ELSEVIER

Pattern Recognition Letters
journal homepage: www.elsevier.com

# Supplementary Material: Derivations and Pseudocode for Learning Maximum Margin Hidden Markov Models for Sequence Classification

Nikolaus Mutsam[a], Franz Pernkopf[a,**]

[a] *Graz University of Technology, Laboratory of Signal Processing and Speech Communication, Graz, Austria*

ABSTRACT

This supplement includes all derivations and the pseudocode for learning the maximum margin hidden Markov model for sequence classification. It uses the extended Baum-Welch framework for optimization.

© 2016 Elsevier Ltd. All rights reserved.

## 1. Maximum Margin Parameter Estimation

The multi-class margin Guo *et al.* (2005); Pernkopf *et al.* (2012) of sample $n$ is

$$\begin{aligned}\tilde{d}^n_{\mathbf{\Theta}} &= \min_{c\neq c^n} \frac{p\left(c^n|\mathbf{x}^n,\mathbf{\Theta}\right)}{p\left(c|\mathbf{x}^n,\mathbf{\Theta}\right)} = \min_{c\neq c^n} \frac{p\left(c^n,\mathbf{x}^n|\mathbf{\Theta}\right)}{p\left(c,\mathbf{x}^n|\mathbf{\Theta}\right)} \\ &= \frac{p(\mathbf{x}^n|\mathbf{\Theta}_{c^n})\rho_{c^n}}{\max_{c\neq c^n} p(\mathbf{x}^n|\mathbf{\Theta}_c)\rho_c}. \qquad (1)\end{aligned}$$

If $\tilde{d}^n_{\mathbf{\Theta}} > 1$, then sample $n$ is correctly classified and vice versa. We replace the max operator by the differentiable approximation $\max_x f(x) \approx \left[\sum_x (f(x))^\eta\right]^{\frac{1}{\eta}}$, where $\eta \geq 1$ and $f(x)$ is non-negative. In the limit of $\eta \to \infty$ the approximation converges to the maximum operator. Replacing the maximum with its approximation, we obtain

$$d^n_{\mathbf{\Theta}} = \frac{p(\mathbf{x}^n|\mathbf{\Theta}_{c^n})\rho_{c^n}}{\left[\sum_{c\neq c^n}\left(p(\mathbf{x}^n|\mathbf{\Theta}_c)\rho_c\right)^\eta\right]^{\frac{1}{\eta}}}. \qquad (2)$$

Usually, the maximum margin approach maximizes the margin of the sample with the smallest margin, i.e. $\min_{n=1,\ldots,N} d^n_{\mathbf{\Theta}}$ for a separable classification problem Schölkopf & Smola (2001). We aim to relax this by introducing a soft margin, i.e. we focus on samples with a $d^n_{\mathbf{\Theta}}$ close to one. Therefore, we consider the *hinge* loss function according to

$$\tilde{J}(\mathcal{X}|\mathbf{\Theta}) = \prod_{n=1}^{N} \min\left[\kappa, d^n_{\mathbf{\Theta}}\right], \qquad (3)$$

where parameter $\kappa > 1$ controls the influence of the margin $d^n_{\mathbf{\Theta}}$ in the hinge loss $\tilde{J}(\mathcal{X}|\mathbf{\Theta})$ and is set by cross-validation. Maximizing this function with respect to the parameters $\mathbf{\Theta}$ implicitly means to increase the margin $d^n_{\mathbf{\Theta}}$ whereas the emphasis is on samples with a margin $d^n_{\mathbf{\Theta}} < \kappa$, i.e. samples with a large positive margin have no impact on the optimization. Maximizing $\tilde{J}(\mathcal{X}|\mathbf{\Theta})$ via EBW or gradient descent is not straight forward due to the discontinuity in the derivative at $d^n_{\mathbf{\Theta}} = \kappa$. Therefore, we propose to use for the *hinge* function $h(y) = \min[\kappa, y]$ a *smooth hinge* function which enables a smooth transition of the derivative and has a similar shape as $h(y)$. We propose the following function inspired by the Huber loss Huber (1964). In particular, we approximate the discontinuity by a circle segment as

$$h(y) = \begin{cases} y + \frac{1}{2}, & \text{if } y \leq \kappa - 1 \\ \kappa - \frac{1}{2}(y-\kappa)^2, & \text{if } \kappa - 1 < y < \kappa \\ \kappa, & \text{if } y \geq \kappa \end{cases} \qquad (4)$$

which requires to divide the data $\mathcal{X}$ into three partitions depending on $y = d^n_{\mathbf{\Theta}}$, i.e. $\mathcal{X}^1$ contains samples where $d^n_{\mathbf{\Theta}} \leq \kappa - 1$, $\mathcal{X}^2$ consists of samples with a margin in the range $\kappa - 1 < d^n_{\mathbf{\Theta}} < \kappa$, and $\mathcal{X}^3 = \mathcal{X} \setminus \left\{\mathcal{X}^1 \cup \mathcal{X}^2\right\}$.

Basically, there are other smoothing techniques available for non-smooth convex objectives, e.g. Nesterov (2005). In our case, smoothing of the objective function makes it amenable for gradient-based optimization methods while still approximating the original objective well. Experiments using a similar *parametrized* smooth hinge function show only a slight influence on performance for maximum margin Bayesian network classifiers Pernkopf *et al.* (2012). Furthermore, a similar approximation of the maximum margin objective outperforms a convex formulation (which requires relaxation of constraints) with respect to computational requirements, while the classifi-

[**] Corresponding author: Tel.: +43-316-873-4436; fax: +43-316-873-104436;
*e-mail:* pernkopf@tugraz.at (Franz Pernkopf)

cation performance is almost identical.

Using the smooth hinge function in (4), our objective function for margin maximization is

$$J(\mathcal{X}|\mathbf{\Theta}) = \prod_{n=1}^{N} h(d_{\mathbf{\Theta}}^{n}) \tag{5}$$
$$=\left\{\prod_{n\in\mathcal{X}^1}\left(d_{\mathbf{\Theta}}^{n}+\frac{1}{2}\right)\right\}\left\{\prod_{n\in\mathcal{X}^2}\left[\kappa-\frac{1}{2}\left(d_{\mathbf{\Theta}}^{n}-\kappa\right)^2\right]\right\}\kappa^{|\mathcal{X}^3|}.$$

### 1.1. Optimization of the Margin Objective

The EBW algorithm is an iterative procedure which can be used to optimize rational functions Gopalakrishnan *et al.* (1991). We use the EBW framework to optimize the margin objective in (5) for the discrete model parameters $\rho_c, \pi_{c,i}, a_{c,i,j}, \alpha_{c,i,m}$. The parameter re-estimation equation of the form

$$\theta_i^j \leftarrow \frac{\theta_i^j\left(\frac{\partial \log J(\mathcal{X}|\mathbf{\Theta})}{\partial \theta_i^j}+D\right)}{\sum_l \theta_l^j\left(\frac{\partial \log J(\mathcal{X}|\mathbf{\Theta})}{\partial \theta_l^j}+D\right)}, \tag{6}$$

is used, where $\theta_i^j \geq 0$, $\sum_i \theta_i^j = 1$, and $j$ indicates a particular discrete variable. EBW requires the partial derivative $\frac{\partial \log J(\mathcal{X}|\mathbf{\Theta})}{\partial \mathbf{\Theta}}$ and $D$. Both terms are provided in the sequel. Specifically, the derivative $\frac{\partial \log J(\mathcal{X}|\mathbf{\Theta})}{\partial \mathbf{\Theta}}$ for the re-estimation equation (6) of the EBW algorithm is

$$\frac{\partial \log J(\mathcal{X}|\mathbf{\Theta})}{\partial \mathbf{\Theta}} = \sum_{n=1}^{N} s^n \frac{\partial \log d_{\mathbf{\Theta}}^{n}}{\partial \mathbf{\Theta}} \tag{7}$$

where $s^n$ denotes a sample dependent weight given as follows:

$$s^n = \begin{cases} \frac{d_{\mathbf{\Theta}}^{n}}{d_{\mathbf{\Theta}}^{n}+\frac{1}{2}}, & \text{if } n \in \mathcal{X}^1 \\ \frac{\kappa d_{\mathbf{\Theta}}^{n}-(d_{\mathbf{\Theta}}^{n})^2}{\kappa-\frac{1}{2}(d_{\mathbf{\Theta}}^{n}-\kappa)^2}, & \text{if } n \in \mathcal{X}^2 \\ 0, & \text{if } n \in \mathcal{X}^3 \end{cases} . \tag{8}$$

Approximating $p(\mathbf{x}|\mathbf{\Theta}_c)$ with the probability of the most probable state sequence of the Viterbi algorithm, i.e.

$$p(\mathbf{x}|\mathbf{\Theta}_c) \approx p^*(\mathbf{x}|\mathbf{\Theta}_c) = \pi_{c,q_1^*} b_{c,q_1^*}(\mathbf{x}_1) \prod_{t=2}^{T} a_{c,q_{t-1}^*,q_t^*} b_{c,q_t^*}(\mathbf{x}_t), \tag{9}$$

the log of the margin $d_{\mathbf{\Theta}}^{n}$ of sample $\mathbf{x}^n$ in Eq. (2) decomposes to

$$\log d_{\mathbf{\Theta}}^{n} = \log(p(\mathbf{x}^n|\mathbf{\Theta}_{c^n})\rho_{c^n}) - \frac{1}{\eta}\log\sum_{c'\neq c^n}(p(c',\mathbf{x}^n|\mathbf{\Theta}_{c'})\rho_{c'})^\eta$$
$$=\log\pi_{c^n,i^{*,n}_{c^n,1}}+\sum_{t=1}^{T^n}\log b_{c,i^{*,n}_{c^n,t}}(\mathbf{x}_t^n)+\sum_{t=2}^{T^n}\log a_{c,i^{*,n}_{c^n,t-1},i^{*,n}_{c^n,t}}+\log\rho_{c^n}$$
$$-\frac{1}{\eta}\log\Big[\sum_{c'\neq c^n}^{C}\Big(\pi_{c',i^{*,n}_{c',1}}\prod_{t=1}^{T^n} b_{c',i^{*,n}_{c',t}}(\mathbf{x}_t^n)\prod_{t=2}^{T^n} a_{c',i^{*,n}_{c',t-1},i^{*,n}_{c',t}}\rho_{c'}\Big)^\eta\Big], \tag{10}$$

where $i^{*,n}_{c^n,t}$ is the most probable state of the HMM of class $c^n$ for a sequence $\mathbf{x}^n$ at time $t$.

the derivative for $\rho_c$ of $\frac{\partial \log d_{\mathbf{\Theta}}^{n}}{\partial \mathbf{\Theta}}$ in (7) is

$$\frac{\partial \log d_{\mathbf{\Theta}}^{n}}{\partial \rho_c} = \frac{\mathbb{1}_{\{c=c^n\}}}{\rho_c} - \frac{\mathbb{1}_{\{c\neq c^n\}}(p(\mathbf{x}^n|\mathbf{\Theta}_c)\rho_c)^{\eta-1}p(\mathbf{x}^n|\mathbf{\Theta}_c)}{\sum_{c'\neq c^n}(p(\mathbf{x}^n|\mathbf{\Theta}_{c'})\rho_{c'})^\eta}\frac{\rho_c}{\rho_c}$$
$$= \frac{1}{\rho_c}\left[\mathbb{1}_{\{c=c^n\}} - \mathbb{1}_{\{c\neq c^n\}}\frac{(p(\mathbf{x}^n|\mathbf{\Theta}_c)\rho_c)^\eta}{\sum_{c'\neq c^n}(p(\mathbf{x}^n|\mathbf{\Theta}_{c'})\rho_{c'})^\eta}\right]$$
$$= \frac{1}{\rho_c}\left[z_c^n - \check{z}_c^n \cdot r_c^{n,\eta}\right], \tag{11}$$

where

$$r_c^{n,\eta} = \frac{(p(\mathbf{x}^n|\mathbf{\Theta}_c)\rho_c)^\eta}{\sum_{c'\neq c^n}(p(\mathbf{x}^n|\mathbf{\Theta}_{c'})\rho_{c'})^\eta}, \tag{12}$$

$$z_c^n = \mathbb{1}_{\{c=c^n\}} \quad \text{and} \tag{13}$$

$$\check{z}_c^n = \mathbb{1}_{\{c\neq c^n\}}. \tag{14}$$

Symbol $\mathbb{1}_{\{i=j\}}$ denotes the indicator function (i.e. equals 1 if the Boolean expression $i = j$ is true and 0 otherwise).

Furthermore, the partial derivatives of $\log d_{\mathbf{\Theta}}^{n}$ with respect to $\pi_i$, $a_{c,i,j}$ and $\alpha_{c,i,m}$ are given as follows:

$$\frac{\partial \log d_{\mathbf{\Theta}}^{n}}{\partial \pi_{c,i}} = \frac{1}{\pi_{c,i}}\left[u_{c,i,1}^n - \check{u}_{c,i,1}^n \cdot r_c^{n,\eta}\right] \tag{15}$$

$$\frac{\partial \log d_{\mathbf{\Theta}}^{n}}{\partial a_{c,i,j}} = \frac{1}{a_{c,i,j}}\left[y_{c,i,j}^n - \check{y}_{c,i,j}^n \cdot r_c^{n,\eta}\right] \tag{16}$$

$$\frac{\partial \log d_{\mathbf{\Theta}}^{n}}{\partial \alpha_{c,i,m}} = \frac{1}{\alpha_{c,i,m}}\sum_{t=1}^{T^n}\left[\gamma_{c,i,m,t}^n\left(u_{c,i,t}^n - \check{u}_{c,i,t}^n \cdot r_c^{n,\eta}\right)\right], \tag{17}$$

where

$$u_{c,i,t}^n = \mathbb{1}_{\{c=c^n,i=i^{*,n}_{c,t}\}} \tag{18}$$

$$\check{u}_{c,i,t}^n = \mathbb{1}_{\{c\neq c^n,i=i^{*,n}_{c,t}\}} \tag{19}$$

$$y_{c,i,j}^n = \sum_{t=2}^{T^n}\mathbb{1}_{\{c=c^n,i=i^{*,n}_{c,t-1},j=i^{*,n}_{c,t}\}} \tag{20}$$

$$\check{y}_{c,i,j}^n = \sum_{t=2}^{T^n}\mathbb{1}_{\{c\neq c^n,i=i^{*,n}_{c,t-1},j=i^{*,n}_{c,t}\}} \tag{21}$$

and

$$\gamma_{c,i,m,t}^n = \frac{\alpha_{c,i,m}\cdot\mathcal{N}(\mathbf{x}_t^n|\boldsymbol{\mu}_{c,i,m},\mathbf{\Sigma}_{c,i,m})}{\sum_{m'=1}^{M}\alpha_{c,i,m'}\cdot\mathcal{N}(\mathbf{x}_t^n|\boldsymbol{\mu}_{c,i,m'},\mathbf{\Sigma}_{c,i,m'})}. \tag{22}$$

### 1.2. Approximation of the Gradient

The derivatives (11), (15), (16) and (17) are sensitive to small parameter values. Merialdo Merialdo (1988) observed that low-valued parameters $\rho_c$, $\pi_{c,i}$, $a_{c,i,m}$ and $\alpha_{c,i,m}$ may cause a large magnitude of the gradient and the optimization concentrates on those parameters. However, small parameter values indicate that they are rarely used during the production of an observation sequence. Hence, there is not sufficiently training data available for reliably estimating very low probabilities and concentrating on low-valued parameters is unreliable. Therefore, he suggests to focus on modifying better estimated high-valued parameters during optimization by using an approximation of the gradients. In particular, for gradients of the form $\frac{\partial \log d^n_{\mathbf{\Theta}}}{\partial \theta^j_i} = \frac{1}{\theta^j_i}(c_{i,j} - c'_{i,j})$, as in our case, he suggests to concentrate on high-valued parameters by replacing the gradient by

$$\frac{\partial \log d^n_{\mathbf{\Theta}}}{\partial \theta^j_i} \approx \frac{c_{i,j}}{\sum_j c_{i,j}} - \frac{c'_{i,j}}{\sum_j c'_{i,j}}. \tag{23}$$

This approximation of the gradients has been used for CLL learning in Normandin & Morgera (1991); Normandin *et al.* (1994). Unfortunately, approximating the gradient by (23) cannot be applied to the derivatives of the margin, because the approximated gradient disappears for any HMM parameter. Therefore, we suggest an alternative approximation in order to obtain reliable parameter updates. Since the unreliability of the updates is caused by small parameter values due to high values of the gradients Merialdo (1988), normalizing the gradient by a sum-to-one constraint of the absolute gradient values keeps the updates reliable. For gradients of the form $\frac{\partial \log d^n_{\mathbf{\Theta}}}{\partial \theta^j_i} = \frac{1}{\theta^j_i}(c_{i,j} - c'_{i,j})$, we propose to approximate the gradient by

$$\frac{\partial \log d^n_{\mathbf{\Theta}}}{\partial \theta^j_i} \approx \frac{\frac{1}{\theta^j_i}(c_{i,j} - c'_{i,j})}{\sum\limits_{i'=1}^{S} \left| \frac{1}{\theta^j_{i'}}(c_{i',j} - c'_{i',j}) \right|}. \tag{24}$$

The resulting approximations of the derivatives in (11), (15), (16) and (17) are provided in the algorithm for maximum margin (MM) training of HMMs in Appendix B. As an alternative, Woodland and Povey Woodland & Povey (2002) proposed an alternative mixture weight update rule using an iterative procedure.

### *1.3. Approximation for the Gaussians*

EBW has been formulated for discrete probability distributions. Normandin and Morgera Normandin & Morgera (1991) introduced a discrete approximation of the Gaussian distribution assuming diagonal covariance matrices. This leads to the re-estimation equation for $\boldsymbol{\mu}_{c,i,m}$ and $\mathbf{\Sigma}_{c,i,m}$ given as

$$\bar{\boldsymbol{\mu}}_{c,i,m} \leftarrow \frac{\sum\limits_{n=1}^{N} s^n \sum\limits_{t=1}^{T^n} \left[\gamma^n_{c,i,m,t}(u^n_{c,i,t} - \check{u}^n_{c,i,t} \cdot r^{n,\eta}_c)\mathbf{x}^n_t\right] + D\boldsymbol{\mu}_{c,i,m}}{\sum\limits_{n=1}^{N} s^n \sum\limits_{t=1}^{T^n} \left[\gamma^n_{c,i,m,t}(u^n_{c,i,t} - \check{u}^n_{c,i,t} \cdot r^{n,\eta}_c)\right] + D} \tag{25}$$

and

$$\bar{\mathbf{\Sigma}}_{c,i,m} \leftarrow \tag{26}$$
$$\frac{g_{c,i,m} + D(\mathbf{\Sigma}_{c,i,m} + (\boldsymbol{\mu}_{c,i,m})^2)}{\sum\limits_{n=1}^{N} s^n \sum\limits_{t=1}^{T^n} \left[\gamma^n_{c,i,m,t}(u^n_{c,i,t} - \check{u}^n_{c,i,t} \cdot r^{n,\eta}_c)\right] + D} - (\bar{\boldsymbol{\mu}}_{c,i,m})^2,$$

where $g_{c,i,m} = \sum\limits_{n=1}^{N} s^n \sum\limits_{t=1}^{T^n} \left[\gamma^n_{c,i,m,t}(u^n_{c,i,t} - \check{u}^n_{c,i,t} \cdot r^{n,\eta}_c)(\mathbf{x}^n_t)^2\right]$ and the squares of $\mathbf{x}^n_t$ and $\boldsymbol{\mu}_{c,i,m}$ are taken element-wise.

### *1.4. Implementation of the MM-HMM EBW Algorithm*

The EBW algorithm converges to a local optimum of $J(\mathcal{X}|\mathbf{\Theta})$ providing a sufficiently large value for $D$. Setting the constant $D$ is not trivial. If it is chosen too large then training is slow and if it is too small the update may fail to increase the objective function. In practical implementations heuristics have been suggested Woodland & Povey (2002); Klautau *et al.* (2003); Pernkopf & Wohlmayr (2010).

In order to obtain positive covariances, the inequality

$$\frac{g_{c,i,m,d} + D(\sigma_{c,i,m,d} + \mu^2_{c,i,m,d})}{h_{c,i,m} + D} - \left(\frac{k_{c,i,m,d} + D\mu_{c,i,m,d}}{h_{c,i,m} + D}\right)^2 > 0 \tag{27}$$

must hold for any covariance $\sigma_{c,i,m,d}$ of dimension $d \in \mathcal{D}$, where

$$h_{c,i,m} = \sum_{n=1}^{N} s^n \sum_{t=1}^{T^n} \left[\gamma^n_{c,i,m,t}(u^n_{c,i,t} - \check{u}^n_{c,i,t} \cdot r^{n,\eta}_c)\right] \tag{28}$$

and

$$k_{c,i,m} = \sum_{n=1}^{N} s^n \sum_{t=1}^{T^n} \left[\gamma^n_{c,i,m,t}(u^n_{c,i,t} - \check{u}^n_{c,i,t} \cdot r^{n,\eta}_c)\mathbf{x}^n_t\right]. \tag{29}$$

Rearranging (27) leads to a quadratic inequality with respect to $D$ Valtchev *et al.* (1997):

$$\begin{aligned} &\underbrace{\sigma_{c,i,m,d}}_{a} D^2 \\ &+ \underbrace{(\sigma_{c,i,m,d}h + \mu^2_{c,i,m,d} + g_{c,i,m,d} - 2k_{c,i,m,d}\mu_{c,i,m,d})}_{b} D \\ &+ \underbrace{g_{c,i,m,d}h - k^2_{c,i,m,d}}_{c} > 0 \end{aligned} \tag{30}$$

We propose to set

$$D = F \cdot \max\{D_1, D_2, D_3\}, \tag{31}$$

where

$$D_{1,2} = \frac{-(b) \pm \sqrt{b^2 - 4ac}}{2a} \quad \text{and} \tag{32}$$

$$D_3 = 1 + \left| \min_{i,j} \frac{\partial \log J(\mathcal{X}|\mathbf{\Theta})}{\partial \theta^j_i} \right|. \tag{33}$$

$D_3$ guarantees a positive parameter after the update in (6) and $F > 1$ regulates the convergence speed of the algorithm. The parameters $\mathbf{\Theta}$ for discriminative learning are initialized to the MLE of the HMM determined by the EM algorithm. Generative model pre-training can be seen as a form of regularization Erhan *et al.* (2010). The class prior is set to the normalized class frequency in $\mathcal{X}$, i.e. $\rho_c = \frac{N_c}{N}$. A detailed algorithm of maximum margin (MM) training for HMM is provided in Appendix B.

## Appendix B: MM-HMM EBW Algorithm

The implementation of the EBW algorithm for maximizing the margin, i.e. MM-HMM EBW algorithm, is stated in Algorithm 1.

The E-step of the MM-HMM EBW algorithm using the approximation of $\frac{\partial \log d^n_{\mathbf{\Theta}}}{\partial \mathbf{\Theta}}$ (see Eq. (24)) is depicted in Algorithm 2.

In Algorithm 3, the M-step of the MM-HMM EBW algorithm using parameter updates of Eq. (6) is illustrated.

```
Input: $\{\mathcal{X}_1, \dots \mathcal{X}_C\}$
Output: $\rho_c, \pi_{c,i}, a_{c,i,j}, \alpha_{c,i,m}, \boldsymbol{\mu}_{c,i,m}, \boldsymbol{\Sigma}_{c,i,m} \quad \forall c \in \{1, \dots, C\},\ \forall i, j \in \{1, \dots, S\}; \forall m \in \{1, \dots, M\}$
Initialization: For each c, $\boldsymbol{\Theta}_c = \{\pi_{c,i}, a_{c,i,j}, \alpha_{c,i,m}, \boldsymbol{\mu}_{c,i,m}, \boldsymbol{\Sigma}_{c,i,m}\}_{i,j\in\{1,\dots,S\},m\in\{1,\dots M\}}$ is
                initialized by MLE using the EM-algorithm. The class prior is set to
                the normalized class frequency, i.e. $\rho_c = \frac{N_c}{N}$
while $J(\mathcal{X}|\boldsymbol{\Theta})$ not converged do
    Determine: $\mathcal{X}^1, \mathcal{X}^2, \mathcal{X}^3$ based on $(d^n_\Theta)^\lambda$
    Determine: $s^n \quad \forall n \in \{1, \dots, N\}$ based on $\mathcal{X}^1, \mathcal{X}^2, \mathcal{X}^3$
    E-Step (see Algorithm 2)
    Determine D (see Section 1.4)
    M-Step (see Algorithm 3)
end
```

*Algorithm 1:* Discriminative Margin-based training of HMMs (MM-HMM EBW algorithm).

```
E-Step:
for $c \leftarrow 1$ to $C$ do
    $r_c^{n,\eta} \leftarrow \frac{(p(\mathbf{x}^n|\boldsymbol{\Theta}_c)\rho_c)^\eta}{\sum_{c' \neq c^n} (p(\mathbf{x}^n|\boldsymbol{\Theta}_{c'})\rho_{c'})^\eta}$
    $\frac{\partial log d^n_{\boldsymbol{\Theta}}}{\partial \rho_c} \leftarrow \frac{\frac{1}{\rho_c}\left[z_c^n - \check{z}_c^n \cdot r_c^{n,\eta}\right]}{\sum_{c'=1}^C \left|\frac{1}{\rho_{c'}}\left[z_c^n - \check{z}_c^n \cdot r_c^{n,\eta}\right]\right|}$
    $\partial \rho_c \leftarrow \sum_{n=1}^N s^n \frac{\partial \log d^n_{\boldsymbol{\Theta}}}{\partial \rho_c}$
    for $i \leftarrow 1$ to $S$ do
        $\frac{\partial \log d^n_{\boldsymbol{\Theta}}}{\partial \pi_{c,i}} \leftarrow \frac{\frac{1}{\pi_{c,i}}\left[u^n_{c,i,1} - \check{u}^n_{c,i,1} \cdot r_c^{n,\eta}\right]}{\sum_{i'=1}^S \left|\frac{1}{\pi_{c,i'}}\left[u^n_{c,i',1} - \check{u}^n_{c,i',1} \cdot r_c^{n,\eta}\right]\right|}$
        $\partial \pi_{c,i} \leftarrow \sum_{n=1}^N s^n \frac{\partial \log d^n_{\boldsymbol{\Theta}}}{\partial \pi_{c,i}}$
        for $j \leftarrow 1$ to $S$ do
            $\frac{\partial \log d^n_{\boldsymbol{\Theta}}}{\partial a_{c,i,j}} \leftarrow \frac{\frac{1}{a_{c,i,j}}\left[y^n_{c,i,j} - \check{y}^n_{c,i,j} \cdot r_c^{n,\eta}\right]}{\sum_{j'=1}^S \left|\frac{1}{a_{c,i,j'}}\left[y^n_{c,i,j'} - \check{y}^n_{c,i,j'} \cdot r_c^{n,\eta}\right]\right|}$
            $\partial a_{c,i,j} \leftarrow \sum_{n=1}^N s^n \frac{\partial \log d^n_{\boldsymbol{\Theta}}}{\partial a_{c,i,j}}$
        end
        for $m \leftarrow 1$ to $M$ do
            $\gamma^n_{c,i,m,t} \leftarrow \frac{\alpha_{c,i,m} \cdot \mathcal{N}(\mathbf{x}^n_t|\boldsymbol{\mu}_{c,i,m}, \boldsymbol{\Sigma}_{c,i,m})}{b_{c,i}(\mathbf{x}^n_t)}$
                $\forall n \in \{1, \dots, C\}$
            $\frac{\partial \log d^n_{\boldsymbol{\Theta}}}{\partial \alpha_{c,i,m}} \leftarrow \frac{\frac{1}{\alpha_{c,i,m}} \sum_{t=1}^{T^n} \left[\gamma^n_{c,i,m,t}\left(u^n_{c,i,t} - \check{u}^n_{c,i,t} r_c^{n,\eta}\right)\right]}{\sum_{m'=1}^M \left|\frac{1}{\alpha_{c,i,m'}} \sum_{t=1}^{T^n} \left[\gamma^n_{c,i,m',t}\left(u^n_{c,i,t} - \check{u}^n_{c,i,t} r_c^{n,\eta}\right)\right]\right|}$
            $\partial \alpha_{c,i,m} \leftarrow \sum_{n=1}^N s^n \frac{\partial \log d^n_{\boldsymbol{\Theta}}}{\partial \alpha_{c,i,m}}$
        end
    end
end
```

*Algorithm 2:* E-step of the MM-HMM EBW algorithm.

```
M-Step:
for $c \leftarrow 1$ to $C$ do
    $\bar{\rho}_c \leftarrow \frac{\rho_c\left(\partial \rho_c + D\right)}{\sum_{c'=1}^C \rho_{c'}\left(\partial \rho_{c'} + D\right)}$
    for $i \leftarrow 1$ to $S$ do
        $\bar{\pi}_{c,i} \leftarrow \frac{\pi_{c,i}\left(\partial \pi_{c,i} + D\right)}{\sum_{i'=1}^S \pi_{c',i}\left(\partial \pi_{c',i} + D\right)}$
        for $j \leftarrow 1$ to $S$ do
            $\bar{a}_{c,i,j} \leftarrow \frac{a_{c,i,j}\left(\partial a_{c,i,j} + D\right)}{\sum_{j'=1}^S a_{c,i,j'}\left(\partial a_{c,i,j'} + D\right)}$
        end
        for $m \leftarrow 1$ to $M$ do
            $\bar{\alpha}_{c,i,m} \leftarrow \frac{\alpha_{c,i,m}\left(\partial \alpha_{c,i,m} + D\right)}{\sum_{m'=1}^M \alpha_{c,i,m'}\left(\partial \alpha_{c,i,m'} + D\right)}$
            $\bar{\boldsymbol{\mu}}_{c,i,m} \leftarrow \frac{\sum_{n=1}^N s^n \sum_{t=1}^{T^n} \left[\gamma^n_{c,i,m,t}\left(u^n_{c,i,t} - \check{u}^n_{c,i,t} \cdot r_c^{n,\eta}\right)\mathbf{x}^n_t\right] + D\boldsymbol{\mu}_{c,i,m}}{\sum_{n=1}^N s^n \sum_{t=1}^{T^n} \left[\gamma^n_{c,i,m,t}\left(u^n_{c,i,t} - \check{u}^n_{c,i,t} \cdot r_c^{n,\eta}\right)\right] + D}$
            $\boldsymbol{\Sigma}_{c,i,m} \leftarrow \frac{g_{c,i,m} + D\left(\boldsymbol{\Sigma}_{c,i,m} + (\boldsymbol{\mu}_{c,i,m})^2\right)}{\sum_{n=1}^N s^n \sum_{t=1}^{T^n} \left[\gamma^n_{c,i,m,t}\left(u^n_{c,i,t} - \check{u}^n_{c,i,t} \cdot r_c^{n,\eta}\right)\right] + D} - (\bar{\boldsymbol{\mu}}_{c,i,m})^2$
            $\boldsymbol{\mu}_{c,i,m} \leftarrow \bar{\boldsymbol{\mu}}_{c,i,m}$
        end
        $\alpha_{c,i,m} \leftarrow \bar{\alpha}_{c,i,m} \quad \forall m$
        $a_{c,i,j} \leftarrow \bar{a}_{c,i,j} \quad \forall j$
    end
    $\pi_{c,i} \leftarrow \bar{\pi}_{c,i} \quad \forall i$
end
$\rho_c \leftarrow \bar{\rho}_c \quad \forall c$
```

*Algorithm 3:* M-step of the MM-HMM EBW algorithm.

## 2. Acknowledgements

This work was supported by the Austrian Science Fund (Project number P27803-N15).

## References

Baum, L.E., & Eagon, J.A. 1967. An inequality with applications to statistical prediction for functions of Markov processes and to a model of ecology. *Bull. Amer. Math. Soc.*, **73**, 360–363.

Erhan, D., Bengio, Y., Courville, A., Manzagol, P.-A., Vincent, P., & Bengio, S. 2010. Why Does Unsupervised Pre-training Help Deep Learning? *Journal of Machine Learning Research*, **11**, 625–660.

Gopalakrishnan, O., Kanevsky, D., Nàdas, A., & Nahamoo, D. 1991. An inequality for rational functions with applications to some statistical estimation problems. *IEEE Transactions on Information Theory*, **37**(1), 107–113.

Guo, Y., Wilkinson, D.F., & Schuurmans, D. 2005. Maximum Margin Bayesian Networks. *Pages 233–242 of: International Conference on Uncertainty in Artificial Intelligence (UAI).*

Huber, P.J. 1964. Robust Estimation of a Location Parameter. *Annals of Statistics*, **53**, 73–101.

Klautau, A., Jevtić, N., & Orlitsky, A. 2003. Discriminative Gaussian Mixture models: A comparison with kernel classifiers. *Pages 353 – 360 of: Inter. Conf. on Machine Learning (ICML).*

Merialdo, B. 1988. Phonetic recognition using hidden Markov models and maximum mutual information training. *Pages 111–114 of: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP).*

Nesterov, Y. 2005. Smooth minimization of non-smooth functions. *Mathematical programming*, **103**(1), 127–152.

Normandin, Y., & Morgera, S.D. 1991. An improved MMIE training algorithm for speaker-independent small vocabulary, continuous speech recognition. *Pages 537–540 of: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP).*

Normandin, Y., Cardin, R., & De Mori, R. 1994. High-performance connected digit recognition using maximum mutual information estimation. *IEEE Trans. on Speech and Audio Proc.*, **2**(2), 299–311.

Pernkopf, F., & Wohlmayr, M. 2010. Large Margin Learning of Bayesian Classifiers based on Gaussian Mixture Models. *Pages 50–66 of: European Conference on Machine Learning (ECML).*

Pernkopf, F., Wohlmayr, M., & Tschiatschek, S. 2012. Maximum Margin Bayesian Network Classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **34**(3), 521–532.

Schölkopf, B., & Smola, A.J. 2001. *Learning with kernels: Support Vector Machines, regularization, optimization, and beyond.* MIT Press.

Valtchev, V., Odell, J. J., Woodland, P.C., & Young, S.J. 1997. MMIE training of large vocabulary recognition systems. *Pages 303–314 of: Speech Communication*, vol. 22.

Woodland, P.C., & Povey, D. 2002. Large scale discriminative training of hidden Markov models for speech recognition. *Computer Speech and Language*, **16**, 25–47.