GMs in On-Line Handwritten Whiteboard Note Recognition: The Influence of Implementation and Modeling

Joachim Schenk and Benedikt Hörnler and Björn Schuller and Artur Braun and Gerhard Rigoll
Institute for Human-Machine Communication
Technische Universität München
Theresienstraße 90, 80333 München
{schenk,hoernler,schuller,braun,rigoll}@mmk.ei.tum.de

Abstract

We present a comparison of two state-of-the-art toolboxes for implementing Graphical Models (GMs), namely the HTK and the GMTK, and their use for discrete on-line handwritten whiteboard note recognition. We then motivate a GM that is capable of modeling the statistical dependencies between the pen's pressure information and the remaining features after vector quantization. Since the number of variable parameters rises when more codebook entries are used for quantization, the proposed model outperforms standard HMMs for low numbers of codebook entries.

1. Introduction

Hidden-Markov-Models (HMMs, see [1]) are used for modeling time-dynamic sequences of variable lengths, which are common in on-line handwriting recognition (HWR). While research in on-line HMM-based HWR using a pen and some sampling device has been studied intensively during the past years, on-line HWR of whiteboard notes is a relatively new task [2]. When using HMMs for recognition, the output probabilities are estimated either in a continuous manner, e. g. by mixtures of Gaussians or discrete, i. e. the relative number of occurrences of the vector quantized features.

In [3], we proposed the use of discrete HMMs for on-line HWR of whiteboard notes for the first time and revealed an interesting fact: when using continuous HMMs the pen's pressure information is vital for recognition [4], whereas in discrete HMM-based HWR of whiteboard notes this feature looses significance. This observation has been further studied in [5] focusing on the influence of the distribution of the features on the recognition performance.

In this paper, our investigations presented in [5] are continued from a Graphical Models' (GMs, see [6]) point of view. Thereby GMs are a natural enhancement of HMMs: By combining probability theory and graph theory, a visual graphical language, and efficient algorithms are provided for probability calculations and decision making.

This paper has two main contributions. First, two state-of-the-art toolboxes for HMM- and GM-based pattern recognition are compared, namely the "Hidden-Markov-Toolkit" (HTK, see [7]) and the "Graphical Models Toolkit" (GMTK, see [8]). Second, the findings of our earlier results on the pen's pressure information presented in [3, 5] are confirmed using GMs. Furthermore, the question is answered, whether the statistical dependencies between the pen's pressure and the remaining features should be modeled within the vector quantizer, as presented in [3] or by statistical inference using GMs. The next section briefly sketches our recognition system, and summarizes vector quantization. Section 3 describes the GMs used in this paper. In Sec. 4, the previously introduced GMs are evaluated, and an explanation for the observed effects is given. An outlook and a conclusion can be found in Sec. 5.

2. System Overview

In this section, we summarize our recognition system, and roughly explain vector quantization.

Recognition System The handwritten data, which is recorded with the EBEAM-System and represented by sample vectors $\mathbf{s}(t)$, is heuristically segmented into lines [2]. Then, preprocessing and normalization is performed, and features are extracted from the sample vector, and form a 24-dimensional feature vector $\mathbf{f}_t = (f_{1,t}, \ldots, f_{24,t})$. The features listed below can be divided into two groups: *on-line* and *off-line* features. While the continuous on-line features are derived from the pen's trajectory, the discrete off-line features evaluate a bitmap gained by binarization of the handwritten script. The on-line features are: f_1 : indicating the pen's "pressure", i.e. $f_1 = 1$ if the pen tip is placed on

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.127

the whiteboard and $f_1 = 0$ otherwise; f_2 : velocity equivalent; $f_{3,4}$: x-and y-coordinate (high pass filtered); $f_{5,6}$: angle α of spatially resampled and normalized strokes (coded as $\sin\,\alpha$ and $\cos\,\alpha$, "writing direction"); $f_{7,8}$: difference of consecutive angles $\Delta\alpha = \alpha(t) - \alpha(t-1)$ (coded as $\sin\,\Delta\alpha$ and $\cos\,\Delta\alpha$, "curvature"); f_9 : logarithmized aspect v of the trajectory between the points $\mathbf{s}(t-\tau)$ and $\mathbf{s}(t)$, whereby $\tau < t$ denotes the τ^{th} sample point before $\mathbf{s}(t)$: $f_9 = \text{sign}(v) \cdot \text{lg}(1 + |v|)$, where $\text{lg}(\cdot) = \log_{10}(\cdot)$; $f_{10,11}$: angle φ between the line $[\mathbf{s}(t-\tau), \mathbf{s}(t)]$ and lower line (coded as $\sin\,\varphi$ and $\cos\,\varphi$, "vicinity slope"); f_{12} : the length of trajectory normalized by the $\max(|\Delta x|; |\Delta y|)$ ("vicinity curliness") ; f_{13} : average square distance to each point and the line $[\mathbf{s}(t-\tau), \mathbf{s}(t)]$.

The off-line features are: f_{14-22} : a 3×3 subsampled bitmap slid along the pen's trajectory ("context map") to incorporate a 30×30 partition of the currently written letter's actual image; $f_{23,24}$: number of pixels above respectively beneath the current sample point $\mathbf{s}(t)$ (the "ascenders" and "descenders")

As the values of the features may vary in different ranges, each dimension d of the feature vector is normalized to a mean of $\mu_d = 0$ and variance of $\text{var}_d = 1$, yielding the features $\tilde{f}_d$. Further details on the used recognition system can be found in [9].

Vector Quantization The Graphical Models presented in this paper model discrete observations $\mathbf{o} = \{\hat{f}_1, \ldots, \hat{f}_T\}$. Therefore, the continuous features in the D-dimensional feature vectors $\mathbf{f}_t \in \mathbb{R}^D$ are mapped to codebook indices $\hat{f}_t \in \mathbb{N}$ provided by a codebook $\mathbf{C} = (\mathbf{c}_1, \ldots, \mathbf{c}_{N_{\text{cdb}}})$, $\mathbf{c}_k \in \mathbb{R}^D$ containing $|\mathbf{C}| = N_{\text{cdb}}$ centroids $\mathbf{c}_i$ [10]. For $D = 1$ this mapping is called *scalar*, and in all other cases ($D \geq 2$) *vector* quantization (VQ). Once a codebook $\mathbf{C}$ is generated, the assignment of the continuous sequence to the codebook entries is a minimum distance search. Various techniques for codebook generation exist. In this paper, we use the well-known k-Means algorithm as described e. g. in [10].

3. Graphical Models

As stated in the Introduction, Graphical Models (GMs) are a combination of probability theory and graph theory. Thus, they provide a visual graphical language and algorithms for probability calculations and decision making [6]. The GMs presented in this section are derived from standard HMMs, for which the following substitutions are common: The state transition probability $p(q_t = s_j | q_{t-1} = s_i) = a_{ij}$ for the S states s_i, s_j with $1 \leq i, j \leq S$ are summarized in the transition matrix $\mathbf{A}$. Each observation $\hat{f}_t$ is made with the observation probability $p(\hat{f}_t | s_i) = b_{s_i}(\hat{f}_t)$ given the current state $q_t = s_i$. All observation probabilities $b_{s_i}(f_t)$ are kept in the observation matrix $\mathbf{B}$. Finally, $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_S)$ substitutes the initial state distribution, i. e $p_i = p(q_1 = s_i)$. All parameters are conveniently summarized in the parameter set $\lambda = (\boldsymbol{\pi}, \mathbf{A}, \mathbf{B})$.

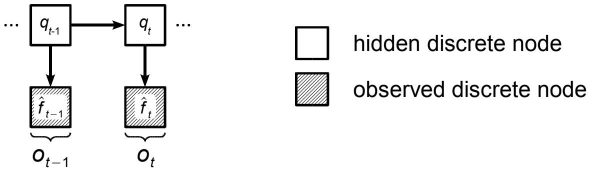

Figure 1: Standard discrete HMM in GM notation.

3.1. GMs used in this Paper

Standard HMM Figure 1 shows on the left-hand side the a standard discrete HMM in GM notation. Following the GM shown in Fig. 1, the production probability of the discrete HMM yields

$$p(\mathbf{o}, \mathbf{q} | \lambda) = p(q_1) \cdot p(o_1 | q_1) \prod_{t=2}^{T} p(q_t | q_{t-1}) \cdot p(o_t | q_t), \quad (1)$$

considering the state sequence $\mathbf{q} = (q_1, \ldots, q_T)$. By marginalizing, i. e. summing Eq. 1 over all possible state sequences $\mathbf{q} \in \mathbf{Q}$, and using the above substitutions a_{ij}, $b_{s_i}(o_t)$ and π_i, the well-known production probability, as presented e. g. in [1],

$$p(\mathbf{o} | \lambda) = \sum_{q \in \mathbf{Q}} \pi_{q_1} b_{q_1}(o_1) \prod_{t=2}^{T} a_{q_{t-1} q_t} b_{q_t}(o_t) \quad (2)$$

is derived.

Multistream HMM In [5] we introduced an HMM with multiple observation streams for on-line HWR of whiteboard notes. The corresponding GM is shown in Fig. 2 left. In the here used HMM, each observation $\mathbf{o}_t = (\tilde{f}_{1,t}, \hat{f}_{\text{r},t})$ is a vector consisting of the normalized pressure information $\tilde{f}_1$ and the remaining, vector quantized features $\hat{f}_{\text{r}}$. Taking the GM as shown on the left-hand side of Fig. 2 into account and again, marginalizing, the production probability $p(\mathbf{O} | \lambda)$ of the observation $\mathbf{O} = (\mathbf{o}_1, \ldots, \mathbf{o}_T)$ yields

$$\begin{aligned} p(\mathbf{O} | \lambda) = \sum_{q \in \mathbf{Q}} \pi_{q_1} \underbrace{p(\tilde{f}_{1,1} | q_1) p(\hat{f}_{\text{r},1} | q_1)}_{(*)} \cdot \\ \prod_{t=2}^{T} a_{q_{t-1} q_t} \underbrace{p(\tilde{f}_{1,t} | q_t) p(\hat{f}_{\text{r},t} | q_t)}_{(*)} . \end{aligned} \quad (3)$$

As indicated by $(*)$ in Eq. 3, the normalized pressure information $\tilde{f}_1$ and the vector quantized, remaining features $\hat{f}_{\text{r}}$ are modeled in a statistically independent manner.

Enhanced GM In order to model the probabilistic dependencies between the pressure information and the remaining features by statistical inference, the enhanced GM

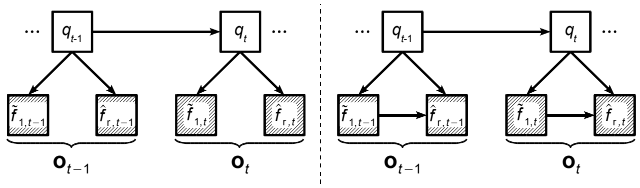

Figure 2: HMM with multiple observations in GM notation (left) and an enhanced GM for modeling the statistical dependencies between the pressure information and the remaining features (right) by statistical inference.

as depicted on the right-hand side of Fig. 2 is introduced. By adding the connection between the observation nodes in each time step, information can be transferred. The production probability for this GM is derived to

$$p(\mathbf{O}|\lambda) = \sum_{q \in \mathbf{Q}} \pi_{q_1} \underbrace{p(\tilde{f}_{1,1}|q_1)p(\hat{f}_{\mathrm{r},1}|q_1, \tilde{f}_{1,1}|q_1)}_{(**)} \cdot \prod_{t=2}^{T} a_{q_{t-1}q_t} \underbrace{p(\tilde{f}_{1,t}|q_t)p(\hat{f}_{\mathrm{r},t}|\tilde{f}_{1,t}|q_t, q_t)}_{(**)} . \quad (4)$$

In contrast to Eq. 3, in Eq. 4 the statistical dependencies between the normalized pressure information $\tilde{f}_1$ and the vector quantized, remaining features $\hat{f}_\mathrm{r}$ is taken into account, as indicated by $(**)$.

3.2. Training and Recognition

Training of the above described GMs is performed by the Expectation-Maximization (EM) algorithm, in case of the HMMs also called Baum-Welch algorithm [1, 11]. Combined recognition and segmentation is enabled by the Viterbi algorithm [1].

4. Experiments

Our experiments are conducted on the IAM-onDB-t1 benchmark of the IAM-OnDB, a database containing handwritten whiteboard notes [12], which consists of 56 different characters and provides writer-disjunct sets (one for training, two for validation, and one for testing). For our experiments, the same HMM topology as in [2] is used. While the experiments in our previous work [3, 5] are conducted using the Hidden-Markov-Toolkit (HTK, see [7]), the experiments presented in this paper are performed with the Graphical Models Toolkit (GMTK, see [8]), realizing statistical inference [13].

The following four experiments are conducted on the combination of both validation sets and with seven different codebook sizes ($N_\mathrm{cdb} \in \{10,100,500,1\,000,2\,000,5\,000,7\,500\}$). For training the vector quantizer, the parameters λ of the discrete HMMs, and the parameters of the GM the IAM-onDB-t1 training set is used. The results with respect to the actual codebook size N_cdb are depicted as *character accuracies* (ACC) in Fig. 3. The first three experiments are a repetition of the experiments presented in [3, 5], while the last experiment evaluates the GM depicted on the right-hand side of Fig. 2. However, these experiments are conducted using the GMTK instead of the HTK.

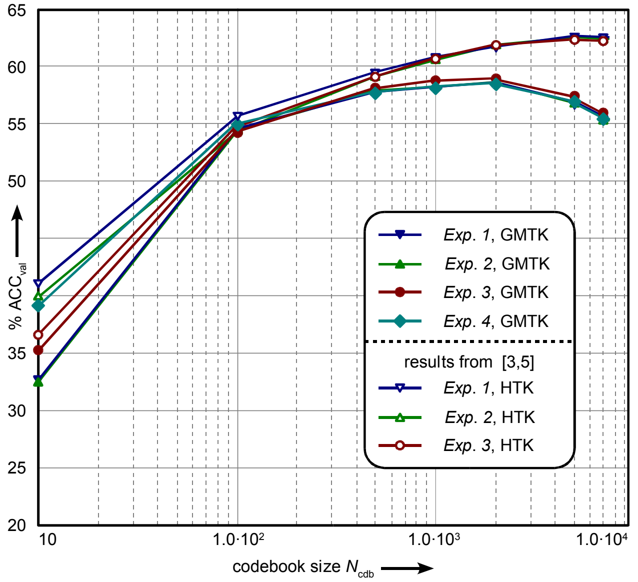

Figure 3: Evaluation of different systems' character accuracy with respect to the codebook size N_cdb estimated on the *validation* set.

Experiment 1 (*Exp. 1*): In the first experiment, all components of the feature vectors $(\tilde{f}_1, \ldots, \tilde{f}_{24})$ are quantized jointly by one codebook. The results shown in Fig. 3 form the baseline for the following experiments. The maximum character ACC of $a_\mathrm{b} = 58.7\,\%$ is achieved for a codebook size of $N_\mathrm{cdb} = 2\,000$. The drop in recognition performance when raising the codebook size to $N_\mathrm{cdb} = 5\,000$ and $N_\mathrm{cdb} = 7\,500$ is due to sparse data [1].

Experiment 2 (*Exp. 2*): To prove that the binary pressure feature $f_1/\tilde{f}_1$ is not adequately quantized by standard VQ, independent of the number of centroids, all features except the pressure information $(\tilde{f}_2, \ldots, \tilde{f}_{24})$ are quantized jointly for the second experiment. As Fig. 3 shows, only little degradation in recognition performance compared to the baseline can be observed. The peak rate of $a_\mathrm{r} = 58.7\,\%$ is again reached at a codebook size of $N_\mathrm{cdb} = 2\,000$. This observation confirms the findings presented in [3, 5]: the pressure information looses its significance when the feature vector is quantized.

Experiment 3 (*Exp. 3*): In order to model the pressure information without loss, it is represented in an own observation stream as shown in the GM on the left-hand side of Fig. 2

is used in this experiment. The result is an improvement of $\Delta r = 0.5\,\%$, and a peak character ACC of $a_{\text{m1}} = 59.0\,\%$ is achieved. This confirms the findings of our previous work.

Experiment 4 (*Exp. 4*): While in the previous experiment the pressure information is modeled statistically independently of the remaining features, in this experiment the GM as shown on the left-hand side of Fig. 2 is evaluated. The statistical bindings between the pressure information and the remaining features are found by statistical inference during training and hence, the multistream HMM is enhanced. In this case, a peak character ACC of $a_{\text{m2}} = 58.6\,\%$ is the result, which translates to a relative drop of $\Delta r = -0.2\,\%$.

When comparing the results of this paper with the result presented in [3, 5], one key result is that the HTK implementations of the recognition systems outperform the systems implemented with GMTK. The use of GMTK for the baseline system translates to a relative drop of $\Delta r = -5.1\,\%$ (from $a_{\text{HTK}} = 61.7\,\%$ to $a_{\text{GMTK}} = 58.7\,\%$) with a codebook consisting of, in this case, $N_{\text{cdb}} = 2\,000$ centroids.

Another observation is that when modeling the pressure information without any loss but disregarding the statistical dependencies between the pressure information and the remaining features the result is a slight improvement in recognition performance (see *Exp. 3*). However, when learning the probabilistic dependencies between the pressure information and the remaining features by training (see *Exp. 4*), the overall recognition performance drops. This is a contradiction to our earlier results: as shown in [3], modeling the pressure information without loss and respecting the statistical dependencies between the pressure information and the remaining features leads to an improvement of recognition performance. In [3] we used a modified vector quantizer to model the statistical dependencies, whereas here the statistical dependencies are learned from the training set. The enhancement of the HMM as shown in Fig. 2 leads to a higher number of variable parameters. As the number of training samples stays the same throughout all the presented experiments, the higher number of variables cannot be trained adequately. The number of trainable parameters also rises with the number of codebooks. For small codebooks (e. g. $N_{\text{cdb}} = 10$), the number of training samples is sufficient and hence, all parameters can be adequately learned. As a result, a higher character ACC is reached for the enhanced GM (see Fig. 3).

5. Outlook and Conclusion

In this paper, we investigated the use of discrete HMMs and discrete GMs for on-line HWR of whiteboard notes. Thereby a series of experiments has been conducted using the GMTK rather than the HTK for implementing the recognition systems. On the one hand, the optimized implementation found in HTK leads to a superior performance compared to the recognition systems implemented with GMTK. On the other hand, the GMTK is more flexible, allowing to design models more sophisticated than the HMMs. Motivated by earlier work (see [3]), in this paper, we presented a GM in order to learn the statistical dependencies between the pen's pressure information and the remaining features. It turns out that while for small codebooks, i. e. a small number of variable parameters, the new model delivers a better recognition performance. However, when the number of codebook entries is further raised, the number of variable parameters also rises. These parameters cannot be trained adequately given the training data. Hence, recognition performance drops.

In future work, the role of the statistical dependencies between the features used in on-line HWR of whiteboard notes is further investigated. Where necessary, optimized implementations are used for building improved recognition systems.

References

[1] L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257 – 285, 1989.

[2] M. Liwicki and H. Bunke, "HMM-Based On-Line Recognition of Handwritten Whiteboard Notes," *Proc. Int. Workshop on Frontiers in Handwriting Recognition*, pp. 595 – 599, 2006.

[3] J. Schenk, S. Schwärzler, G. Ruske, and G. Rigoll, "Novel VQ Designs for Discrete HMM On-Line Handwritten Whiteboard Note Recognition," *Proc. 30th Symposium of DAGM*, pp. 234 – 243, 2008.

[4] M. Liwicki and H. Bunke, "Feature Selection for On-Line Handwriting Recognition of Whiteboard Notes," *Proc. Conf. of the Graphonomics Society*, pp. 101 – 105, 2007.

[5] J. Schenk, S. Schwärzler, and G. Rigoll, "Discrete Single Vs. Multiple Stream HMMs: A Comparative Evaluation of Their Use in On-Line Handwriting Recognition of Whiteboard Notes," *Proc. Int. Conf. on Frontiers in Handwriting Recognition*, pp. 550 – 555, 2008.

[6] J. Bilmes, "Graphical Models and Automatic Speech Recognition," in *Mathematical Foundations of Speech and Language Processing*, M. Johnson, S.P. Khudanpur, M. Ostendorf, and R. Rosenfeld, Eds., pp. 191 – 246. Springer, 2004.

[7] S. Young, G. Evermann, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book (for HTK Version 3.2.1)*, Cambridge University Engineering Department, 2002.

[8] J. Bilmes and G. Zweig, "The Graphical Model Toolkit: An Open Source Software System for Speech and Time-Series Processing," *IEEE Int. Conf. Acoustics, Speech and Signal Processing*, pp. 3916 – 3919, 2002.

[9] J. Schenk, J. Lenz, and G. Rigoll, "Novel Script Line Identification Method for Script Normalization and Feature Extraction in On-Line Handwritten Whiteboard Note Recognition," *Pattern Recognition Journal*, p. in press, 2009.

[10] R.M. Gray, "Vector Quantization," *IEEE ASSP Magazine*, pp. 4 – 29, 1984.

[11] L.E. Baum and T. Petrie, "Statistical Inference for Probabilistic Functions of Finite State Markov Chains," *The Annals of Mathematical Statistics*, vol. 37, pp. 1554 – 1563, 1966.

[12] M. Liwicki and H. Bunke, "IAM-OnDB - an On-Line English Sentence Database Acquired from Handwritten Text on a Whiteboard," *Proc. Int. Conf. on Document Analysis and Recognition*, vol. 2, pp. 1159 – 1162, 2005.

[13] K. Murphy, *Dynamic Bayesian Networks: Representation, Inference and Learning*, Dissertation. University of California, Berkeley, 2002.