Scoring functions for learning Bayesian networks

Alexandra M. Carvalho

Plan

- Learning Bayesian networks
- Scoring functions for learning Bayesian networks:
  - Bayesian scoring functions:
    - BD (Bayesian Dirichlet) (1995)
    - BDe ("'e"' for likelihood-equivalence) (1995)
    - BDeu ("'u"' for uniform joint distribution) (1991)
    - K2 (1992)
  - Information-theoretic scoring functions:
    - LL (Log-likelihood) (1912-22)
    - MDL/BIC (Minimum description length/Bayesian Information Criterion) (1978)
    - AIC (Akaike Information Criterion) (1974)
    - NML (Normalized Minimum Likelihood) (2008)
    - MIT (Mutual Information Tests) (2006)
  - Decomposability and score equivalence
- Experiments
- Conclusion

Bayesian networks

Definition. ***Bayesian network***

A n-*dimensional Bayesian network* (BN) is a triple $B = (\mathbf{X}, G, \Theta)$ where:

- $\mathbf{X}$ is a n-dimensional finite random vector where each random variable X_i ranged over by a finite domain D_i. Henceforward, we denote the joint domain by $\mathbf{D} = \prod_{i=1}^{n} D_i$.
- $G = (N, E)$ is a directed acyclic graph (DAG) with nodes $N = \{X_1, \ldots, X_n\}$ and edges E representing direct dependencies between the variables.
- Θ encodes the parameters $\{\theta_{ijk}\}_{i \in 1 \ldots n,\, j \in D_{\Pi_{X_i}},\, k \in D_i}$ of the network, where

$$\theta_{ijk} = P_B(X_i = x_{ik} | \Pi_{X_i} = w_{ij}),$$

Π_{X_i} denotes the set of parents of X_i in G, $D_{\Pi_{X_i}}$ denotes the joint domain of the variables in Π_{X_i}, x_{ik} is the k-th value of X_i and w_{ij} is the j-th configuration of Π_{X_i}.

Bayesian networks

A BN defines a unique joint probability distribution over $\mathbf{X}$ given by

$$P_B(X_1, \ldots, X_n) = \prod_{i=1}^{n} P_B(X_i | \Pi_{X_i}).$$

- A BN encodes the independence assumptions over the component random variables of $\mathbf{X}$.
- An edge (j, i) in E represents a direct dependency of X_i from X_j.
- The set of all Bayesian networks with n variables is denoted by $\mathcal{B}_n$.

Learning Bayesian networks

Learning a BN:

- The problem of learning a BN given data T consists on finding the BN that best fits the data T.
- In order to quantify the fitting of a BN a scoring function ϕ is considered.

Definition. ***Learning a Bayesian network***
Given a data $T = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$ and a scoring function ϕ, the *problem of learning a Bayesian network* is to find a Bayesian network $B \in \mathcal{B}_n$ that maximizes the value $\phi(B, T)$.

Hardness results

- Cooper (1990) showed that the inference of a general BN is a NP-hard problem.
  $\Longrightarrow$ **APPROXIMATE SOLUTIONS**
- Dagum and Luby (1993) showed that even finding an approximate solution is NP-hard.
  $\Longrightarrow$ **RESTRICT SEARCH SPACE**
- First attempts confined the network to **tree structures** and used Edmonds (1967) and Chow-Liu (1968) optimal branching algorithms to learn the network.
- More general classes of BNs have eluded efforts to develop efficient learning algorithms.
- Chickering (1996) showed that learning the structure of a BN is NP-hard even for networks constrained to have in-degree at most 2.
- Dasgupta (1999) showed that even learning 2-polytrees is NP-hard.
- Due to these hardness results exact polynomial-time bounded approaches for learning BNs have been restricted to tree structures.

Standard methodology

- The standard methodology for addressing the problem of learning BNs became **heuristic search, based on scoring metrics optimization, conducted over some search space**.
- Search space:
  - Network structures
  - Equivalence classes of network structures
  - Orderings over the network variables
- Algorithm to search the space:
  - Greedy hill-climbing
  - Simulated annealing
  - Genetic algorithms
  - Tabu search
- Scoring functions are commonly classified into two main categories:
  - **Bayesian scoring functions**
  - **Information-theoretic scoring functions**

Notation

r_i	number of states of the finite random variable X_i
x_{ik}	k-th value of X_i
$q_i = \prod_{X_j \in \Pi_{X_i}} r_j$	number of possible configurations of the parent set Π_{X_i} of X_i
w_{ij}	j-thconfiguration of Π_{X_i} $(1 \leq j \leq q_i)$
N_{ijk}	number of instances in the data T where the variable X_i takes its k-th value x_{ik} and the variables in Π_{X_i} take their j-th configuration w_{ij}
$N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$	number of instances in the data T where the variables in Π_{X_i} take their j-th configuration w_{ij}
$N_{ik} = \sum_{j=1}^{q_i} N_{ijk}$	number of instances in the data T where the variable X_i takes its k-th value x_{ik}
N	total number of instances in the data T

Bayesian scoring functions

- Compute the posterior probability distribution, starting from a prior probability distribution on the possible networks, conditioned to data T, that is, $P(B|T)$.
- The best network is the one that maximizes the posterior probability.
- Since the term $P(T)$ is the same for all possible networks, in practice, for comparative purposes, computing $P(B,T)$ is sufficient.
- As it is easier to work in the logarithmic space, the scoring functions use the value $\log(P(B,T))$ instead of $P(B,T)$.

BD scoring function

Heckerman, Geiger and Chickering (1995) proposed the ***Bayesian Dirichlet* (BD) score** by making **four assumptions on** $P(B,T)$.

Notation.

$\Theta_G = \{\Theta_i\}_{i=1,\ldots,n}$	Encodes parameters of a BN B with underlying DAG G
$\Theta_i = \{\Theta_{ij}\}_{j=1,\ldots,q_i}$	Encodes parameters concerning only the variable X_i of $\mathbf{X}$ in B
$\Theta_{ij} = \{\theta_{ijk}\}_{k=1,\ldots,r_i}$	Encodes parameters for variable X_i of $\mathbf{X}$ in B given that its parents take their j-th configuration

Assumption 1. ***Multinomial sample***

For any data $T = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$, Bayesian network B, variable X_i of $\mathbf{X}$ in B and instance $\mathbf{y}_t \in T$,

$$P_B(\mathbf{y}_{ti} = x_{ik} | \mathbf{y}_{t\Pi_{X_i}} = w_{ij}, T_t) = P_B(X_i = x_{ik} | \Pi_{X_i} = w_{ij}) = \theta_{ijk}$$

for $k = 1, \ldots, r_i$ and $j = 1, \ldots, q_i$, where $T_t = \{\mathbf{y}_1, \ldots, \mathbf{y}_{t-1}\}$.

BD scoring function

Assumption 2. ***Dirichlet***

Given a directed acyclic graph G such that $P(G) > 0$ then Θ_{ij} is Dirichlet for all Θ_{ij} in Θ_G.

Assumption 3. ***Parameter independence***

Given a directed acyclic graph G such that $P(G) > 0$ then

1. $\rho(\Theta_G|G) = \prod_{i=1}^{n} \rho(\Theta_i|G)$ (**global parameter independence**), and
2. $\rho(\Theta_i|G) = \prod_{j=1}^{q_i} \rho(\Theta_{ij}|G)$ for all $i = 1, \ldots, n$ (**local parameter independence**).

Assumption 4. ***Parameter modularity***

Given two directed acyclic graphs, G and G', such that $P(G) > 0$ and $P(G') > 0$, if X_i has the same parents in G and G', then

$$\rho(\Theta_{ij}|G) = \rho(\Theta_{ij}|G')$$

for all $j = 1, \ldots, q_i$.

BD scoring function

Theorem. ***Heckerman, Geiger and Chickering (HGC95)***
Under assumptions 1 through 4 we have that

$$P(B,T) = P(B) \times \prod_{i=1}^{n} \prod_{j=1}^{q_i} \left(\frac{\Gamma(N'_{ij})}{\Gamma(N_{ij} + N'_{ij})} \times \prod_{k=1}^{r_i} \frac{\Gamma(N_{ijk} + N'_{ijk})}{\Gamma(N'_{ijk})} \right)$$

where Γ is the Gamma function and P(B) represents the prior probability of the network B.

BD scoring function

The HGC95 theorem induces the ***Bayesian Dirichlet* (BD) score**:

$$\mathrm{BD}(B,T) = \log(P(B)) + \sum_{i=1}^{n}\sum_{j=1}^{q_i}\left(\log\left(\frac{\Gamma(N'_{ij})}{\Gamma(N_{ij}+N'_{ij})}\right) + \sum_{k=1}^{r_i}\log\left(\frac{\Gamma(N_{ijk}+N'_{ijk})}{\Gamma(N'_{ijk})}\right)\right).$$

The BD score is unusable in practice:

- Specifying all hyperparameters N'_{ijk} for all i, j and k is formidable, to say the least.
- There are some particular cases of the BD score that are useful...

K2 scoring function

Cooper and Herskovits (1992) proposed a particular case of the BD score, called the ***K2* score**,

$$\text{K2}(B,T) = \log(P(B)) + \sum_{i=1}^{n}\sum_{j=1}^{q_i}\left(\log\left(\frac{(r_i-1)!}{(N_{ij}+r_i-1)!}\right) + \sum_{k=1}^{r_i}\log(N_{ijk}!)\right),$$

with the uninformative assignment $N'_{ijk} = 1$ (corresponding to zero pseudo-counts).

BDe scoring function

Heckerman, Geiger and Chickering (1995) turn around the problem of hyperparameter specification by considering two additional assumptions: **likelihood equivalence** and **structure possibility**.

Definition. ***Equivalent directed acyclic graphs***
Two directed acyclic graphs are *equivalent* if they can encode the same joint probability distributions.

Given a Bayesian network B, data T can be seen as a multinomial sample of the joint space $\mathbf{D}$ with parameters

$$\Theta_{\mathbf{D}} = \{\theta_{x_1 \ldots x_n}\}_{x_i=1,\ldots,r_i,\, i \in 1 \ldots n}$$

where $\theta_{x_1 \ldots x_n} = \prod_{i=1}^{n} \theta_{x_i|\Pi_{x_i}}$.

Assumption 5. ***Likelihood equivalence***
Given two directed acyclic graphs, G and G', such that $P(G) > 0$ and $P(G') > 0$, if G and G are equivalent then $\rho(\Theta_{\mathbf{D}}|G) = \rho(\Theta_{\mathbf{D}}|G')$.

BDe scoring function

The ***skeleton*** of any DAG is the undirected graph resulting from ignoring the directionality of every edge.

Definition. ***Complete directed acyclic graph***
A directed acyclic graph is said to be *complete* if its skeleton is complete.

Assumption 6. ***Structure possibility***
For any complete directed acyclic graph G, we have that $P(G) > 0$.

BDe scoring function

Theorem. ***Heckerman, Geiger, Chickering (HGC95)***
Suppose that $\rho(\Theta_{\mathbf{D}}|G)$ is Dirichlet with equivalent sample size N' for some complete directed acyclic graph G in $\mathbf{D}$. Then, for any Bayesian network B in $\mathbf{D}$, Assumptions 1 through 6 imply

$$P(B,T) = P(B) \times \prod_{i=1}^{n} \prod_{j=1}^{q_i} \left(\frac{\Gamma(N'_{ij})}{\Gamma(N_{ij}+N'_{ij})} \times \prod_{k=1}^{r_i} \frac{\Gamma(N_{ijk}+N'_{ijk})}{\Gamma(N'_{ijk})} \right)$$

where $N'_{ijk} = N' \times P(X_i = x_{ik}, \Pi_{X_i} = w_{ij}|G)$.

The ***equivalent sample size*** N' expresses the strength of our belief in the prior distribution.

BDe scoring function

The HGC95 theorem induces the ***likelihood-equivalence Bayesian Dirichlet*** **(BDe) score** and its expression is identical to the BD expression.

The BDe score is of little practical interest:

- It requires knowing $P(X_i = x_{ik}, \Pi_{X_i} = w_{ij}|G)$ for all i, j and k, which might not be elementary to find.

BDeu scoring function

Buntine (1991) proposed a particular case of BDe score, called the ***BDeu* score**:

$$\text{BDeu}(B,T) = \log(P(B)) + \sum_{i=1}^{n}\sum_{j=1}^{q_i}\left(\log\left(\frac{\Gamma(\frac{N'}{q_i})}{\Gamma(N_{ij}+\frac{N'}{q_i})}\right) + \sum_{k=1}^{r_i}\log\left(\frac{\Gamma(N_{ijk}+\frac{N'}{r_i q_i})}{\Gamma(\frac{N'}{r_i q_i})}\right)\right),$$

which appears when

$$P(X_i = x_{ik}, \Pi_{X_i} = w_{ij}|G) = \frac{1}{r_i q_i}.$$

This score only depends on one parameter, the equivalent sample size N':

- Since there are no generally accepted rule to determine the hyperparameters $N'_{x_1 \ldots x_n}$, there is no particular good candidate for N'.
- In practice, the BDeu score is very sensitive with respect to the equivalent sample size N' and so, several values are attempted.

Information-theoretic scoring functions

Information-theoretic scoring functions are based on compression:

- The score of a Bayesian network B is related to the compression that can be achieved over the data T with an optimal code induced by B.
- Shannon's source coding theorem (or noiseless coding theorem) establishes the **limits to possible data compression**.

Theorem. ***Shannon source coding theorem***
As the number of instances of an i.i.d. data tends to infinity, no compression of the data is possible into a shorter message length than the total Shannon entropy, without losing information.

Several optimal codes asymptotically achieve Shannon's limit:

- **Fano-Shannon** code and **Huffman code**, for instance.
- Building such codes requires a probability distribution over data T.

Information-theoretic scoring functions

Information content of T by B:

- The size of an optimal code, induced by the distribution B, when encoding T.
- This value can be used to score the BN B.

$$
\begin{aligned}
L(T|B) &= -\log(P_B(T)) \\
&= -\sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i} N_{ijk}\log(\theta_{ijk}) \\
&= -\sum_{i=1}^{n}\sum_{j=1}^{q_i} N_{ij}\sum_{k=1}^{r_i}\frac{N_{ijk}}{N_{ij}}\log(\theta_{ijk}).
\end{aligned}
$$

Information-theoretic scoring functions

Lemma. ***Gibb's inequality***

Let $P(x)$ and $Q(x)$ be two probability distributions over the same domain, then

$$\sum_x P(x)\log(Q(x)) \leq \sum_x P(x)\log(P(x)).$$

Some observations from Gibb's inequality:

- When fixing the DAG structure of a BN B, $L(T|B)$ is minimized when

$$\theta_{ijk} = \frac{N_{ijk}}{N_{ij}}.$$

- $L(T|B)$ is minimal when the likelihood $P_B(T)$ of T given B is maximal.
- The parameters of B that induces a code that compresses T the most is precisely the parameters that maximizes the probability of observing T.

LL scoring function

The ***log-likelihood (LL)* score** is defined in the following way:

$$\mathrm{LL}(B|T) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} \log \left(\frac{N_{ijk}}{N_{ij}} \right) .$$

- The LL score tends to favor complete network structures and it does not provide an useful representation of the independence assumptions of the learned network.
- This phenomenon of ***overfitting*** is usually avoided in two different ways:
  - By limiting the number of parents per network variable.
  - By using some ***penalization factor*** over the LL score:
    - MDL/BIC (*Occam's razor* approach)
    - AIC
    - NML (Stochastic complexity)

MDL scoring function

The ***minimum description length* (MDL) score** is an *Occam's razor* approach to fitting, preferring simple BNs over complex ones:

$$\text{MDL}(B|T) = \text{LL}(B|T) - \frac{1}{2}\log(N)|B|,$$

where

$$|B| = \sum_{i=1}^{n}(r_i - 1)q_i$$

denotes the **network complexity**, that is, the number of parameters in Θ for the network B.

- The first term of the MDL score measures how many bits are needed to describe data T based on the probability distribution P_B.
- The second term of the MDL score represents the length of describing the network B, that is, it counts the number of bits needed to encode B, where $\frac{1}{2}\log(N)$ bits are used for each parameter in Θ.

AIC/BIC scoring function

The measure of the quality of a BN can be computed in several different ways:

$$\phi(B|T) = \mathsf{LL}(B|T) - f(N)|B|,$$

where $f(N)$ is a non-negative penalization function.

- If $f(N) = 1$, we have the ***Akaike Information Criterion*** **(AIC) scoring function**:

$$\mathsf{AIC}(B|T) = \mathsf{LL}(B|T) - |B|.$$

- If $f(N) = \frac{1}{2}\log(N)$, we have the ***Bayesian Information Criterion*** **(BIC) score** based on Schwarz Information Criterion, which coincides with the MDL score.
- If $f(N) = 0$, we have the LL score.

NML scoring function

Recently, **Roos, Silander, Konthanen and Myllymäki (2008)**, proposed a new scoring function based on the MDL principle.

Insights about the MDL principle:

- To explain data T one should always choose the hypothesis with smallest description that generates T.
- What is a ***description*** and its ***length***?
- First candidate: ***Kolmogorov complexity*** of T, that is, the size of the smallest program that generates T written in a fixed universal programming language.
  - Kolmogorov complexity is undecidable.
  - The size of the description depends on the chosen programming language.

NML scoring function

Given

- data T, and
- a set of probability distributions $\mathcal{H}$ that may be used to describe T,

we take the ***length of describing T with H*** to be the sum $L(T|H) + L(H)$, where

- $L(T|H)$ is the length (in bits) of the description of T when encoded with H, and
- $L(H)$ is the length of the description of H.

Defining $L(H)$ has never been consensual:

- Both BIC/MDL and AIC scores agree in setting $L(T|H) = -\text{LL}(H|T)$.
- AIC sets $L(H) = |B|$.
- BIC/MDL sets $L(H) = \frac{1}{2}\log(N)|B|$.

NML scoring function

Using $|B|$ in the expression of the complexity of a BN is, in general, an error:

- The parameters of a BN are conditional distributions. Thus, if there are probabilities in Θ taking value 0, they do not need to appear in the description of Θ.
- The same distribution (or probability value) might occur several times in Θ leading to patterns that can be exploited to compress Θ significantly.

There have been attempts to correct $L(H)$:

- Most of the works are supported more on empirical evidence than on theoretical results.
- The main breakthrough in the community was to consider ***normalized minimum likelihood codes***.

NML scoring function

The idea behind normalized minimum likelihood codes is the same of ***universal coding***:

- Suppose an encoder is about to observe data T which he plans to compress as much as possible.
- The encoder has a set of candidate codes $\mathcal{H}$ and he believes one of these codes will allow to compress the incoming data significantly.
- However, he has to choose the code before observing the data.
- In general, there is no code which, no mater what incoming data T is, will always mimic the best code for T.
- So what is the best thing that the encoder can do?
- There are simple solutions to this problem when $\mathcal{H}$ is finite, however, this is not the case for BNs.

NML scoring function

Recasting the problem in a stochastic wording:

- Given a set of probability distributions $\mathcal{H}$ the encoder thinks that there is one distribution $H \in \mathcal{H}$ that will assign high likelihood (low code length) to the incoming data T of fixed size N.
- We woukd like to design a code that for all T will compress T as close as possible to the code associated to $H \in \mathcal{H}$ that maximizes the likelihood of T.
- We call to this $H \in \mathcal{H}$ the ***best-fitting hypothesis***.

We can compare the ***performance of a distribution H w.r.t. H' of modeling T of size N*** by computing

$$-\log(P(T|H)) + \log(P(T|H')).$$

NML scoring function

Given a set of probability distributions $\mathcal{H}$ and a distribution $\overline{H}$ not necessarily in $\mathcal{H}$, the ***regret of $\overline{H}$ relative to $\mathcal{H}$ for T of size N*** is

$$-\log(P(T|\overline{H})) - \min_{H \in \mathcal{H}}(-\log(P(T|H))).$$

In many practical cases, given a set of hypothesis $\mathcal{H}$ and data T, we are always able to find the $H_{\mathcal{H}}(T) \in \mathcal{H}$ that minimizes $-\log(P(T|H))$:

- The regret of $\overline{H}$ relative to $\mathcal{H}$ for T of size N can be rewritten as

$$-\log(P(T|\overline{H})) + \log(P(T|H_{\mathcal{H}}(T))).$$

NML scoring function

The ***worst-case regret of $\overline{H}$ relative to $\mathcal{H}$ for data of size N*** is given by

$$\max_{T:|T|=N}(-\log(P(T|\overline{H})) + \log(P(T|H_{\mathcal{H}}(T)))).$$

Definition. ***Universal distribution***

Let $\mathcal{H}$ be a set of probability distributions for which it is always possible to find the distribution $H_{\mathcal{H}}(T) \in \mathcal{H}$ that minimizes $-\log(P(T|H))$. The *universal distribution relative to $\mathcal{H}$ for data of size N* is the probability distribution $H_{\mathcal{H}}(N)$ such that

$$H_{\mathcal{H}}(N) = \min_{\overline{H}} \max_{T:|T|=N}(-\log(P(T|\overline{H})) + \log(P(T|H_{\mathcal{H}}(T)))),$$

where the minimum is taken over all distributions on the data space of size N.

NML scoring function

The ***parametric complexity of $\mathcal{H}$ for data of size N*** is

$$\mathbf{C}_N(\mathcal{H}) = \log \left(\sum_{T:|T|=N} P(T|H_{\mathcal{H}}(T)) \right) .$$

Theorem. ***Shtakov (1987)***
Let $\mathcal{H}$ be a set of probability distributions such that $\mathbf{C}_N(\mathcal{H})$ is finite. Then, the universal distribution relative to $\mathcal{H}$ for data of size N is given by

$$P_{\mathcal{H}}^{\mathsf{NML}}(T) = \frac{P(T|H_{\mathcal{H}}(T))}{\sum_{T':|T'|=N} P(T'|H_{\mathcal{H}}(T'))}.$$

The distribution $P_{\mathcal{H}}^{\mathsf{NML}}(T)$ is called the ***normalized maximum likelihood*** **(NML)** distribution.

NML scoring function

Given data T of size N and two sets of probability distributions $\mathcal{H}_1$ and $\mathcal{H}_2$, the MDL principle states we should pick $\mathcal{H}_j$ that maximizes $P^{\mathsf{NML}}_{\mathcal{H}_j}(T)$, that is, we should pick $\mathcal{H}_j$ that maximizes

$$\begin{aligned} \log(P^{\mathsf{NML}}_{\mathcal{H}_j}(T)) &= \log(P(T|H_{\mathcal{H}_j(T)})) - \mathbf{C}_N(\mathcal{H}_j) \\ &= \mathsf{LL}(H_{\mathcal{H}_j(T)}|T) - \mathbf{C}_N(\mathcal{H}_j). \end{aligned}$$

The quantity $-\log(P^{\mathsf{NML}}_{\mathcal{H}_j}(T))$ is called the ***stochastic complexity of data T relative to $\mathcal{H}_j$***.

Let $\mathcal{B}_G$ denote the set of all BNs with network structure G. For a fixed a network structure G, the **NML score** is defined as

$$\mathsf{NML}(B|T) = \mathsf{LL}(B|T) - \mathbf{C}_N(\mathcal{B}_G).$$

NML scoring function

There is no hope for computing $\mathbf{C}_N(\mathcal{B}_G)$ efficiently:

- It involves an exponential sum over all possible data of size N.
- It is not decomposable over the network structure.

Roos, Silander, Konthanen and Myllymäki (2008), proposed to approximate $\mathbf{C}_N(\mathcal{B}_G)$ by considering only the contribution to the parametric complexity of the multinomial distributions associated to each variable given a parent configuration:

$$\mathbf{fC}_T(\mathcal{B}_G) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \mathbf{C}_{N_{ij}}(\mathcal{M}_{r_i}),$$

where $\mathcal{M}_{r_i}$ is the set of all multinomial distributions with r_i parameters.

NML scoring function

The ***factorized Normalized Maximum Likelihood* (fNML) score** is given by:

$$\text{fNML}(B|T) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \left(\sum_{k=1}^{r_i} N_{ijk} \log \left(\frac{N_{ijk}}{N_{ij}} \right) - \mathbf{C}_{N_{ij}}(\mathcal{M}_{r_i}) \right).$$

Computing $\mathbf{C}_{N_{ij}}(\mathcal{M}_{r_i})$:

- It seems exponential in N_{ij}, since it involves an exponential sum over all possible data of size N_{ij}.
- However, it was recently proposed by **Konthanen and Myllymäki (2007)** a **linear-time algorithm for computing the stochastic complexity in the case of N_{ij} observations of a single multinomial random variable**.
- For that purpose an elegant recursion formula was proposed based on the mathematical technique of **generating functions**.

MIT scoring function

A scoring function based on mutual information, called ***mutual information tests* (MIT) score**, was proposed by **de Campos (2006)** and its expression is given by

$$\text{MIT}(B|T) = \sum_{\substack{i=1 \\ \Pi_{X_i} \neq \emptyset}}^{n} \left(2NI(X_i; \Pi_{X_i}) - \sum_{j=1}^{s_i} \chi_{\alpha, l_{i\sigma_i^*(j)}} \right),$$

where $I(X_i; \Pi_{X_i})$ is the mutual information between X_i and Π_{X_i} in the network which measures the degree of interaction between each variable and its parents.

MIT scoring function

- The second term is a penalization related to the Pearson χ^2 test of independence:
  - α is a free parameter representing the confidence level associated with the statistical test.
  - $\sigma_i^* = (\sigma_i^*(1), \ldots, \sigma_i^*(s_i))$ denotes any permutation of the index set $(1, \ldots, s_i)$ of the variables in $\Pi_{X_i} = \{X_{i1}, \ldots, X_{is_i}\}$ satisfying

$$r_{i\sigma_i^*(1)} \geq r_{i\sigma_i^*(2)} \geq \cdots \geq r_{i\sigma_i^*(s_i)},$$

    where r_{ij} represents the number of possible configurations when the parent set of X_i is restricted only to X_j.
  - The number of degrees of freedom $l_{i\sigma_i^*(j)}$ is given by:

$$l_{i\sigma_i^*(j)} = \begin{cases} (r_i - 1)(r_{i\sigma_i^*(j)} - 1) \prod_{k=1}^{j-1} r_{i\sigma_i^*(k)} & j = 2, \ldots, s_i \\ (r_i - 1)(r_{i\sigma_i^*(j)} - 1) & j = 1. \end{cases}$$

Experiments

About the implementation:

- We implemented the **Chow-Liu tree learning algorithm and its extensions** in Mathematica 6.0, on top of the **Combinatorica package (Pemmaraju and Skiena, 2003)**.
- The package was extended with a **non-recursive, and efficient, version of Edmonds' algorithm** to build a maximal directed spanning tree of a strongly connected weighted directed graphs.
- A package to learn Bayesian network classifiers was implemented, and at the moment it allows to **learn an optimal TAN classifier for any score discussed in this work**.
- The package also contains the **entropy based discretization algorithm** by Fayyad and Irani (1993) to deal with continuous datasets.

Experiments

Scores used in the experiments:

- Information-theoretic scores: LL, BIC/MDL, NML and MIT with a 99% confidence level.
- Bayesian scores: K2 and BDeu with equivalent sample sizes 1, 4 and 16.

The **accuracy** of each classifier is based on the percentage of successful predictions on the test sets of each dataset:

- Accuracy was measured via the ***holdout method*** for larger training sets, and via ***5-fold cross-validation*** for smaller ones.
- Accuracy is annotated by a 95% confidence interval.

Experiments

Dataset	n	$\lvert D_C \rvert$	Train	Test
letter	16	26	15000	5000
satimage	36	6	4435	2000
chess	36	2	2130	1066
vehicle	18	4	846	CV-5
diabetes	8	2	768	CV-5
soybean-large	35	19	562	CV-5
vote	16	2	435	CV-5
heart	13	2	270	CV-5
glass	9	7	214	CV-5
iris	4	3	150	CV-5
lymphography	18	4	148	CV-5
hepatitis	19	2	80	CV-5

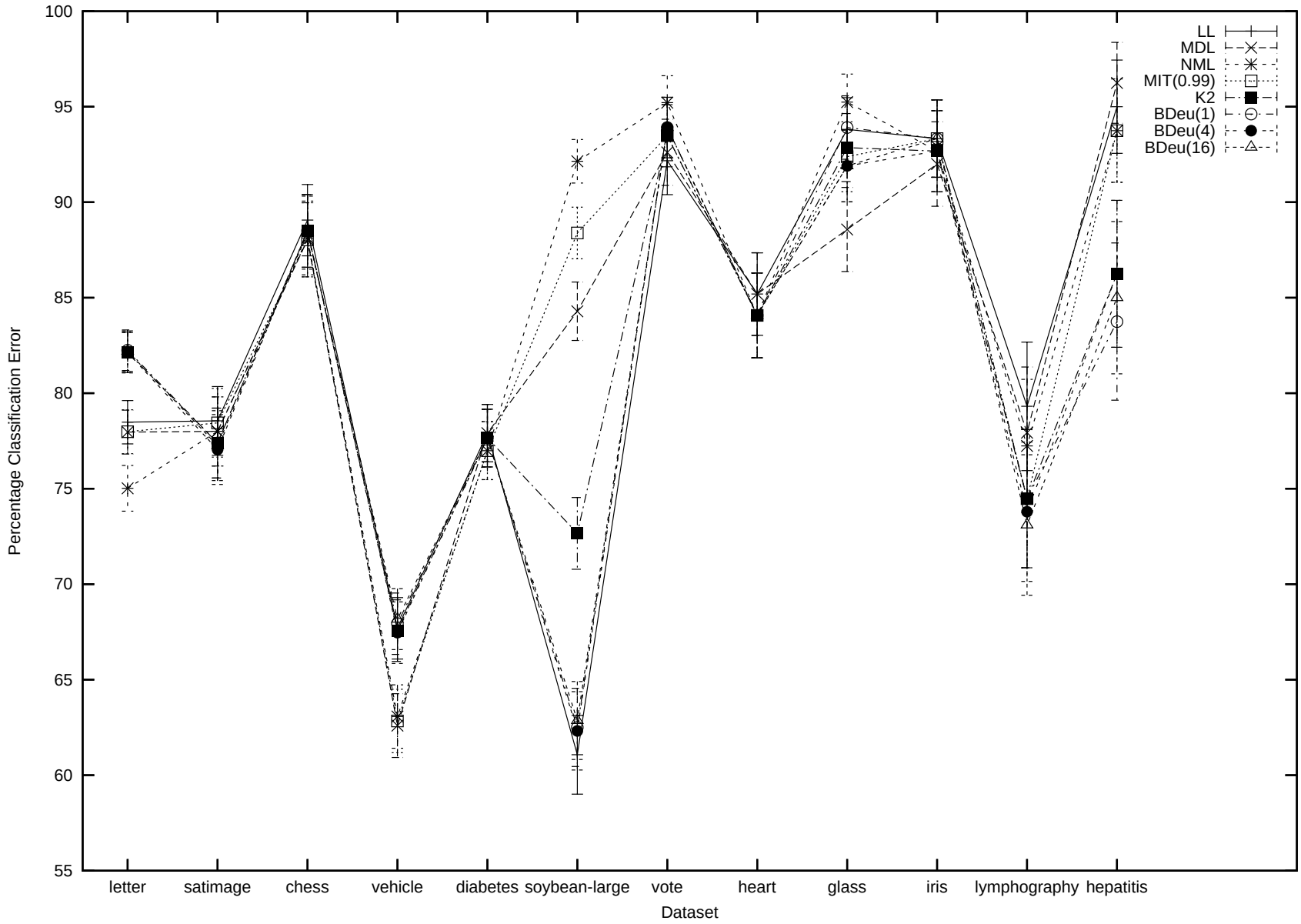
Experiments

Data set	LL	BIC/MDL	NML	MIT(0.99)
letter	78.48 ± 1.13	77.96 ± 1.15	75.02 ± 1.20	77.98 ± 1.15
satimage	78.55 ± 1.80	78.00 ± 1.81	78.00 ± 1.81	78.45 ± 1.80
chess	89.06 ± 1.87	88.03 ± 1.94	88.13 ± 1.93	88.03 ± 1.94
vehicle	$\mathbf{67.69 \pm 1.61}$	62.60 ± 1.67	63.07 ± 1.66	62.84 ± 1.66
diabetes	77.91 ± 1.50	77.91 ± 1.50	76.99 ± 1.52	76.99 ± 1.52
soybean-large	61.07 ± 2.06	84.29 ± 1.53	$\mathbf{92.14 \pm 1.14}$	88.39 ± 1.35
vote	92.17 ± 1.77	92.61 ± 1.73	95.21 ± 1.41	93.48 ± 1.63
heart	85.19 ± 2.16	85.19 ± 2.17	84.07 ± 2.22	84.07 ± 2.22
glass	93.81 ± 1.66	88.57 ± 2.20	95.24 ± 1.47	92.38 ± 1.83
iris	93.33 ± 2.03	92.00 ± 2.21	92.67 ± 2.12	93.33 ± 2.03
lymphography	79.31 ± 3.36	77.93 ± 3.44	77.24 ± 3.48	74.48 ± 3.62
hepatitis	$\mathbf{95.00 \pm 2.44}$	$\mathbf{96.25 \pm 2.12}$	$\mathbf{93.75 \pm 2.71}$	$\mathbf{93.75 \pm 2.71}$

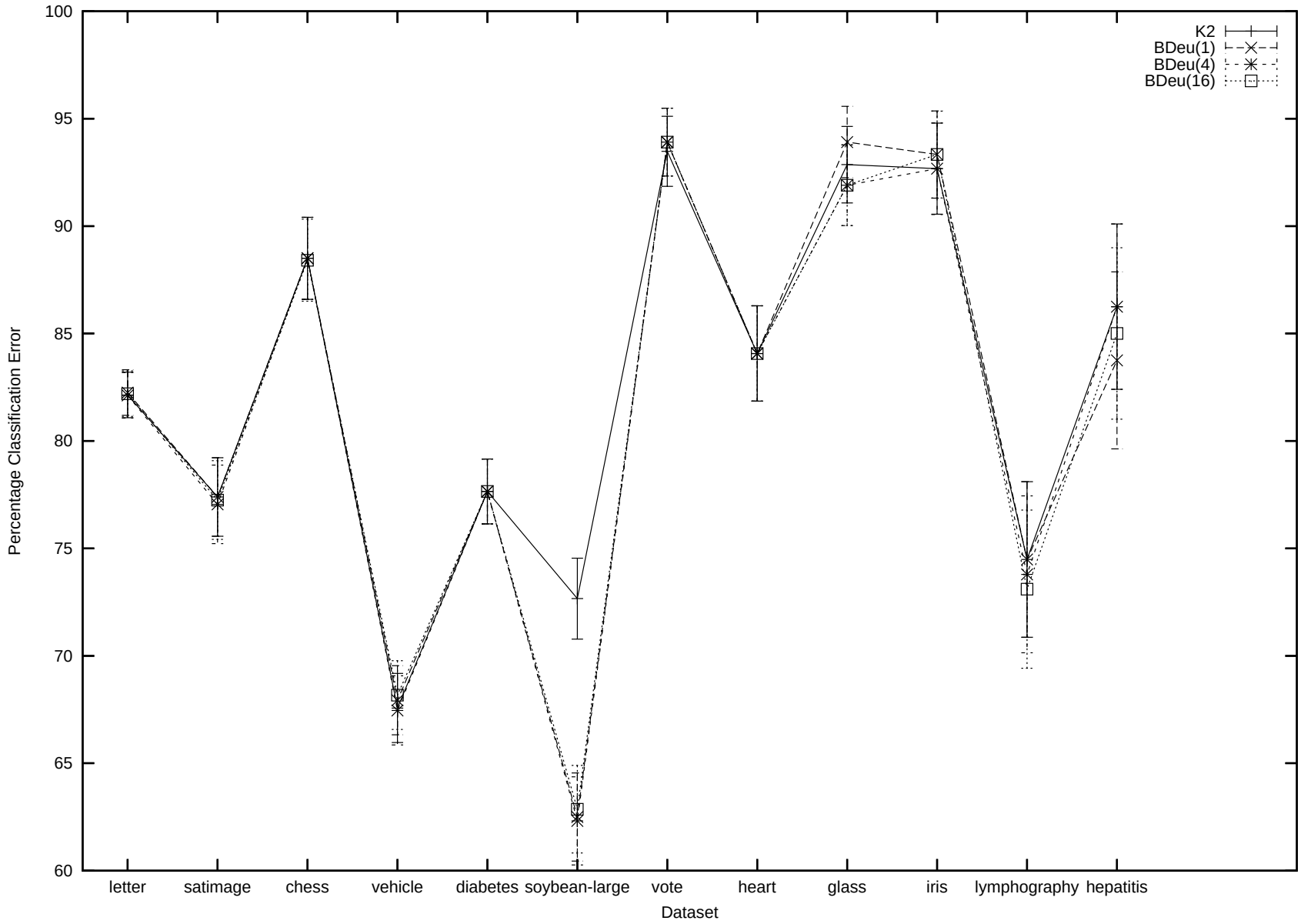
Experiments

Data set	K2	BDeu(1)	BDeu(4)	BDeu(16)
letter	$\mathbf{82.14 \pm 1.06}$	$\mathbf{82.25 \pm 1.06}$	$\mathbf{82.12 \pm 1.06}$	$\mathbf{82.20 \pm 1.06}$
satimage	77.39 ± 1.83	77.39 ± 1.83	77.05 ± 1.83	77.25 ± 1.83
chess	88.50 ± 1.91	88.50 ± 1.91	88.50 ± 1.91	88.41 ± 1.91
vehicle	$\mathbf{67.57 \pm 1.61}$	$\mathbf{67.93 \pm 1.61}$	$\mathbf{67.46 \pm 1.61}$	$\mathbf{68.17 \pm 1.60}$
diabetes	77.65 ± 1.51	77.65 ± 1.51	77.65 ± 1.51	77.65 ± 1.51
soybean-large	72.66 ± 1.88	62.50 ± 2.05	62.32 ± 2.05	62.86 ± 2.04
vote	93.48 ± 1.63	93.91 ± 1.58	93.91 ± 1.58	93.91 ± 1.58
heart	84.07 ± 2.22	84.07 ± 2.22	84.07 ± 2.22	84.07 ± 2.22
glass	92.86 ± 1.78	93.81 ± 1.66	91.90 ± 1.88	91.90 ± 1.88
iris	92.67 ± 2.12	93.33 ± 2.03	92.67 ± 2.13	93.33 ± 2.02
lymphography	74.48 ± 3.62	74.48 ± 3.62	73.79 ± 3.65	73.10 ± 3.68
hepatitis	86.25 ± 3.85	83.75 ± 4.12	86.25 ± 3.85	85.00 ± 3.99

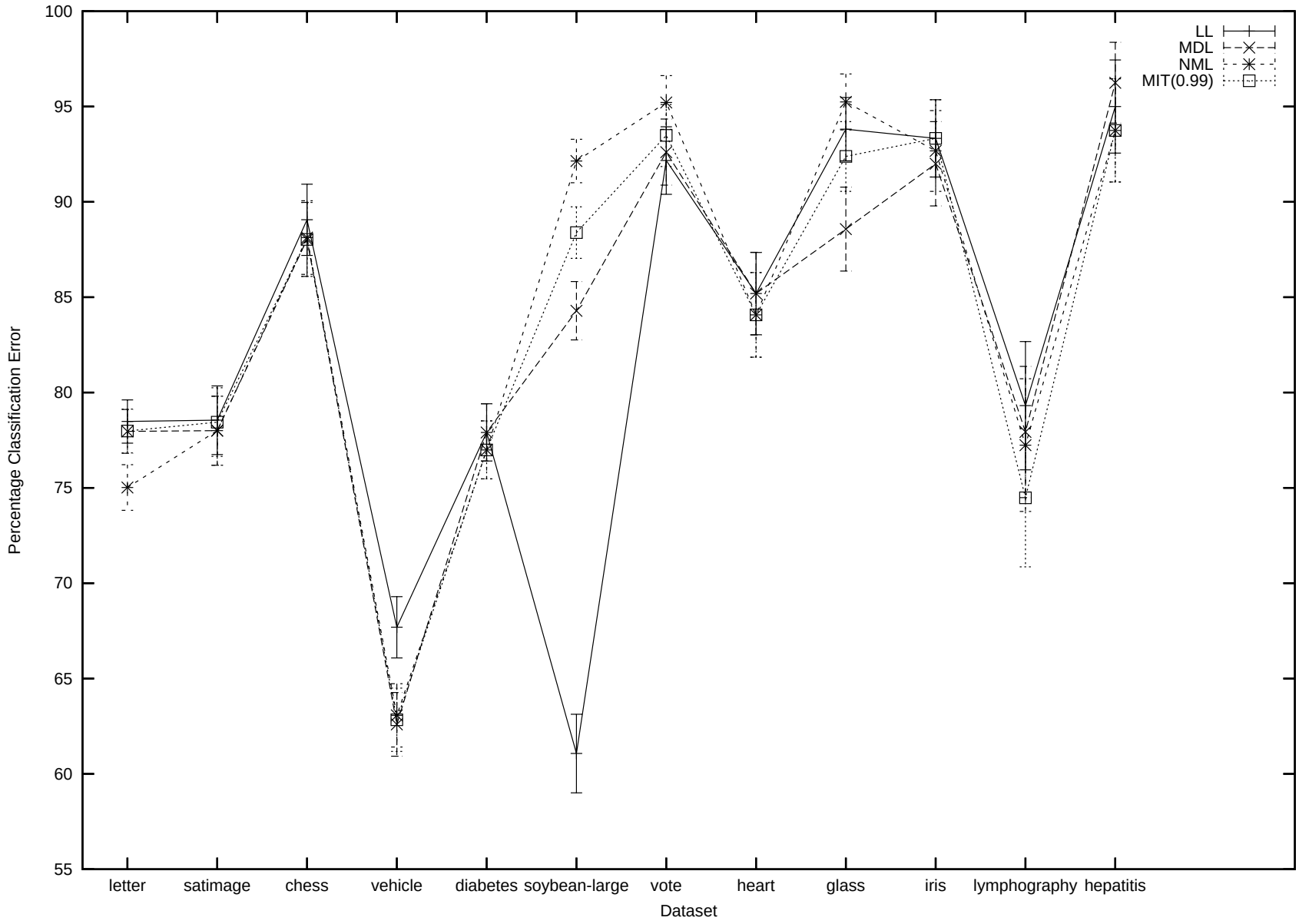
Experiments



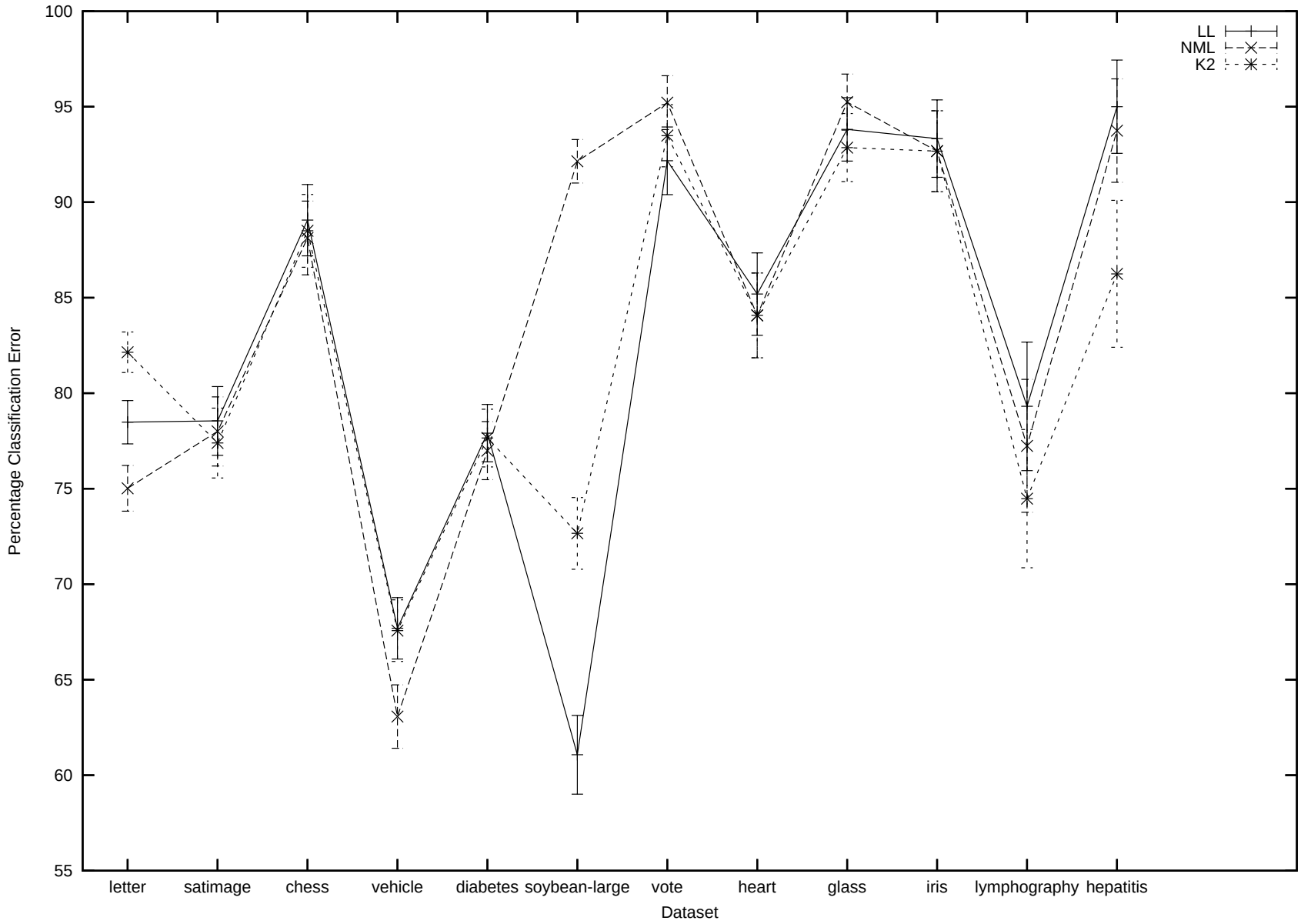
Experiments



Experiments



Experiments



Conclusions

- The results show that Bayesian scores are hard to distinguish, performing well for large datasets.
- The most impressive result was due to the NML score for the soybean-large dataset.
- It seems that a good choice is to consider K2 for large datasets and NML for small ones.