Latent Classification Models

*Classification in continuous domains*

Helge Langseth and Thomas D. Nielsen
Outline

➤ Probabilistic classifiers
➤ Naïve Bayes classifiers
  ● Relaxing the assumptions
➤ Latent classification models (LCMs)
  ● A linear generative model
  ● A non-linear generative model
➤ Learning LCMs
  ● Structural learning
  ● Parameter learning
➤ Experimental results
Classification in a probabilistic framework

A Bayes optimal classifier will classify an instance $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$ to the class:

$$y^* = \arg \min_{y \in \text{sp}(Y)} \sum_{y' \in \text{sp}(Y)} L(y, y') P(Y = y|\mathbf{X} = \mathbf{x}),$$

where $L(\cdot, \cdot)$ is the loss-function.

When learning a probabilistic classifier, the task is to learn the probability distribution $P(Y = y|\mathbf{X} = \mathbf{x})$ from a set of $N$ labeled training samples $\mathcal{D}_N = \{D_1, \ldots, D_N\}$. 
Learning a probabilistic classifier

An immediate approach:

➤ Learn a Bayesian network using standard score functions like e.g. MDL, BIC or BDe for Gaussian networks.

But such global score functions are not necessarily well-suited for learning a classifier!

Instead we could:

(i) Use predictive MDL as the score function.
(ii) Use the wrapper approach to score a classifier, i.e., apply cross-validation on the training data and use the estimated accuracy as the score.

Unfortunately, (i) does not decompose, and (ii) comes at the cost of high computational complexity!

The complexity problem can be relieved by focusing on a particular sub-class of BNs!
A Naïve Bayes classifier for the crabs domain:

Class: Blue male, Blue female
Orange male, Orange female

$X_1$: Width of frontal lib
$X_2$: Length along the midline
$X_3$: Racket width
$X_4$: Maximum width of the carapace
$X_5$: Body length

The two assumptions:

- The attributes are conditionally independent given the class.
- The continuous attributes are generated by a specific parametric family of distributions.
Conditional correlations: the crabs domain

A plot of Width of frontal lib vs. Rear Width for each of the four classes:

Note that there is a strong conditional correlation between the attributes.

➤ This is inconsistent with the independence assumptions of the Naïve Bayes classifier (in our experiments it achieves an accuracy of only 39.5% in this domain).
Handling conditional correlations

Methods for handling conditional dependencies can roughly be characterized as either:

➤ Allowing a more general correlation structure among the attributes.
  • **TAN** extended to the continuous domain (Friedman et al., 1998).
  • **Linear Bayes**; the continuous attributes are grouped together and associated with a multivariate Gaussian (Gama, 2000).

➤ Relying on a preprocessing of the data. For example:
  i) Transform the data into a vector of independent factors (e.g. *class-wise PCA* or *FA*).
  ii) Learn an **NB classifier** based on the expected values of the factors.
Transformation of the data: an example

A plot of the first two factors found by applying a PCA on the crabs database:

Note that the conditional correlation is reduced significantly.

➤ The Naïve Bayes classifier achieves an accuracy of 94.0% when working on the transformed data.
The Naïve Bayes usually assumes that the continuous attributes are generated by a specific parametric family of distributions (usually Gaussians) but this is not always appropriate:

The figure shows a histogram of the silicon contents in float processed glass taken from the glass2 domain.

To avoid having to make the Gaussian assumption, one could:

- Discretize the continuous attributes (Fayyad and Irani, 1993).
- Use kernel density estimation with Gaussian kernels (John and Langley, 1995).
- Use a dual representation (Friedman et al., 1998)
- Apply a finite mixture model (Monti and Cooper, 1999).
- ...
Consider a factor analysis for dimensionality reduction, i.e., the attributes $X$ is modeled with a $q$-dimensional vector of factor variables $Z$:

$$X = LZ + \epsilon,$$

where:
- $L$ is the regression matrix
- $Z \sim \mathcal{N}(0, I)$
- $\epsilon \sim \mathcal{N}(0, \Theta)$ is an $n$-dimensional random variable with diagonal covariance matrix $\Theta$.

In this model, the factor variables model the dependencies among the attributes, and $\epsilon$ can be interpreted as the sensor noise associated with the attributes.
Motivation: Consider the following simple idea for handling the conditional dependencies among the attributes.

(i) Learn a factor analyzer from the data.

(ii) For each data-point \( D_i = (x^i, y^i) \) let the expectation \( \mathbb{E}(Z|X = x^i, Y = y^i) \) represent observation \( x^i \) in the database.

(iii) Learn a Naïve Bayes classifier using the transformed dataset generated in step (ii).

Unfortunately:

- The conditional independencies are not consistent with the NB classifier.
- When we substitute the actual observations with the expected values of the factor variables in step (ii), we disregard information about the uncertainty in the generative model.
Latent Classification Models: The idea

Latent classification models can be seen as a combination of the generative model from an FA and a Naïve Bayes:

![Diagram of Latent Classification Models]

- Class
- $Z_1$
- $Z_2$
- $Z_3$
- $Z_4$

- $Z'_1$
- $Z'_2$
- $Z'_3$
- $Z'_4$

- $X_1$
- $X_2$
- $X_3$
- $X_4$
- $X_5$
Latent Classification Models: The linear case

For the quantitative part we have:

- **Conditionally on** $Y = j$ the latent variables, $Z$, follow a Gaussian distribution with 
  $\mathbb{E}[Z | Y = j] = \mu_j$ and $\text{Cov}(Z | Y = j) = \Gamma_j$.

- **Conditionally on** $Z = z$, the variables, $X$, follow a Gaussian distribution with 
  $\mathbb{E}(X|Z = z) = L z$ and $\text{Cov}(X|Z = z) = \Theta$.

Note that:

- Both $\Gamma_j$ and $\Theta$ must be diagonal.

- As opposed to the generative model underlying factor analysis, we do not assume that 
  $\mathbb{E}(Z|Y) = 0$ and $\text{Cov}(Z|Y) = I$.

The complexity is not that bad!!!
Linear Latent Classification Models: Expressive power

Proposition:
Assume that \( Y \) follows a multinomial distribution, and that \( X \mid \{ Y = j \} \sim \mathcal{N}(\alpha_j, \Sigma_j) \) for a given set of parameters \( \{\alpha_j, \Sigma_j\}_{j=1}^{\text{sp}(Y)} \). Assume that \( \text{rank}(\Sigma_j) = n \) for at least one \( j \). Then the joint distribution \((Y, X)\) can be represented by an LCM with \( q \leq n \cdot |\text{sp}(Y)| \).

Proof outline:
The proof is constructive. First we must show that for a given \( \{\alpha_j, \Sigma_j\}_{j=1}^{\text{sp}(Y)} \) we can define \( L, \Theta, \{\mu_j\}_{j=1}^{\text{sp}(Y)} \) and \( \{\Gamma_j\}_{j=1}^{\text{sp}(Y)} \) such that \( X \mid \{ Y = j \} \sim \mathcal{N}(\alpha_j, \Sigma_j) \).

- If \( L \) has rank \( n \), then \( L\mu = \alpha \) has a solution for any \( \alpha \in \mathbb{R}^n \).

Next we need to define \( L \) and \( \{\Gamma_j\}_{j=1}^{\text{sp}(Y)} \) (we assume that \( \Theta = 0 \)). Pick a \( \Sigma_j \) and note that since \( \Sigma_j \) is a positive semi-definite and symmetric matrix we can create a singular value decomposition of \( \Sigma_j \) as \( \Sigma_j = L_j \Lambda_j L_j^T \); \( L_j \) contains the eigenvectors of \( \Sigma_j \) and \( \Lambda_j \) is the diagonal matrix holding the corresponding eigenvalues.

- Define \( L \) by the \( L_j \) matrices, and let \( \Gamma_j \) be the block diagonal matrix constructed from the \( \Lambda_j \) and \( 0 \).
Non-Linear Latent Classification Models: Motivation

In the (linear) model, one of the main assumptions is that there exists a linear mapping, $L$, from the factor space to the attribute space:

- Conditionally on $Y = j$, the attributes are assumed to follow a multivariate Gaussian distribution with mean $L\mu_j$ and covariance matrix $L\Gamma_jL^T + \Theta$.

Unfortunately, this is not sufficiently accurate for some domains:

The figure shows the empirical probability distribution of the variables Silicon and Potassium for float processed glass (taken from the glass2 domain).

The proposed linear LCM achieved an accuracy of only 66.9%.
Non-Linear Latent Classification Models

In order to extend the expressive power of linear LCMs we propose a natural generalization, termed non-linear LCMs. Analogously to the linear LCM, the non-linear LCM can be seen as combining a NB classifier with a mixture of FA:

- a mixture variable $M$ governing the mixture distribution
- a (generalized) FA for each mixture component $M = m$:

$$X \mid \{M = m\} = L_m Z + \eta_m + \epsilon_m,$$

where $Z \sim \mathcal{N}(0, I)$ and $\epsilon \sim \mathcal{N}(0, \Theta_m)$. Note that $\eta_m$ models the data mean.
Latent Classification Models: the non-linear case

For the quantitative part everything is identical to the linear LCM except that:

➤ The mixture variable follows a multinomial distribution.

➤ Conditionally on \{\mathbf{Z} = z, M = m\}, the variables \mathbf{X} follow a Gaussian distribution with:
  • \( \mathbb{E}(\mathbf{X}|z, m) = L_m z + \eta_m \) and
  • \( \text{Cov}(\mathbf{X}|z, m) = \Theta_m \), where \( \Theta_m \) is assumed to be diagonal for all \( m \).
Proposition:
Assume that $Y$ follows a multinomial distribution, and that $X \mid \{Y = j\} \sim P(\Psi_j)$, where $P(\cdot)$ is a distribution function over $\mathbb{R}^n$ and $\{\Psi_j\}_{j=1}^{sp(Y)}$ are the parameters. Then the joint distribution for $(Y, X)$ can be approximated arbitrarily well by an LCM model.

Proof outline:
Set $q \geq 1$ and let $L_m = 0$, for all $m$. Thus:
- $X \mid \{Y = j, M = m\} \sim \mathcal{N}(\eta_m, \Theta_m)$, and
- $X \mid \{Y = j\}$ is therefore a mixture of (multivariate) Gaussians.

The result now follows trivially as any distribution over continuous variables can be approximated arbitrarily well by such a mixture of multivariate Gaussians.

Note that we can interpret the non-linear LCM as concurrently performing clustering and, within each cluster, local classification.
A few things to note:

➤ We score a model based on its accuracy (estimated using the wrapper approach).
➤ In order to traverse the search space we need:
  • A systematic approach for selecting values for \(q\) and \(|\text{sp}(M)|\).
  • Algorithms for learning the edge-set and the parameters in the model.
Learning Latent Classification Models

A few things to note:

➤ We **score** a model based on its accuracy (**estimated using the wrapper approach**).
➤ In order to **traverse the search space** we need:
  - A systematic approach for selecting values for \( q \) and \(|\text{sp}(M)|\).
  - Algorithms for learning the **edge-set** and the **parameters** in the model.

More formally, a general LCM learning algorithm can be formulated as:

1. **For** possible values of \( q \) and \(|\text{sp}(M)|\):
   (a) **Partition** the database into \( W \) wrapper folds \( \mathcal{W}_1 \ldots \mathcal{W}_W \).
   (b) **For** \( w = 1 \ldots W \):
      i. **Learn** a classifier from the dataset \( \mathcal{D}_N \setminus \mathcal{W}_w \) (both edge-set and parameters).
      ii. **Calculate** the accuracy on the remaining training-set \( \mathcal{W}_w \).
   (c) **Score** the parameter-pair \((q, |\text{sp}(M)|)\) by the average accuracy obtained over the wrapper folds.

2. **Select** the optimal values of \( q \) and \(|\text{sp}(M)|\).

3. **Return** classifier learned with these parameters.
Learning the edge set

First of all, note that:

➤ We will only consider the edge-set between $Z$ and $X$.
➤ All these edges must be directed from $Z$ to $X$.

The reason for considering this set is that even though the number of parameters is only proportional to the number of arcs in the model we still risk the problem of overfitting:
Learning the edge set

Geiger and Heckerman (1994) give a decomposable score function for Gaussian networks that can be extended to a certain subset of hybrid BNs:

- If a discrete variable $A$ is a parent of a continuous variable that appear in a connected subgraph (consisting of only continuous variables), then $A$ should be a parent of all variables in that subgraph.

Unfortunately, this requirement is not met by the LCM structure:
Learning the edge set

Geiger and Heckerman (1994) give a decomposable score function for Gaussian networks that can be extended to a certain subset of hybrid BNs:

- If a discrete variable $A$ is a parent of a continuous variable that appear in a connected subgraph (consisting of only continuous variables), then $A$ should be a parent of all variables in that subgraph.

However, what we are actually interested in is the score-difference between:

![Diagram](image-url)
Learning the edge set

Geiger and Heckerman (1994) give a decomposable score function for Gaussian networks that can be extended to a certain subset of hybrid BNs:

- If a discrete variable $A$ is a parent of a continuous variable that appear in a connected subgraph (consisting of only continuous variables), then $A$ should be a parent of all variables in that subgraph.

Due to the decomposability of the score function we can (instead) consider the difference between:

The score-difference between the structures to the left is the same as the score difference between:
Learning the parameters

The parameters are learned by applying the EM-algorithm.

For the M-step we have:

\[ \hat{\mu}_k \leftarrow \frac{1}{\alpha_k} \sum_{j:y_j=k} \sum_m P(M = m|D_j) \mathbb{E}(Z|M = m, D_j) \]

\[ \hat{\Gamma}_k \leftarrow \frac{1}{\alpha_k} \sum_{j:y_j=k} \sum_m P(M = m|D_j) \cdot \mathbb{E} \left[ (Z - \hat{\mu}_k) (Z - \hat{\mu}_k)^T | M = m, D_j \right] \]

\[ \hat{i}_{m,i} \leftarrow \left[ \sum_{j=1}^N \frac{1}{P(M_i = m|D_j)} \mathbb{E} \left( \tilde{Z} \tilde{Z}^T | D_j, M_i = m \right) \right]^{-1} \]

\[ \hat{\theta}_{.,i} \leftarrow \frac{1}{N} \sum_{j=1}^N \left[ x_{i,j} - \sum_{m=1}^{\text{sp}(M)} P(M = m|D_j) \hat{i}_{m,i} \mathbb{E}(\tilde{Z}|D_j, M = m) \right] x_{i,j} \]

\[ \hat{\theta}_{m,i} \leftarrow \frac{1}{N} \sum_{j=1}^N P(M = m|D_j) \left[ x_{i,j} - \hat{i}_{m,i}^T \mathbb{E}(\tilde{Z}|D_j, M = m) \right] x_{i,j} \]

\[ \hat{P}(M = m|Y = y) \leftarrow \frac{1}{\alpha_y} \sum_{j:y_j=y} P(M = m|D_j) \]

For the E-step we have \((P(M = m|D_j)\) follows by Bayes rule):

\[ \mathbb{E}(Z|D_j, M = m) = \beta_{y,m} D_j \hat{x} + (I - \beta_{y,m} L_m) \mu_y ; \]

\[ \mathbb{E} \left( Z Z^T | D_j, M = m \right) = (I - \beta_{y,m} L_m) \Gamma_y + \mathbb{E}(Z|D_j, M = m) \mathbb{E}(Z|D_j, M = m)^T , \]

where \( \beta_{y,m} = (L_m \Gamma_y)^T \left( L_m \Gamma_y L_m^T + \Theta_m \right)^{-1} \).
**Experimental results: Data sets**

We have considered 15 different datasets from the UCI machine learning repository:

<table>
<thead>
<tr>
<th>Database</th>
<th>#Att</th>
<th>#Cls</th>
<th>Inst</th>
</tr>
</thead>
<tbody>
<tr>
<td>balance-scale</td>
<td>4</td>
<td>3</td>
<td>625</td>
</tr>
<tr>
<td>breast</td>
<td>10</td>
<td>2</td>
<td>683</td>
</tr>
<tr>
<td>crabs</td>
<td>5</td>
<td>4</td>
<td>200</td>
</tr>
<tr>
<td>diabetes</td>
<td>8</td>
<td>2</td>
<td>768</td>
</tr>
<tr>
<td>glass</td>
<td>9</td>
<td>7</td>
<td>214</td>
</tr>
<tr>
<td>glass2</td>
<td>9</td>
<td>2</td>
<td>163</td>
</tr>
<tr>
<td>heart</td>
<td>13</td>
<td>2</td>
<td>270</td>
</tr>
<tr>
<td>iris</td>
<td>4</td>
<td>3</td>
<td>150</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Database</th>
<th>#Att</th>
<th>#Cls</th>
<th>Inst</th>
</tr>
</thead>
<tbody>
<tr>
<td>liver</td>
<td>6</td>
<td>2</td>
<td>345</td>
</tr>
<tr>
<td>pima</td>
<td>8</td>
<td>2</td>
<td>768</td>
</tr>
<tr>
<td>sonar</td>
<td>60</td>
<td>2</td>
<td>208</td>
</tr>
<tr>
<td>tae</td>
<td>5</td>
<td>3</td>
<td>151</td>
</tr>
<tr>
<td>thyroid</td>
<td>5</td>
<td>3</td>
<td>215</td>
</tr>
<tr>
<td>vehicle</td>
<td>18</td>
<td>4</td>
<td>846</td>
</tr>
<tr>
<td>wine</td>
<td>13</td>
<td>3</td>
<td>178</td>
</tr>
</tbody>
</table>

For each dataset, the accuracy of a classifier was estimated as the percentage of correctly classifier instances using 5-fold cross validation.
Experimental results: LCM setting

For linear LCMs:

➤ We use the wrapper approach to determine the number of latent variables, $q$.
➤ Following the proposition we require that $q \leq n \cdot |\text{sp } (Y)|$.

For non-linear LCMs:

➤ We use the wrapper approach to determine the number of latent variables as well as the number of mixture components.
➤ We use $n \cdot |\text{sp } (Y)|$ as an upper bound for $q$ and $q \cdot |\text{sp } (M)| \leq N$. as a general bound.
➤ A semi-greedy search strategy is applied:
  • A greedy search for $q$ is performed, where each $q$ is scored with the maximum accuracy it achieves when coupled with all possible values of $|\text{sp } (M)|$.

For parameter learning we applied the EM-algorithm with standard parameter settings:

➤ The algorithm terminates if the relative increase in log-likelihood falls below $10^{-3}$ or after a maximum of 100 iterations.
➤ The algorithm was run with 100 restarts.
Experimental results: Other classifiers

We have compared the LCM classifiers with 9 other classifiers in order to:

➤ Evaluate the accuracy of the LCM classifier.
➤ Explore the effect of relaxing the two assumptions of the NB classifier.

The classifiers that have been considered:

- **NB**: Naive Bayes with Gaussian leaves.
- **NB/M**: NB with mixture of Gaussians.
- **FA/BIC**: NB; factor analysis; $q$ chosen according to BIC score.
- **PCA/$\lambda$**: NB; PCA; only eigenvectors with eigenvalue larger than the average were included.
- **PCA/$n$**: NB; data transformed by PCA where all $n$ eigenvectors were included.
- **CW/PCA/$n$**: One PCA/$n$ fitted per class; classification by Bayes rule.
- **CG/PCA/$n$**: Unsuper. clust. #clusters decided by BIC; One PCA/$n$ per cluster; classification by voting.
- **NB/D**: Naive Bayes with discretized data.
- **TAN/D**: TAN classifier with discretized data.
- **LCM($q$)**: Linear LCM model where $q$ was found by the wrapper approach.
- **LCM($q$/S)**: Structural learning of LCM($q$).
- **LCM($q$, $m$; $T$)**: Non-linear LCM, $q$ and $|sp(M)|$ found by wrappers. Covariances tied.
- **LCM($q$, $m$; $U$)**: As LCM($q$, $m$; $T$), but covariances untied.
- **LCM($q$, $m$; $T$/S)**: Structural learning of LCM($q$, $m$; $T$).
## Experimental results

<table>
<thead>
<tr>
<th>Database</th>
<th>NB</th>
<th>NB/M</th>
<th>FA/BIC</th>
<th>PCA/λ</th>
<th>PCA/v</th>
<th>CW/PCA/v</th>
<th>CG/PCA/v</th>
</tr>
</thead>
<tbody>
<tr>
<td>balance-scale</td>
<td>~86.0 +/− 1.4 *53.9 +/− 2.0 *69.0 +/− 1.4 ~86.0 +/− 1.4 90.9 +/− 1.2 76.0 +/− 1.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>breast</td>
<td>~96.0 +/− 0.7 97.0 +/− 0.6 ~95.0 +/− 0.8 ~94.0 +/− 0.8 ~94.0 +/− 0.8 93.0 +/− 1.1 94.0 +/− 0.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>crabs</td>
<td>*39.5 +/− 3.5 *39.5 +/− 3.5 *63.0 +/− 3.4 *83.0 +/− 2.7 93.0 +/− 1.7 94.0 +/− 1.7 94.0 +/− 1.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>diabetes</td>
<td>75.0 +/− 1.5 *72.0 +/− 1.6 *69.0 +/− 1.7 74.0 +/− 1.6 74.0 +/− 1.6 69.0 +/− 1.7 74.0 +/− 1.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>glass</td>
<td>*36.0 +/− 3.3 *65.0 +/− 3.2 *48.0 +/− 3.4 *46.0 +/− 3.4 61.0 +/− 3.3 58.0 +/− 3.4 72.0 +/− 3.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>glass2</td>
<td>*62.0 +/− 3.8 78.0 +/− 3.2 *60.0 +/− 3.8 *58.0 +/− 3.9 *65.0 +/− 3.7 67.0 +/− 3.7 72.0 +/− 3.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>heart</td>
<td>84.0 +/− 2.2 82.0 +/− 2.3 82.0 +/− 2.3 77.0 +/− 2.6 80.0 +/− 2.4 81.0 +/− 2.4 77.0 +/− 2.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>iris</td>
<td>*95.0 +/− 1.7 *92.0 +/− 2.1 *88.0 +/− 2.6 96.0 +/− 1.5 *92.0 +/− 2.1 96.0 +/− 1.5 96.0 +/− 1.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>liver</td>
<td>*58.0 +/− 2.7 *67.0 +/− 2.5 *52.0 +/− 2.7 *60.0 +/− 2.6 60.0 +/− 2.6 66.0 +/− 2.5 69.0 +/− 2.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pima</td>
<td>75.0 +/− 1.6 74.0 +/− 1.6 *68.0 +/− 1.7 74.0 +/− 1.6 74.0 +/− 1.6 70.0 +/− 1.6 72.0 +/− 1.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sonar</td>
<td>*70.0 +/− 3.2 *63.0 +/− 3.3 *51.0 +/− 3.5 *54.0 +/− 3.5 *57.0 +/− 3.4 70.0 +/− 3.2 *70.0 +/− 3.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tae</td>
<td>54.0 +/− 4.1 52.0 +/− 4.1 *39.0 +/− 4.0 *39.0 +/− 4.0 52.0 +/− 4.1 55.0 +/− 4.0 53.0 +/− 4.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>thyroid</td>
<td>96.0 +/− 1.3 95.0 +/− 1.4 94.0 +/− 1.5 90.0 +/− 2.0 89.0 +/− 2.1 95.0 +/− 1.4 89.0 +/− 2.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vehicle</td>
<td>*44.0 +/− 1.7 *59.0 +/− 1.7 *67.0 +/− 1.6 *75.0 +/− 1.5 *76.0 +/− 1.5 *66.0 +/− 1.6 *77.0 +/− 1.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>wine</td>
<td>~97.0 +/− 1.2 ~97.0 +/− 1.2 ~93.0 +/− 1.8 ~94.0 +/− 1.7 ~96.0 +/− 1.5 98.0 +/− 0.8 ~97.0 +/− 1.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Database</th>
<th>NB/D</th>
<th>TAN/D</th>
<th>LCM(q)</th>
<th>LCM(q)/S</th>
<th>LCM(q, m; T)</th>
<th>LCM(q, m; U)</th>
<th>LCM(q, m; T)/S</th>
</tr>
</thead>
<tbody>
<tr>
<td>balance-scale</td>
<td>~65.0 +/− 1.9 ~64.0 +/− 1.9 88.0 +/− 1.3 88.0 +/− 1.3 90.0 +/− 1.2 89.0 +/− 1.3 87.0 +/− 1.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>breast</td>
<td>96.0 +/− 0.7 96.0 +/− 0.7 96.0 +/− 0.7 96.0 +/− 0.7 96.0 +/− 0.7 96.0 +/− 0.7 96.0 +/− 0.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>crabs</td>
<td>94.5 +/− 3.5 *46.0 +/− 3.5 94.5 +/− 1.6 94.5 +/− 1.6 94.5 +/− 1.5 94.5 +/− 1.5 94.5 +/− 1.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>diabetes</td>
<td>75.0 +/− 1.5 75.0 +/− 1.6 75.0 +/− 1.5 75.0 +/− 1.5 75.0 +/− 1.5 75.0 +/− 1.5 72.0 +/− 1.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>glass</td>
<td>71.0 +/− 3.1 71.0 +/− 3.1 71.0 +/− 3.1 71.0 +/− 3.1 71.0 +/− 3.1 71.0 +/− 3.1 67.3 +/− 3.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>glass2</td>
<td>81.0 +/− 3.0 81.0 +/− 3.0 *66.0 +/− 3.7 *67.0 +/− 3.7 85.0 +/− 3.0 81.0 +/− 3.3 79.0 +/− 3.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>heart</td>
<td>83.0 +/− 2.2 84.0 +/− 2.2 83.0 +/− 2.2 83.0 +/− 2.2 83.0 +/− 2.2 83.0 +/− 2.2 82.0 +/− 2.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>iris</td>
<td>*93.0 +/− 2.0 *93.0 +/− 2.0 98.0 +/− 1.1 96.0 +/− 1.5 96.0 +/− 1.6 97.0 +/− 1.3 95.0 +/− 1.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>liver</td>
<td>*58.0 +/− 2.7 *58.0 +/− 2.7 74.0 +/− 1.6 74.0 +/− 1.6 75.0 +/− 1.6 75.0 +/− 1.5 73.0 +/− 1.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pima</td>
<td>76.0 +/− 1.5 76.0 +/− 1.6 76.0 +/− 1.5 76.0 +/− 1.5 76.0 +/− 1.5 84.0 +/− 2.5 83.2 +/− 2.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sonar</td>
<td>~76.0 +/− 2.9 ~75.0 +/− 2.9 81.0 +/− 2.7 ~76.0 +/− 2.9 80.0 +/− 2.8 84.1 +/− 2.5 83.2 +/− 2.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tae</td>
<td>*49.0 +/− 4.1 *51.0 +/− 4.1 56.0 +/− 4.0 53.0 +/− 4.1 61.0 +/− 4.0 58.0 +/− 4.0 56.0 +/− 4.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>thyroid</td>
<td>~92.0 +/− 1.8 ~93.0 +/− 1.7 95.0 +/− 1.4 95.0 +/− 1.4 94.0 +/− 1.6 95.0 +/− 1.4 89.0 +/− 2.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vehicle</td>
<td>*59.0 +/− 1.7 *68.0 +/− 1.6 84.0 +/− 1.3 80.0 +/− 1.4 83.0 +/− 1.3 84.0 +/− 1.3 79.0 +/− 1.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>wine</td>
<td>98.0 +/− 0.8 *96.0 +/− 1.5 100.0 +/− 0.0 99.0 +/− 0.6 99.0 +/− 1.0 98.0 +/− 0.8 97.0 +/− 1.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Average     | 73.0    | 72.0    | 74.0    | 80.9    | 83.0    | 83.2    | 81.7    |

LCM – p.26/??