\},$$
- By a classification of the set $\mathbf{D}$ we mean the partition of D into k disjoint sets (classes) $c_j \subseteq D$, $j = 1, \ldots, k$.
- If the classes c_j have a rule of membership known in advance, we a talk about <u>supervised</u> classification.
- If the rule of membership is also found from data, we talk about <u>unsupervised</u> classification.
- The number of classes, k, need *not* be given in advance but can also learned from data in an unsupervised classification mode.

KTH Matematik

Basic Viewpoints

- A classification is viewed as a model of the data.
- The model is represented by class-conditional probability distributions factorized along Chow-Liu trees (tree-aided classification).
- The classifiers are trained by minimization of stochastic complexity (SC), which also selects the number of classes.

KTH Matematik

Aided by Trees

- Finding of the tree structure is part of the learning problem.
- In our implemented version we are searching for the structure in terms of dependencies in a tree (a DAG with unique path between any two nodes).
- The incorporation of dependencies is expected to improve the Bayesian classifier performance vs. that of the naive Bayesian classifier.

KTH Matematik

Indicators for Classification

We define for any given class c_j the class membership indicators

$$u_j^{(l)} = 1 \text{ if } \mathbf{x}^{(l)} \in c_j$$

and

$$u_j^{(l)} = 0 \text{ otherwise,}$$

and incorporate these in the matrix

$$U^t = \left\{ u_j^{(l)} \right\}_{l=1,j=1}^{N,k}.$$

Stochastic Complexity given a Classification

$$SC_k\left(\mathbf{D}|U^N,\mathcal{T}_d\right) = -\log\prod_{j=1}^{k}\int_{\mathbf{\Theta}_j}\prod_{l=1}^{N}\left[p(\mathbf{x}^{(l)}|\theta_j,\mathcal{T}_{d,j})w(\theta_j)\right]^{u_j^{(l)}}d\theta_j$$

We could think of finding a classification by maximizing this as function of U^t for every k. However, prevalence is needed for a complete account of SC.

Prevalence

- The prevalence of class j is $\lambda_j = P_\Lambda(u_j^{(l)} = 1)$, and $\{\lambda_j\}_{1 \geq j \geq n}$ is a probability distribution over j.
- We write

$$P(U^N) = \int_\Lambda \prod_{j=1}^{k} \lambda_j^{t_j} \psi(\lambda) d\lambda$$

where t_j is the number of items in c_j.

Prevalence

$$P(U^N) = \int_\Lambda \prod_{j=1}^{k} \lambda_j^{t_j} \psi(\lambda) d\lambda$$

We take the prior $\psi(\lambda)$ to be Jeffreys' prior (= a Dirichlet density with hyperparameters =1/2) and then

$$-\log P(U^N) = \log \frac{\Gamma\left(\sum_{j=1}^{k} t_j + \frac{1}{2}\right)}{\Gamma\left(\frac{k}{2}\right)} \prod_{j=1}^{k} \frac{\Gamma\left(\frac{1}{2}\right)}{\Gamma\left(t_j + \frac{1}{2}\right)}$$

where t_j is the number of items in c_j, and Γ is the Euler gamma function.

KTH Matematik

Stochastic Complexity of the Clustering

$$SC_k\left(\mathbf{D}, U^t|\mathcal{T}_d\right) = -\log P(U^N) - \log P\left(\mathbf{D}|\mathcal{T}_d, U^N\right)$$

The interpretation is description length:

- $-\log P\left(\mathbf{D}|\mathcal{T}_d, U^N\right)$ is the number of bits needed to describe the data vectors as members of the classes.
- $-\log P(U^N)$ is the number of bits needed to describe the classes.

KTH Matematik

Stochastic Complexity and Mutual Information

By the theorem above

$$-\frac{1}{N}\log P(\mathbf{D}|U^N,\mathcal{T}_d) =$$

$$-\sum_{j=1}^{k}\sum_{i=1}^{d}\left[\sum_{a=0}^{1}\sum_{b=0}^{1}\hat{P}_{X_i,X_{\Pi_i}}(a,b)\log\frac{\hat{P}_{X_i,X_{\Pi_{ij}}}(a,b)}{\hat{P}_{X_i}(a)\,\hat{P}_{X_{\Pi_{ij}}}(b)}\right.$$

$$\left.-\frac{1}{2N}\left[\lg\left(n_{j_i}(1)\right)+\lg\left(n_{j_i}(0)\right)\right]\right]+c$$

Here $\hat{P}_{X_i,X_{\Pi_{ij}}}(a,b)$ is the empirical relative frequency of the pair a,b of child and parent states (depends on the class c_j and its structure of $\mathcal{T}_{d,j}$).

Stochastic Complexity and Mutual Information

$$-\sum_{j=1}^{k}\sum_{i=1}^{d}\left[\sum_{a=0}^{1}\sum_{b=0}^{1}\hat{P}_{X_i,X_{j_i}}(a,b)\log\frac{\hat{P}_{X_i,X_{j_i}}(a,b)}{\hat{P}_{X_i}(a)\,\hat{P}_{X_{j_i}}(b)}\right.$$
$$\left.-\frac{1}{2N}\left[\lg\left(n_{j_i}(1)\right)+\lg\left(n_{j_i}(0)\right)\right]\right]+c$$

when N is large. c does not depend on the graph structure (or topology). $n_{j_i}(0)$ is the number of times we see 0 as the parent state of node i in class j.

Stochastic Complexity and Mutual Information

$$H\left(\mathcal{T}_{d,j}, \mathbf{D}\right) :=$$

$$-\sum_{i=1}^{d}\sum_{a=0}^{1}\sum_{b=0}^{1}\hat{P}_{X_i,X_{j_i}}\left(a,b\right)\log\frac{\hat{P}_{X_i,X_{j_i}}\left(a,b\right)}{\hat{P}_{X_i}\left(a\right)\hat{P}_{X_{j_i}}\left(b\right)}$$

Here $\hat{P}_{X_i,X_{j_i}}\left(a,b\right)$ is the empirical relative frequency of the pair a, b of child and parent states (depends on the class c_j and its structure of $\mathcal{T}_{d,j}$).

Mutual Information

$$H\left(\mathcal{T}_{d,j}, D\right) :=$$

$$-\sum_{i=0}^{d}\sum_{a=0}^{1}\sum_{b=0}^{1}\hat{P}_{X_i,X_{j_i}}\left(a,b\right)\log\frac{\hat{P}_{X_i,X_{j_i}}\left(a,b\right)}{\hat{P}_{X_i}\left(a\right)\hat{P}_{X_{j_i}}\left(b\right)}.$$

is (essentially, we have penalty terms, too) to be maximized ! This defines a minimum spanning tree problem.

Minimum Spanning Tree

We will solve MST on a complete graph with a version of Prim's algorithm, since it can be implemented with reasonable efficiency, i.e., $\mathcal{O}(|E|)$ with fewer cache misses (more efficient memory access) than other $\mathcal{O}(|E|)$ algorithms such as Borukva and Chazelle algorithms.

The Clustering Algorithm (1)

for k=k_{min} to k_{max}

set w=0 and store a random $U_{(w)}$ of size k.

while $\left(U_{(w)} \neq U_{(w-1)} \| w \neq 0\right)$

- modify pairwise SC cache and where necessary to apply changes between $U_{(w)}$ and $U_{(w-1)}$ in parallel
- find $\mathcal{T}_{(w)}$ using SCApprox() in parallel

KTH Matematik

The Clustering Algorithm (2)

- given $\mathcal{T}_{(w)}$ determine $U_{(w+1)}$ by iteratively $\forall l$ by first removing x_l from its cluster and then reinserting it where it reduces SC the most.

  if then (k $= k_{\min} \| \mathrm{SC}\left[U_{(w)}\right] < \mathrm{SC}\ [U_{\max}])$ then $U_{\max} \leftarrow U_{(w)}$

Unsupervised classification

Unsupervised classification of a set of observed data is understood as learning a postulated hidden structure underlying the

Adansonia digitata

observations.

Application to microbial classification (1)

Material:

- $D = \left\{\mathbf{x}^{(l)}\right\}_{l=1}^{5313}$ is a set of feature vectors (outcomes of biochemical tests) of *Enterobacteriaceae* with $d = 47$
- collected and analyzed by the group of J.J. Farmer (1985), and with SC and modeling by M. Gyllenberg et.al. .

Application to microbial classification (1): comparison of SC for various classifications

		SC	# classes
1	Farmer	74348.10	104
2	baps	74070.89	117
3	Chow-Liu (20 trials/cluster size)	71347.22	41
4	Binclass (50 trials/cluster size)	71341.46	62
5	Binclass $\longrightarrow$ Chow-Liu trees	70820.74	62

KTH Matematik

Application (1): comparison of SC for various classifications

In the preceding the SC-values were computed using

$$SC_k\left(\mathbf{D}, U^N|\mathbf{G}\right) = -\log P(U^N) - \log P\left(\mathbf{D}|\mathcal{T}_d, U^N\right)$$

i.e, the dimension is not binary information units per strain, and by estimating a Chow-Liu tree for all classes in all classifications.

KTH Matematik

Application (1): comparison of SC for various classifications

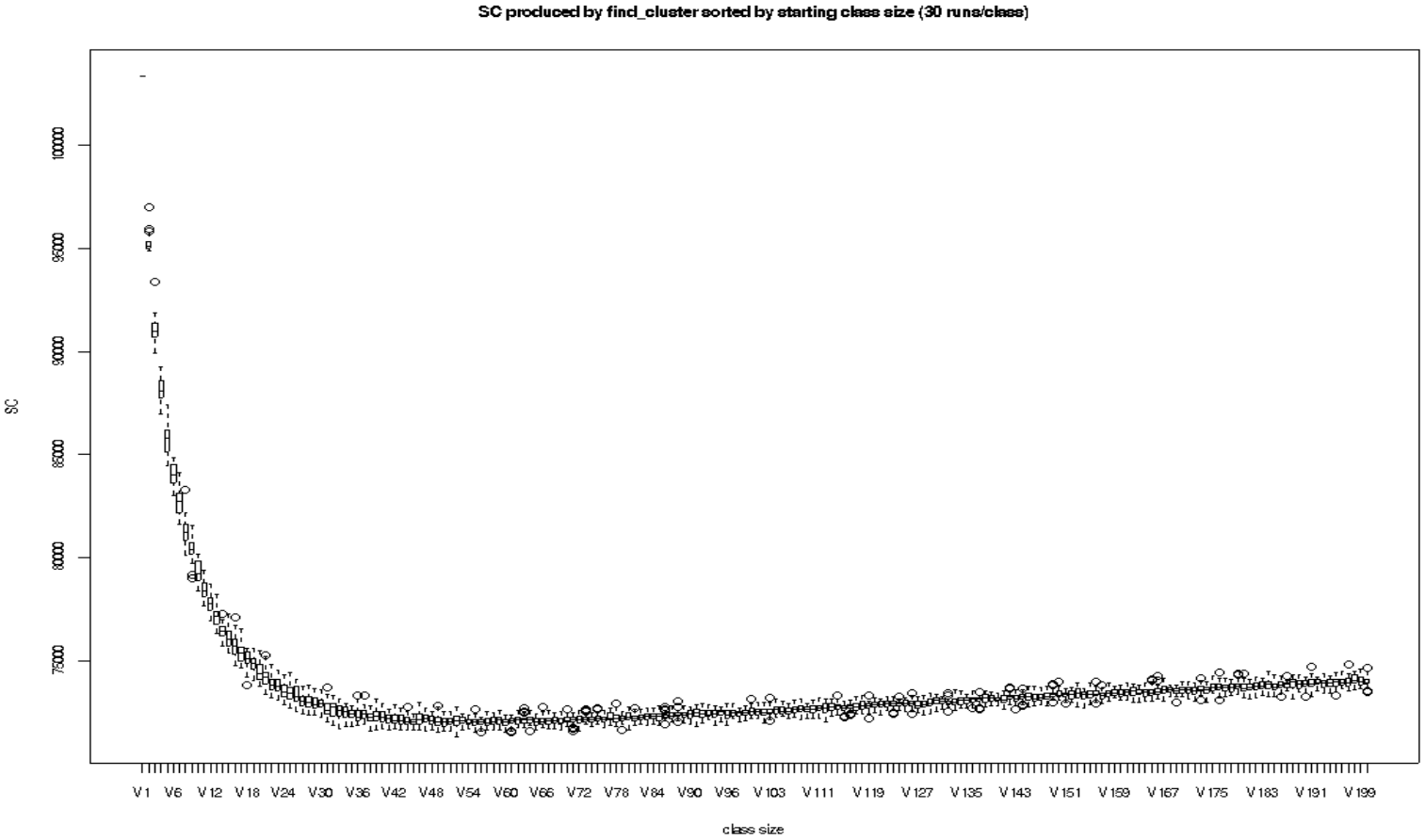
In this application it seems that there is no appreciable gain in using the clustering algorithm with tree dependency.

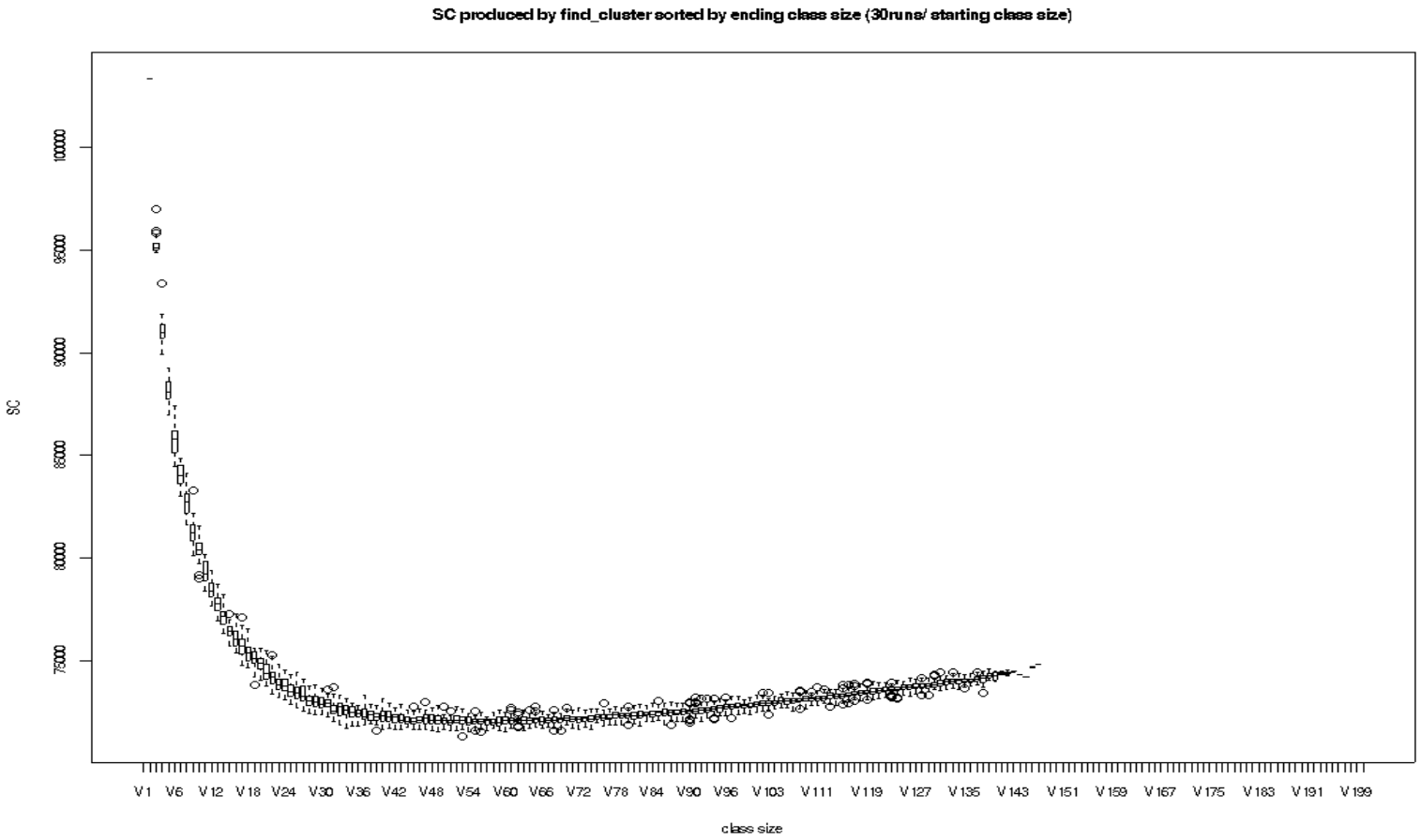
Application (1): Rand values

	1	2	3	4	5
1	1.0000	.89377	.90229	.89671	.89687
2	.89377	1.0000	.96504	.98546	.98529
3	.90229	.96504	1.0000	.96639	.96665
4	.89671	.98546	.96639	1.0000	.99674
5	.89687	.98529	.96665	.99674	1.0000

KTH Matematik

SC

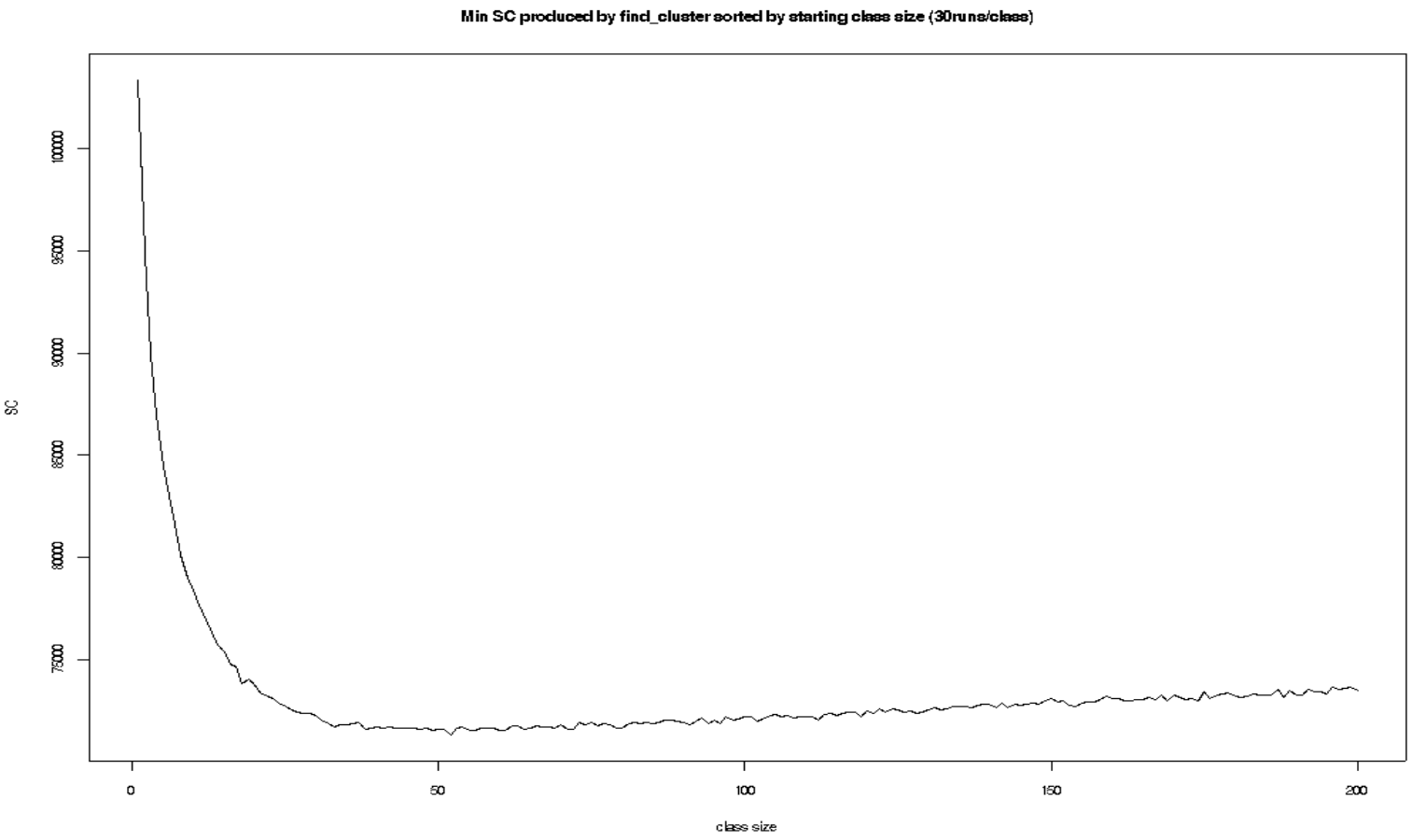

SC produced by find_cluster sorted by starting class size (30 runs/class)
SC
100000
95000
90000
85000
80000
75000
V1 V6 V12 V18 V24 V30 V36 V42 V48 V54 V60 V66 V72 V78 V84 V90 V96 V103 V111 V119 V127 V135 V143 V151 V159 V167 V175 V183 V191 V199
class size



SC

All the ending SC values boxplotted.

SC


Min SC produced by find_cluster sorted by starting class size (30runs/class)
SC
100000
95000
90000
85000
80000
75000
0
50
100
150
200
class size

Application (2) to microbial classification

Source of data:
F.L. Thompson, B. Hoste, K. Vandenmeulebroecke and J. Swings (2001): Genomic Diversity Amongst *Vibrio* Isolates from Different Sources Determined by Fluorescent Amplified Fragment Length Polymorphism. *Systematic and Applied Microbiology*, 24, pp. 520−538.

Application (2) to microbial classification

A restriction endonuclease recognizes a specific sequence of nucleotide pairs and cleaves it. The number and locations of restriction sites vary with nucleotide sequence. The higher the similarity of the two DNA sequences compared, the closer the cleavage pattern.

KTH Matematik

AFLP (=Amplified Fragment Length Polymorphism)

The DNA fingerprinting technique known as AFLP (=Amplified Fragment Length Polymorphism) is based on the selective amplification of genomic restriction fragments by PCR (=Polymerase Chain Reaction) to differentiate bacterial strains at the subgeneric level and consists of three steps:

- (a) Digestion of total cellular DNA with two restriction enzymes and ligation of restriction halfsite-specific adaptors to all restriction fragments.
- (b) Selective amplification of these fragments with two PCR primers that have corresponding adaptor- and restriction-site-sequences as their target sites.
- (c) Electrophoretic separation of the PCR products on a gel.

KTH Matematik

AFLP (=Amplified Fragment Length Polymorphism)

In electrophoresis, each organism is characterized by a banding pattern, which are thus a direct reflection of the genetic relationship between the bacterial strains examined and therefore these banding patterns can be considered as genomic fingerprints allowing computational algorithms for characterization (typing) and identification purposes. A collection of fingerprints can be translated into a matrix of binary variables, band present (1) or band absent (0)

KTH Matematik

Application (2) to microbial classification

- The 505 FAFLP (=Fluorescent Amplified Fragment Length Polymorphism, AFLP without radioactive materials) fingerprints of genomes of *Vibrio* strains listed in were binarized using standard binarizing software. The preceding algorithm was then applied to this set of data with several randomized initial $U_{(w)}$.

KTH Matematik

Application (2) to microbial classification

- 994 bit *Vibrionaceae* data
-

		SC
1	Binclass (50 trials/cluster size)	156029.7
2	Thompson et.al.	147960.1
3	Chow-Liu tree (5trials/cluster size)	121021.2

KTH Matematik

Application (2): comparison of SC for various classifications

The SC-values were computed using

$$SC_k\left(D, U^N|\mathcal{T}\right) = -\log P(U^N) - \log P\left(D|\mathcal{T}, U^N\right)$$

and by estimating a Chow-Liu tree for all classes in all classifications.

KTH Matematik

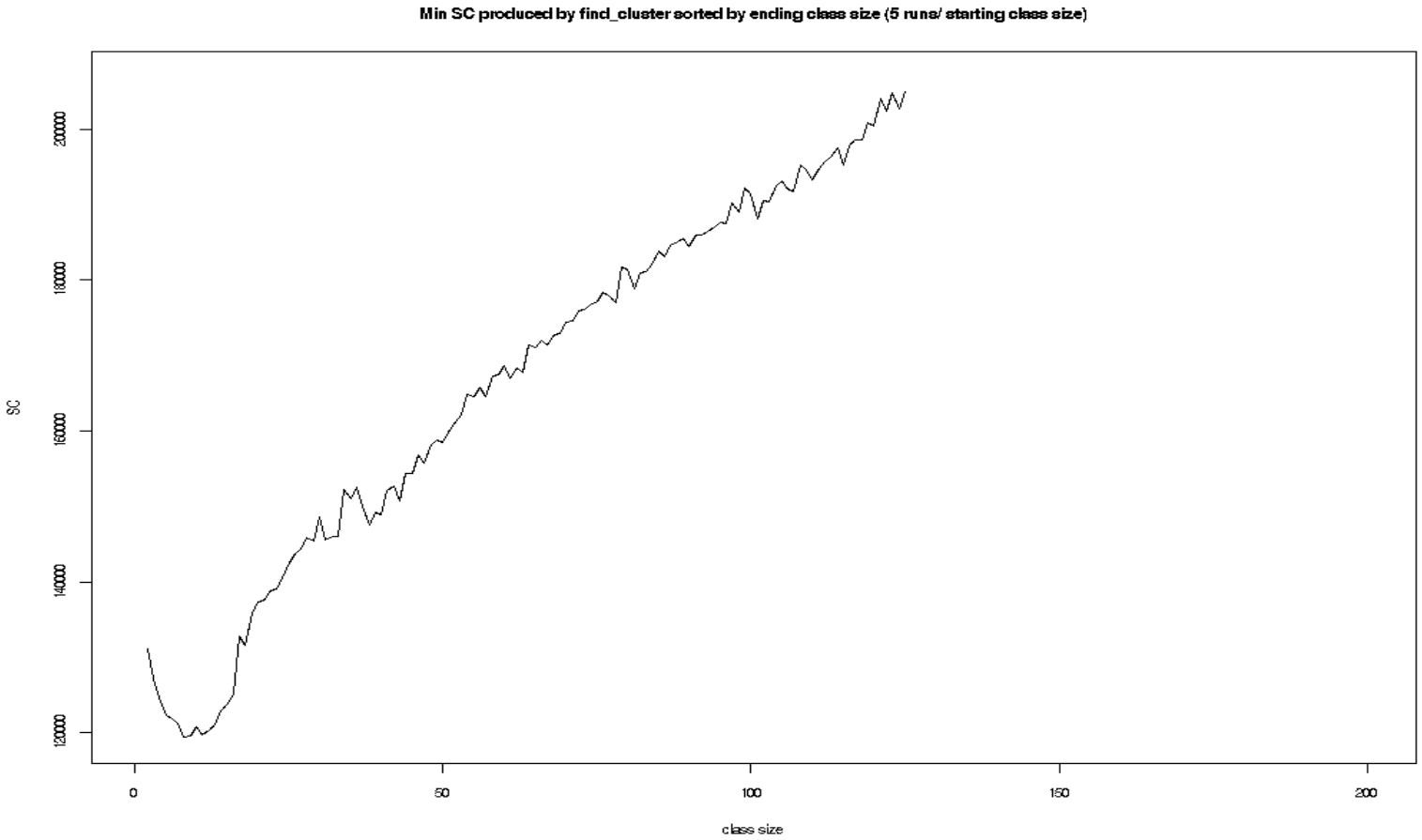
Application (2) : Rand similarities

	1	2	3
1	1.0000	.98603	.77212
2	.98603	1.0000	.78031
3	.77212	.78031	1.0000

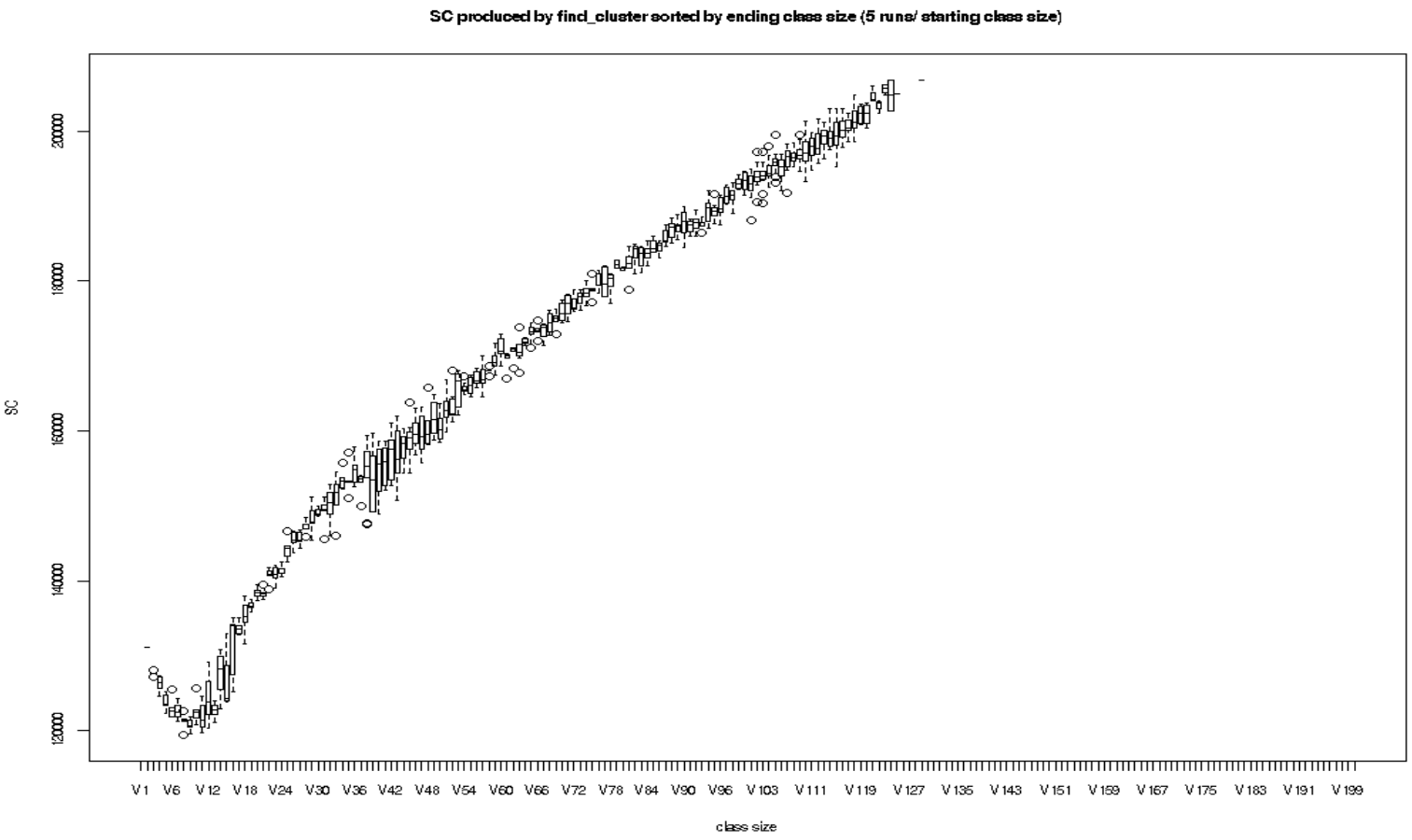
KTH Matematik

SC

Here is a graphical description of the SC found when scanning 1-200 classes sorted by ending class size.



SC



KTH Matematik

SC & Genomic Diversity

Because classification by minimization of SC compresses data, it comes as no surprise that the optimum number of clusters (classes) found by the algorithm turns out to be smaller than the number of clusters (=69) established in loc.cit.. The last mentioned clustering is obtained from a dendrogram, based on Dice's coefficient of similarity and Ward's method of agglomeration cut at a subjectively chosen threshold value.

KTH Matematik

SC & Genomic Diversity

The classification minimizing SC should in this context be evaluated with respect to its biological relevance. By inspection of the results one finds that 49 of the 69 clusters in Thompson have remained intact or have all but one or two of their strains in the same larger class in the classification minimizing SC. However, the two clusters with the type strains *Vibrio penaeicida* and *Vibrio tubiashi*, respectively, have been completely disintegrated in the classification minimizing SC.

KTH Matematik

SC & Genomic Diversity

Another way to look at the biological meaning of the result is by means of the class memberships of the *type strains*. The clusters in Thompson et.al. can in broad terms be described as falling in three categories: (a) isolates with genomes related to only one known type strain, (b) isolates clustering to one or more type strains, and (c) isolates with genome unrelated to any type strain (less than 45 % similarity).

KTH Matematik

SC & Genomic Diversity

It is clear that Bayes rule does not measure similarity in the same explicit biological sense as Dice's coefficient. Hence it is interesting to note that in the classification minimizing SC those 49 classes that have remained intact have the corresponding type strains in the same cluster in the classification minimizing SC or are clusters that do not have any type strain either in Thompson et.al. or in the classification minimizing SC.

KTH Matematik

Thank you !

The flower of monkey−bread tree

KTH Matematik