TMR PHYSTA Report [1]
Neural networks for mapping features to symbols

S Stroeve, S Gielen
University of Nijmegen, The Netherlands

September 15, 1999

[1]'Principled Hybrid systems: Theory and Applications'
Research contract FMRX-CT97-0098 (DG 12 - BDCN)

Contents

1 Introduction

In a rule-based artificial intelligence (AI) system high level symbolic knowledge is used for classification or decision-support tasks. Although such a system acts in a clearly defined manner and can explain its actions to the user, it often lacks the robustness and learning capabilities which can be achieved by neural networks. The research objective of the PHYSTA (Principled HYbrid Systems: Theory and Applications) project is to develop a general methodology for integrating rule-based (symbolic) and neural network (subsymbolic) processing stages. In such a hybrid system neural modules are used to link and supplement rule-based components on the one hand, and to generate symbolic rules within a suitable confidence level, on the other hand. In the PHYSTA project the developed hybrid system methodology will be applied to the recognition of human emotions based on facial and vocal cues. As such it must be sufficiently general such that it can deal with still as well as moving images and with speech signals. In the hierarchical architectures, which are envisioned within the PHYSTA project, the (image and speech) signals are mapped to a feature set which forms the basis for a 'feature-to-symbol' mapping operation. The low-dimensional set of symbols can be used in a rule-based system to attain a solution for the overall classification task. Examples of features in the framework of an emotion recognition system are (changes of) the size of the pupil, the shape of the mouth, the redness of the skin, pitch and loadness changes of the voice and the speed of talking. In this context the symbols may express the expectation that a subset of the features is related to a certain emotional state. As an example of the subsequent processing of the symbols in rule-based systems we might envision, that the eyes of a subject indicate that (s)he is excited, without giving a clear cue of whether (s)he is happy or angry. By combining the emotional class related to the eyes with, e.g., a symbol extracted from the shape of the mouth a classification may be reached with a higher confidence level than could have been obtained by either of the subclassifications.

This report focuses on techniques for the feature-to-symbol mapping stage. Section 2 discusses generative vector quantisation, Section 3 summarizes classification trees and Section 4 discusses combination of multiple experts as well as suitable feature processors for such a framework. Section 5 discusses the suitability of the methods for the classification of emotions based on speech and facial features.

2 Symbol extraction by generative vector quantisation

The generative vector quantisation (GVQ) methodology of Westerdijk et al. (1999a) was developed to describe patterns by binary combinations of a set of unknown features. GVQ both finds the appropriate features and gives an optimal characterization of the patterns in terms of the features. As such, it is well suited for signal to feature mapping. Moreover, GVQ can be used at a higher abstraction level for the combination of features to symbols. It fits well within the symbolic/neural hybrid scheme, since the binary combination of basis vectors facilitates easy understanding of the attained mappings. These remarks will turn more transparent after the following description of GVQ.

The goal of GVQ is to find a codebook representation of data points $\mathbf{x}^\mu$ using a small set of n_f basis vectors $\{\mathbf{f}^1, \ldots, \mathbf{f}^{n_f}\}$. Each codebook vector $\mathbf{v}$ is a binary combination of the basis vectors

$$\mathbf{v} = \sum_{i=1}^{n_f} s_i \mathbf{f}^i \equiv \mathbf{F}\mathbf{s} \tag{1}$$

with states $s_i \in \{0, 1\}$, $\mathbf{s}$ the state vector and $\mathbf{F}$ the basis matrix with features. There are thus $M = 2^{n_f}$ possible codebook vectors. Typically, each data point $\mathbf{x}^\mu$ is associated with a particular codebook vector, indexed by c_μ, such that the squared Euclidean distance E between the data set and the codebook representation is minimized:

$$E = \sum_{\mu=1}^{P} ||\mathbf{x}^\mu - \mathbf{F}\mathbf{s}^{c_\mu}||^2 \tag{2}$$

Westerdijk et al. (1999a) proposed to a two-step iterative optimization scheme to determine the optimal association between data points and codebook vectors and, the best set of basis vectors:

$$\text{Step 1:} \qquad \forall\mu: \quad c_\mu = \text{argmin}_j ||\mathbf{x}^\mu - \mathbf{F}\mathbf{s}^j||^2 \tag{3}$$

$$\text{Step 2:} \qquad \mathbf{F} = \text{argmin}_{\tilde{\mathbf{F}}} \sum_\mu ||\mathbf{x}^\mu - \mathbf{F}\mathbf{s}^{c^\mu}|| \tag{4}$$

The EM-algorithm is based on this iterative procedure and provides a fast convergence to the optimal features and codebook vectors. Using some additional learning heuristics good results could be obtained by GVQ for the horizontal and vertical bar problem, for feature extraction of handwritten digits and for an image compression task. For more details, see Westerdijk et al. (1999a).

3 Explained classification by decision trees

A decision tree is represented as a directed acyclic graph comprising internal and terminal nodes. All internal nodes contain splits, which test the value of a mathematical expression of the attributes. In a univariate decision tree the test uses a single attribute, whereas multiple attributes are employed in a multivariate decision tree. As a special case of multivariate trees stand oblique trees, which use linear combinations of the attributes. Each terminal node has a class label associated with it. The nodes are connected by edges. The construction algorithm tries to find a rule at each node which splits the data such that the entropy or a related quantity is minimized at each node. Construction of a tree from a training set (tree induction) usually proceeds in a greedy top-down fashion (Murthy, 1995):

- If all the training examples at the current node t belong to category c, create a leaf node with class c and halt.

- Otherwise,

  1. score each one of the sets of possible splits using a goodness measure;

  2. choose the best split s as the test at the current node, and create as many child nodes as there are distinct outcomes of s;

  3. label edges between the parent and child nodes with outcomes of s, and partition the training data into the child nodes.

Several criteria can be used to define the splits at a node. The goodness of a split can be evaluated by defining an impurity function $I(t)$ for every internal node t. Optimal splits are such that the decrease in impurity $\Delta I(s,t)$ due to a split s which divides the node t into a left child t_l with a proportion p_l of the data and a right child t_r with a proportion p_r of the data:

$$\Delta I(s,t) = I(t) - I(t_l)p_l - I(t_r)p_r \tag{5}$$

is minimized. Different impurity functions may be applied. Breiman et al. (1984) uses the Gini index in their CART system:

$$It) = \sum_j \sum_{i \neq j} p(i|t)p(j|t) \tag{6}$$

with $p(i|t)$ the conditional probability for class i at node t. Quinlan (1993) uses the mutual information to find the maximum information gain at a split to decide on the goodness of a split. In lazy learning algorithms (as IGTree of Daelemans et al. (1997)) similarity of a new feature to stored features is used to find the nearest neighbors of the new instance. The classes associated with the nearest neighbors are then used to predict the class in the new feature. Results of Daelemans et al. (1997) on linguistic classification tasks indicate that an accuracy similar to Quilan's C4.5 can be achieved whereas the speed is improved considerably.

Greedy tree induction results in a tree, which is prone to overfitting of the data and a pruning stage is necessary to obtain a decision tree which generalizes well. In the pruning stage the size of the tree is reduced by optimizing a cost-complexity measure which takes into account both the estimated misclassification rate and a complexity measure of the tree. Proper pruning requires an independent test set, cross-validation methods, or bootstrap techniques.

Techniques for developing decision trees are available for dealing with attributes with continuous values, large sets of discrete values, and in cases in which training examples are missing values for attributes. The basic decision tree methods are widely used and a large literature exists to assist the practitioner in its application (see e.g. Quinlan, 1993; Breiman et al., 1984).

In general, AI techniques for mapping subsymbolic data into symbols by classification or decision trees are very effective when sufficient data is available or when explicit expert knowledge is available. However, these techniques have problems with noisy data or inconsistent data. The latter typically occurs when data are missing such that two apparently identical data sets correspond to different classes because crucial data are missing, which would discriminate between the two data sets.

4 Features to symbols by combination of experts

By definition, the output of the feature-to-symbol processing stage in a hybrid reasoning system does not consist of one symbol which is uniquely related to the overall classification problem. Rather, it specifies a series of symbols which give cues about the overall class and which need to be combined properly to extract the class. Due to this synthesis the performed classification can be explained to the user, prior knowledge can be applied in the reasoning system, and specialized rather than very general feature-to-symbol extractors can be used.

A straightforward architecture of the feature-to-symbol processing stage consists of a series of mappings which are specialized experts for subsets of the features. Consider a classification problem based upon a feature space which is divided in m subspaces. In the i-th subspace n_i features are represented in a vector $\mathbf{u}_i$. Based upon the total feature space m classification processes are carried out which estimate m posterior class probabilities $\mathbf{x}_i$ as well as confidence levels $\mathbf{c}_i$. Based on the confidence levels of the classifiers these class probabilities can be combined to find the overall classification $\mathbf{y}$ plus an overall confidence measure $\mathbf{c}$.

Section 4.1 discusses methods for combining experts. Some architectures obeying the requirements for expert combination are subsequently discussed in Section 4.2.

4.1 Expert combination

4.1.1 Expert combination by supra Bayesian methods

In supra Bayesian methods (Jacobs, 1995) a decision maker (DM) combines the probability assessments of the experts with its own prior distribution via Bayes' law. If θ is the quantity of interest, m the number of experts, $P_i = p_i(\theta|H_i)$ the probability distribution of expert i if its knowledge is H_i, $p(\theta|H)$ the DM's prior probability given H, and $p(P_1, \ldots, P_m|\theta, H)$ the likelihood of the experts' opinions, then the posterior probability for θ given the knowledge H and the experts' opinions $P_1, \ldots, P_m$ can be calculated by Bayes' rule

$$p(\theta|H, P_1, \ldots, P_m) \propto p(P_1, \ldots, P_m|\theta, H)p(\theta|H) \tag{7}$$

Consider a classification process for the event A $(\theta = 1)$ or the non-event $\bar{A}$ $(\theta = 0)$ where the DM uses only a single expert which generates a probability assessment $P_1 = p(A|H_1)$. By using the DM's belief of the probability of P_1 given the event A and the knowledge H, $p(P_1|A, H)$ we can calculate the posterior probability for A by Bayes' rule

$$p(A|H, P_1) \propto p(P_1|A, H)p(A|H) \tag{8}$$

Dividing the equation for A by the similar equation for $\bar{A}$ and taking the logarithm gives the log-odds for event A given knowledge H and P_1

$$\mathrm{lo}(A|H, P_1) \equiv \log \frac{p(A|H, P_1)}{p(\bar{A}|H, P_1)} = \log \frac{p(P_1|A, H)}{p(P_1|\bar{A}, H)} + \log \frac{p(A|H)}{p(\bar{A}|H)} \tag{9}$$

The posterior log-odds thus equals the sum of the prior log-odds, denoted $\mathrm{lo}(A|H)$, and the log-likelihood ratio for A versus $\bar{A}$. Thus, if the DM believes that P_1 is indicative of the event A, then the log-likelihood ratio is positive and the posterior log-odds are larger than the prior log-odds.

If we define the log-odds of the expert as

$$q_1 \equiv \mathrm{lo}(A|H_1) = \log \frac{p(A|H_1)}{p(\bar{A}|H_1)} = \log \frac{P_1}{1 - P_1} \tag{10}$$

then the posterior log-odds of the DM can also be expressed as

$$\mathrm{lo}(A|H, q_1) = \log \frac{p(q_1|A, H)}{p(q_1|\bar{A}, H)} + \mathrm{lo}(A|H) \tag{11}$$

Following the common assumption that the log-odds for q_1 are normally distributed with variance σ^2 and averages μ_1 for A and μ_0 for $\bar{A}$, the posterior log-odds are

$$\mathrm{lo}(A|H, q_1) = \frac{\mu_1 - \mu_0}{\sigma^2}(q_1 - \frac{\mu_1 + \mu_0}{2}) + \mathrm{lo}(A|H) \tag{12}$$

The expert's log-odds q_1 are corrected by a bias term $(\mu_1+\mu_0)/2$ which is zero in the symmetric case if $\mu_1 = -\mu_0$. This corrected value is weighted by $(\mu_1 - \mu_0)/\sigma^2$ and added to the prior log-odds, meaning that a large difference in the means and a small variance results in a large influence on the posterior log-odds. This result illustrates the necessity for information about the confidence levels for A and $\bar{A}$. Methods for estimating confidence levels of the feature processing networks are discussed shortly in Section 4.2 and is explored in more detail in (Stroeve et al., 1999).

This view can be easily extended to the case of multiple experts. If the vector $\mathbf{q}$ denotes the experts' log-odds and it is assumed that they are normally distributed with means $\mathbf{m}_1$ and $\mathbf{m}_0$ and covariance matrix $\mathbf{C}$, then the posterior log-odds are

$$\mathrm{lo}(A|H,\mathbf{q}) = [\mathbf{q} - (\mathbf{m}_1 + \mathbf{m}_0)/2]^T \mathbf{C}^{-1}(\mathbf{m}_1 - \mathbf{m}_0) + \mathrm{lo}(A|H) \tag{13}$$

Also the general case of multiple exclusive events can be handled by supra Bayesian techniques. According to (Jacobs, 1995) supra Bayesian techniques have been applied successfully in a general statistical framework, but this paper does not discuss applications with neural network classifiers.

4.1.2 Expert combination by linear opinion pools

In a linear opinion pool the probability distribution of the decision maker (DM) is a linear combination of the experts' opinions

$$p(\theta) = \sum_{i=1}^{m} w_i P_i \quad \text{with} \sum_{i=1}^{m} w_i = 1 \tag{14}$$

In such a linear opinion pool, the weights (1) can be viewed as veridical probabilities or (2) are the result from a regression of the probability of θ against the experts' opinions.

In the first approach the DM assumes that θ is generated by one of the experts' distributions $P_1, \ldots, P_m$. The weight w_i is the probability that P_i is the true distribution. The veridical assumption is, e.g., used in the mixtures-of-experts (ME) architecture of Jacobs et al. (1991) which consists of a gating network and several expert networks. Note that the veridical assumption is not correct if one wants to combine multiple classifiers which all provide a reasonable class prediction. This approach is therefore not useful in the context of the PHYSTA project.

In second approach, which might be useful in the PHYSTA context, the weights of the linear opinion pool are optimized such that the variance of the overall prediction is minimized. If $\hat{\theta}_i$ is the estimate of θ made by the i-th expert, and the experts' predictions are unbiased, the weighted average of the experts' opinion is unbiased

$$\theta = E(\hat{\theta}) = E(\sum_{i=1}^{m} w_i \hat{\theta}_i) \tag{15}$$

The weights w_i can now be selected such that they minimize the variance of $\hat{\theta}$ under the condition that they sum to one. The solution of this optimization problem is

$$\mathbf{w} = \mathbf{C}^{-1}\mathbf{1}(\mathbf{1}^T\mathbf{C}^{-1}\mathbf{1})^{-1} \tag{16}$$

with $\mathbf{C}$ the covariance matrix for the experts' errors and $\mathbf{1}$ a vector with all ones. Further extensions of this linear opinion pool are discussed by Jacobs (1995) and Perrone (1993).

Both the variance reducing linear opinion pools and the supra Bayesian framework are probabilistic methods for combining experts' knowledge which may be applied in the PHYSTA project. In practice they both require knowledge on the covariance between the experts. Moreover, the supra Bayesian approach is suited to include prior knowledge.

4.2 Feature processors

Section 4.1 indicates that combination of experts' opinions requires at least knowledge of the variance of the experts if they are independent, or of the covariance matrix of the different experts if their errors are dependent. Optimally, the experts represent the conditional probability distributions of the feature-to-symbol mappings. A logical choice for the outputs of the feature-to-symbol mappings are class probabilities, e.g., the probability of anger or happiness. However, the outputs might also represent continuous variables, e.g., the activation, evaluation and angle values on Plutchnik's emotion wheel (Cowie, 1998). This section discusses feature processing networks which fulfill at least the minimum requirements and might thus be applied in hybrid reasoning systems.

4.2.1 Multilayer perceptron for regression

Placed in a probabilistic context (Bishop, 1995; MacKay, 1992a,b,c,d) the multilayer perceptron (MLP) network is a suitable feature processor in a hybrid reasoning system. A short summary of the Bayesian view on the MLP is presented here.

Given a single output network $y(\mathbf{x}, \mathbf{w})$ with input vector $\mathbf{x}$ and weight vector $\mathbf{w}$. The prior distribution of the weights $p(\mathbf{w})$ can be written as

$$p(\mathbf{w}) = \frac{1}{Z_W(\alpha)} \exp(-\alpha E_W) \tag{17}$$

with $Z_W(\alpha)$ a normalization constant such that $\int p(\mathbf{w}) d\mathbf{w} = 1$. Proper generalization of the network can be achieved by favouring small weights which is operationalized by choosing E_W as

$$E_W = \frac{1}{2}\|\mathbf{w}\|^2 = \frac{1}{2}\sum_{i=1}^{W} w_i^2 \tag{18}$$

with W the number of weights in the network. For this prior distribution the normalization constant is

$$Z_W(\alpha) = (2\pi/\alpha)^{W/2} \tag{19}$$

Given a regression problem with a target set $D = \{t^1, t^2, \ldots, t^N\}$ and assuming that the target variables t^n are generated by a deterministic function with added Gaussian noise, the probability distribution of the target variables is

$$\begin{aligned} p(t|\mathbf{x}) &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(y(\mathbf{x};\mathbf{w}) - t)^2}{2\sigma^2}) & (20) \\ &\propto \exp(-\frac{\beta}{2}(y(\mathbf{x};\mathbf{w}) - t)^2) & (21) \end{aligned}$$

with σ^2 the variance of the noise process and $\beta = 1/\sigma^2$. If the data points are drawn independently from this distribution, the conditional probability distribution $p(D|\mathbf{w})$ is

$$\begin{aligned} p(D|\mathbf{w}) &= \prod_{n=1}^{N} p(t^n|\mathbf{x}^n, \mathbf{w}) && (22) \\ &= \frac{1}{Z_D(\beta)} \exp(-\frac{\beta}{2} \sum_{n=1}^{N} \{y(\mathbf{x}^n; \mathbf{w}) - t^n\}^2) && (23) \\ &= \frac{1}{Z_D(\beta)} \exp(-\beta E_D) && (24) \end{aligned}$$

with E_D an error function and $Z_D(\beta)$ a normalization constant which is given by

$$Z_D(\beta) = (2\pi/\beta)^{N/2} \tag{25}$$

Now Bayes theorem can be applied to obtain the probability distribution of the weights:

$$\begin{aligned} p(\mathbf{w}|D) &= \frac{p(D|\mathbf{w})p(\mathbf{w})}{p(D)} && (26) \\ &= \frac{p(D|\mathbf{w})p(\mathbf{w})}{\int p(D|\mathbf{w})p(\mathbf{w})d\mathbf{w}} && (27) \\ &= \frac{1}{Z_S(\alpha, \beta)} \exp(-\beta E_D - \alpha E_W) = \frac{1}{Z_S(\alpha, \beta)} \exp(-S(\mathbf{w})) && (28) \end{aligned}$$

with $Z_S(\alpha, \beta)$ again a normalization constant.

The (most probable) weight vector $\mathbf{w}_{MP}$ which maximizes the posterior distribution of Eq. 28 is the weight vector which minimizes $S(\mathbf{w})$. For the Gaussian noise model of the target data and the prior distribution of the weights this comes down to minimization of

$$S(\mathbf{w}) = \frac{\beta}{2} \sum_{n=1}^{N} \{y(x^n; \mathbf{w}) - t^n\}^2 + \frac{\alpha}{2} \sum_{i=1}^{W} w_i^2 \tag{29}$$

meaning minimization of the sum-of-squares error function with a weight-decay regularization term.

In general, the normalization constant $Z_S(\alpha, \beta)$ cannot be evaluated analytically and simplifying approximations are needed to evaluate the posterior distribution. To obtain such an approximation the Taylor expansion of $S(\mathbf{w})$ around its minimum $\mathbf{w}_{MP}$ is truncated for terms higher than second-order:

$$S(\mathbf{w}) = S(\mathbf{w}_{MP}) + \frac{1}{2}(\mathbf{w} - \mathbf{w}_{MP})^T \mathbf{A}(\mathbf{w} - \mathbf{w}_{MP}) \tag{30}$$

where the matrix $\mathbf{A}$ is the Hessian of the error function S:

$$\mathbf{A} = \nabla^2 S_{MP} = \beta \nabla^2 E_D^{MP} + \alpha \mathbf{I} \tag{31}$$

Using this approximation, the posterior distribution is

$$p(\mathbf{w}|D) = \frac{1}{Z_S^*} \exp\{-S(\mathbf{w}_{MP}) - \frac{1}{2} \Delta\mathbf{w}^T \mathbf{A} \Delta\mathbf{w}\} \tag{32}$$

where the normalization constant is

$$Z_S^* = \exp\{-S(\mathbf{w}_{MP})\}(2\pi)^{W/2} \det(A)^{-1/2} \tag{33}$$

Confidence levels Using the Gaussian approximation to the posterior probability distribution and assuming that the network function $y(\mathbf{x}, \mathbf{w})$ is well represented by a linear expansion around $\mathbf{w}_{MP}$

$$y(\mathbf{x}; \mathbf{w}) = y(\mathbf{x}; \mathbf{w}_{MP}) + \mathbf{g}^T \Delta\mathbf{w} \tag{34}$$

with

$$\mathbf{g} = \nabla_{\mathbf{w}} y|_{\mathbf{w}_{MP}} \tag{35}$$

the distribution of the output for an input vector $\mathbf{x}$ may be represented by a Gaussian distribution:

$$p(t|\mathbf{x}, D) = \frac{1}{\sqrt{2\pi\sigma_t^2}} \exp(-\frac{(t - y_{MP})^2}{2\sigma_t^2}) \tag{36}$$

where

$$\sigma_t^2 = 1/\beta + \mathbf{g}^T \mathbf{A}^{-1} \mathbf{g} \tag{37}$$

The standard deviation σ_t is a measure for the confidence level of the mean value y_{MP}.

4.2.2 MLP for classification

In a classification problem the network output vector represents class probabilities. In a two-class problem a sigmoidal activation function $y = g(a) = 1/(1 + \exp(-a))$ is appropriate. The output y represents the posterior probabilities $P(t = 1|\mathbf{x})$ of class C_1 and $P(t = 0|\mathbf{x}) = 1 - y$ of class C_2. The probability of observing either target value is

$$p(t|\mathbf{x}) = y^t (1 - y)^{1-t} \tag{38}$$

and the likelihood function for the data is now given by

$$p(D|\mathbf{w}) = \prod_n y(\mathbf{x}^n)^{t^n} (1 - y(\mathbf{x}^n))^{1-t^n} = \exp(-G(D|\mathbf{w})) \tag{39}$$

where G is the cross-entropy error function:

$$G(D|\mathbf{w}) = -\sum_n \{t^n \ln(y(\mathbf{x}^n)) + (1 - t^n) \ln(1 - y(\mathbf{x}^n))\} \tag{40}$$

For a multiclass problem a 'softmax' network with probabilistic outputs y_i and a cross-entropy function G are appropriate:

$$y_i = \frac{e^{a_i}}{\sum_j e^{a_j}} \tag{41}$$

$$G = \sum_m \sum_i t_i \log y_i(\mathbf{x}^m; \mathbf{w}) \tag{42}$$

Using a regularization term E_W for the weights the posterior distribution is

$$p(\mathbf{w}|D) = \frac{1}{Z_S} \exp(-G - \alpha E_W) = \frac{1}{Z_S} \exp(-S(\mathbf{w})) \tag{43}$$

For a binary classification task, the probability of membership of class C_1 for a new input vector $\mathbf{x}$ is

$$P(C_1|\mathbf{x}, D) = \int P(C_1|\mathbf{x}, \mathbf{w})p(\mathbf{w}|D)d\mathbf{w} = \int y(\mathbf{x}; \mathbf{w})p(\mathbf{w}|D)d\mathbf{w} \tag{44}$$

Assuming a Gaussian approximation of the posterior distribution $p(\mathbf{w}|D)$ centered around the maximum posterior weight vector $\mathbf{w}_{MP}$, and assuming that a is a local linear function of the weights, it can be derived that

$$p(a|\mathbf{x}, D) = \frac{1}{\sqrt{2\pi s^2}} \exp(-\frac{(a - a_{MP})^2}{2s^2}) \tag{45}$$

where the variance s^2

$$s^2(\mathbf{x}) = \mathbf{g}^T \mathbf{A}^{-1} \mathbf{g} \tag{46}$$

is a function of the Hessian matrix $\mathbf{A} = \nabla^2_{\mathbf{w}} S$ and the gradient vector $\mathbf{g} = \nabla_{\mathbf{w}} a$. The probability of class C_1 now is

$$P(C_1|\mathbf{x}, D) = \int g(a)p(a|\mathbf{x}, D)da \tag{47}$$

which may be approximated by

$$P(C_1|\mathbf{x}, D) \approx g(\kappa(s)a_{MP}) \tag{48}$$

with

$$\kappa(s) = (1 + \pi s^2/8)^{-1/2} \tag{49}$$

For a multi-class task no analytical approximation for the integral over a Gaussian posterior distribution is known (MacKay, 1996).

4.2.3 Radial basis function network

A radial basis function (RBF) network with basis functions determined by a mixture density estimation procedure can be a useful architecture in a hybrid reasoning system. The mapping of a RBF network with input $\mathbf{x}$, output $\mathbf{y}$, weights w_{kj} and basis functions ϕ_j is given by

$$y_k(\mathbf{x}) = \sum_{j=1}^{M} w_{kj}\phi_j(\mathbf{x}) + w_{k0} \tag{50}$$

In this context the density of the input data is modeled by a mixture model:

$$p(\mathbf{x}) = \sum_{j=1}^{M} P(j)\phi_j(\mathbf{x}) \tag{51}$$

with $\phi_j(\mathbf{x})$ a basis function and $P(j)$ a mixing coefficient (Bishop, 1995). The mixing coefficient $P(j)$ can be regarded as the prior probability of the data points having been generated by the j-th component of the mixture. Therefore, the mixing coefficients should satisfy the constraints:

$$\sum_{j=1}^{M} P(j) = 1 \tag{52}$$

$$0 \leq P(j) \leq 1 \tag{53}$$

The basis functions can be regarded as class-conditional densities:

$$\phi_j(\mathbf{x}) = p(\mathbf{x}|j) \tag{54}$$

A common choice for the basis function is the following Gaussian distribution:

$$\phi_j(\mathbf{x}) = \frac{1}{(2\pi\sigma_j^2)^{d/2}} \exp(-\frac{\|\mathbf{x} - \mu_j\|^2}{2\sigma_j^2}) \tag{55}$$

with d the dimension of $\mathbf{x}$, and μ_j and σ_j^2 the center and variance of the j-th Gaussian. The mixture model parameters are determined by maximizing the likelihood function

$$L = \prod_n p(\mathbf{x}^n) \tag{56}$$

using a gradient or an expectation-maximization algorithm (Bishop, 1995).

After the parameters of the basis functions have been determined, the weights w_{kj} in the second layer of the RBF network (Eq. 50) can be determined by a supervised learning procedure. Combination of the mixture model of Eq. 51 and the RBF network of Eq. 50 now provides a network prediction together with an estimate of the likelihood of the input vector (Bishop, 1994). A low likelihood of the input vector corresponds with a low confidence in the network prediction.

4.2.4 Boltzmann machines and mixture models

Rather than an estimate of the conditional probability distribution which can be represented by a feedforward neural network, a Boltzmann machine (BM) can form a representation of the joint probability distribution. The standard Boltzmann machine (Ackley et al., 1985) consists of all-to-all coupled stochastic, binary neurons, which are divided in input, hidden and output neurons. With input vector $\mathbf{u}$, hidden states $\mathbf{x}$, output states $\mathbf{y}$ and a symmetric weights matrix $\mathbf{W}$, the joint probability of the combined state vector $\mathbf{s}$ is given by a Gibbs distribution:

$$p(\mathbf{u}, \mathbf{x}, \mathbf{y}) = p(\mathbf{s}) = \frac{1}{Z} \exp(\beta \sum_i \sum_{j \neq i} w_{ij} s_i s_j) \tag{57}$$

where Z is the partition function, which normalizes the probability distribution. The weights can be adapted such that the Kullback divergence between the probability distributions of the Boltzmann machine and the data-generating process is minimized. The resulting learning process is however very slow. Extensions of the standard Boltzmann machine concern (1) higher order connections, such as $w_{ijk} s_i s_j s_k$, to attain better approximations of general probability distributions (e.g., Leisink and Kappen, 1998), (2) neurons with more than two states, which are known as Potts neurons (Bollé et al., 1992), and (3) extensions of the Boltzmann machine architecture such it can deal with continuous input signals: the Boltzmann perceptron (BP) and the radial basis Boltzmann machines (RBBM) (Kappen, 1995). Since, within the PHYSTA-project, binary as well as continuous features should be accounted for, the extensions proposed by Kappen (1995) would be necessary. The BP can only represent conditional probability distributions (similar to a feedforward network) and the methods of Heskes et al. (1995) can be applied to obtain confidence intervals. The RBBM can represent joint probability distributions and thus can be readily applied to obtain confidence levels or to support active decision processes. Both the BP and the RBBM use inhibitory connections in the output layer such that the number of possible states and the calculation time are reduced. Figure 1 shows the architecture of the RBBM. It describes the probability distribution $p(\mathbf{x}, \mathbf{y}, \mathbf{s})$ of visible continuous units x_i, visible binary units y_k and hidden binary units s_j. The joint probability distribution is given by

$$p(\mathbf{x}, \mathbf{y}, \mathbf{s}) = \frac{1}{Z} \exp\{-\beta \sum_{j=1}^{h} \|\mathbf{x} - \mathbf{w}_j\|^2 s_j + \sum_{j=1}^{h} \sum_{k=1}^{m} v_{jk} s_j y_k - J(\sum_{j=1}^{h} s_j - h_0)^2\} \tag{58}$$

where β controls the width of the Gaussians, $\mathbf{w}_j$ are the centers of the Gaussians, v_{jk} are the weights between $\mathbf{s}$ and $\mathbf{y}$ and the lateral inhibition term $J(\sum_{j=1}^{h} s_j - h_0)^2$ controls the number of units y_k which may be 'on' simultaneously.

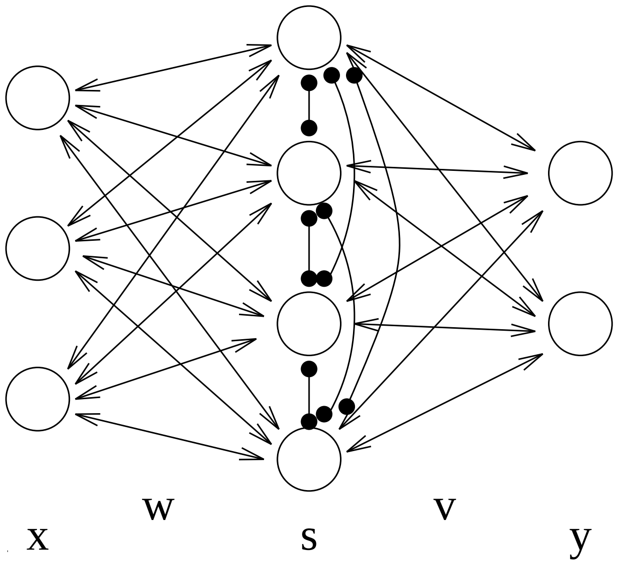

Figure 1: *Radial basis Boltzmann machine for approximation of the joint probability distribution between real-valued units x_i and binary-valued units y_k using hidden binary-valued neurons s_j. Weights $\mathbf{w}$ and $\mathbf{v}$ connect the layers. The hidden layer has lateral inhibition which limits the number of simultaneous high states s_j.*

Closely related to the RBBM is the mixture model applied by Westerdijk et al. (1998). Their joint probability model forms the basis for an active decision process which determines the most informative input signal if not all input data are readily known (Westerdijk et al., 1998). Active decision means the process of finding the most informative input i to make a good classification. If the joint probability $p(\mathbf{x}, \mathbf{y})$ is encoded during the training phase the most informative input i given knowledge of the input $\mathbf{x}_d$ is found by minimizing the expected entropy of the class distribution:

$$\begin{aligned} &\min_i \int p(x_i|\mathbf{x}_d) S(x_i, \mathbf{x}_d) dx_i = \\ &\min_i - \int p(x_i|\mathbf{x}_d) \sum_k p(y_k|x_i, \mathbf{x}_d) \ln p(y_k|x_i, \mathbf{x}_d) dx_i \end{aligned} \tag{59}$$

with $S(x_i, \mathbf{x}_d)$ the information contained in the input $(x_i, \mathbf{x}_d)$. Westerdijk et al. (1998) have shown, that this approach

1. gives a better classification performance than expert systems;

2. gives a better selection of the most informative input units which contribute to correct classification;

3. gives simpler rules than expert systems.

5 Discussion

A range of techniques suitable as feature processors in a hybrid reasoning system combining symbolic (rule based) and subsymbolic (neural network) processing were discussed in this report. In the PHYSTA project such methods are applied on the recognition of emotions based on vocal and visual cues. At the time of production of this report the PHYSTA emotion database with facial and speech data was still under development and only a limited speech database was available to test the discussed techniques. The limited speech database contains 196 feature vectors with 32 potentially relevant speech characteristics associated with 5 emotions. Given the close interaction between the estimation of confidence levels and the classification of emotions, results regarding this speech database are discussed in the PHYSTA report on confidence levels and on line retraining (Stroeve et al., 1999).

Generative Vector Quantisation (GVQ), discussed in Section 2, can be used to represent symbols or class labels as binary combinations of input features. It was applied successfully to image processing tasks as feature extraction from a pallet of horizontal and vertical bars and from handwritten digits (Westerdijk et al., 1999a,b). In this line, it may also be well suited for the extraction of (higher order) features from facial image data in the context of emotion classification. It is, however, unlikely that GVQ is suitable for the extraction of higher order features from speech data, since an emotion type is expected to depend in continuous rather than binary fashion on the speech features.

A decision tree classifier provides direct insight into its decision strategy and therefore in general is a useful candidate for a feature to symbol mapping. However, the large number of features and the small dataset available for learning in the present PHYSTA setting implies that the number of data points on which a split of the tree has to be decided becomes very small only after a few nodes. Thus, the decision tree will be strongly prone to overfitting. It is therefore not useful for the speech feature processing.

Classification by multilayer perceptron (MLP) networks and the estimation of confidence levels could be obtained for the limited speech data set (Stroeve et al., 1999). Best results were obtained for MLPs trained in a Bayesian framework. About 53% correct classification could be obtained for the five emotions. Although this rate is not high, empirical studies on human recognition of emotions based on speech cues also show far from perfect recognition rates of about 65%. Therefore, it is likely that the combination of visual and auditory signals are essential for good emotion discrimination by humans. Appropriate techniques for expert combination were discussed in Section 4.1. Since at the time of production of this report only speech data were available, combination of classifiers based on different signal modalities (speech and vision) was not feasible yet. The essential combination of these signals will be discussed in future PHYSTA publications.

References

Ackley DH, Hinton GE, Sejnowski TJ (1985) A learning algorithm for Boltzmann machines. Cognitive Science 9:147-169

Bishop CM (1994) Novelty detection and neural network validation. IEE Proceedings on Vision and Image Signal Processing 141(4):217-222

Bishop CM (1995) Neural networks for pattern recognition. Oxford University Press, Oxford, England

Bollé D, Dupont P, Huyghebaert J (1992) Thermodynamic properties of the Q-state Potts-glass neural network. Physical Review A 45(6):4194-4197

Breiman L, Friedman JH, Olshen, Stone CJ (1984) Classification and regression trees. Wadsworth Int., Belmont (CA), USA

Cowie R (1998) Review of existing techniques for human emotion understanding and applications in human-computer interaction. TMR PHYSTA report.

Daelemans W, Bosch A van de, Weijters T (1997) IGTree: using trees for compression and classification in lazy learning algorithms. Artificial Intelligence Review 11: 407-423

Heskes T, Kappen B, Pastoors A, Gielen C (1995) Confidence intervals for neural networks. DSP95 396-401

Jacobs RA (1995) Methods for combining experts' probability assessments. Neural Computation 7:867-888

Jacobs RA, Jordan MA, Nowlan SJ, Hinton GE (1991) Adaptive mixtures of local experts. Neural Computation 3:79-87

Kappen HJ (1995) Deterministic learning rules for Boltzmann machines. Neural Networks 8(4):537-548

Leisink MAR, Kappen HJ (1998) Learning higher order Boltzmann Machines using linear response. Proceedings of the 8-th International Conference on Artificial Neural Networks

MacKay, DJC (1992a) Bayesian interpolation. Neural Computation 4(3):415-447

MacKay, DJC (1992b) The evidence framework applied to classification networks. Neural Computation 4(5):720-736

MacKay, DJC (1992c) Information-based objective functions for active data selection. Neural Computation 4(4):590-604

MacKay, DJC (1992d) A practical Bayesian framework for backpropagation networks. Neural Computation 4(3):448-472

MacKay DJC (1996) Bayesian methods for backpropagation networks. In: Domany E, Hemmen JL van, Schulten K; Models of neural networks III. Springer-Verlag, New-York, USA

Murthy KVS (1995) On growing better decision trees from data. PhD-thesis, John Hopkins University, Baltimore (MD), USA

Perrone MP (1993) Improving regression estimation: Averaging methods for variance reduction with extensions to general convex measure optimization. PhD-thesis, Department of Physics, Brown University

Quinlan JR (1993) C4.5: programs for machine learning. Morgan Kaufmann Publishers, San Mateo (CA), USA

Stroeve S, Gielen S, Doulamis A, Doulamis N, Kollias S, Cowie R, Cowie-Douglas E, McGilloway S (1999) Confidence estimation and on line retraining of neural networks. TMR PHYSTA report.

Westerdijk MJD, Gielen CCAM, Wiegerinck W (1998) Comparing mixture models with decision trees for case specific feature selection. Submitted

Westerdijk M, Barber D, Wiegerinck W (1999a) Feature discovery using generative vector quantisation. Submitted

Westerdijk M, Barber D, Wiegerinck W (1999b) Generative vector quantisation for feature discovery and classification. Submitted