THE GRAPHICAL MODELS TOOLKIT: AN OPEN SOURCE SOFTWARE SYSTEM FOR SPEECH AND TIME-SERIES PROCESSING

Jeff Bilmes and Geoffrey Zweig

Department of Electrical Engineering, University of Washington, Seattle WA 98195
IBM T. J. Watson Research Center, Yorktown Heights, NY, 10598
bilmes@ee.washington.edu,gzweig@us.ibm.com

ABSTRACT

This paper describes the Graphical Models Toolkit (GMTK), an open source, publically available toolkit for developing graphical-model based speech recognition and general time series systems. Graphical models are a flexible, concise, and expressive probabilistic modeling framework with which one may rapidly specify a vast collection of statistical models. This paper begins with a brief description of the representational and computational aspects of the framework. Following that is a detailed description of GMTK's features, including a language for specifying structures and probability distributions, logarithmic space exact training and decoding procedures, the concept of switching parents, and a generalized EM training method which allows arbitrary sub-Gaussian parameter tying. Taken together, these features endow GMTK with a degree of expressiveness and functionality that significantly complements other publically available packages. GMTK was recently used in the 2001 Johns Hopkins Summer Workshop, and experimental results are described in detail both herein and in a companion paper.

1. INTRODUCTION

Although the statistical approach to pattern classification has been an integral part of automatic speech recognition (ASR) for over 30 years, the general paradigm is in no way exhausted. Today, new and promising statistical models are proposed for ASR every year. Sometimes one can simulate these new models using hidden Markov model (HMM) toolkits, but in such cases the new models cannot stray far from the basic HMM methodology. More often, a new model requires significant modifications on top of existing and already complex software. This is inefficient because a large amount of human effort must be placed into building new systems without having any guarantees about their performance. Therefore it is important to develop an over-arching and unifying statistical framework within which novel ASR methods can be accurately, succinctly, and rapidly employed.

Graphical models (GMs) are such a flexible statistical framework. With GMs, one uses a graph to describe a statistical process, and thereby defines one of its most important attributes, namely conditional independence. Because GMs describe these properties visually, it is possible to rapidly specify a variety of models without much effort. Interestingly, GMs subsume much of the statistical underpinnings of existing ASR techniques — no other known statistical abstraction appears to have this property. For example, it has been shown that the standard HMM Baum-Welch algorithm is only a special case of GM inference [16]. More importantly, the space of statistical algorithms representable with a GM is enormous; much larger than what has so far been explored for ASR. The time therefore seems ripe to start seriously examining such models.

Of course, this task is not possible without a (preferably freely-available and open-source) toolkit with which one may maneuver through the model space easily and efficiently. This paper describes the first version of GMTK, an open source, publically available toolkit for developing graphical-model based speech recognition systems. GMTK is meant to complement rather than replace other publically available packages — it has unique features, ones that are different from both standard ASR-HMM [17, 2, 3] and standard Bayesian network [1, 4] packages.

This paper provides an overview of GMTK, its notation, algorithms, main features, and reports baseline GMTK results. Research-related results are described in detail in a companion paper [19] which describes how GMTK was used in a recent Johns Hopkins University summer workshop. Section 2 describes the main representational ability of graphical models including the meanings of graphs, factorization of joint probability distributions, conditional independence properties, and parameterizations. The section also outlines the many computational benefits offered by GMs. Section 3 describes the main features of GMTK. Section 4 provides baseline results on the Aurora noisy speech corpus in the case when GMTK simulates an HMM system. Additional results, obtained at the recent JHU workshop are included in the companion paper [19]. Finally, Section 5 discusses future plans for the toolkit.

2. REPRESENTATION AND COMPUTATION

Two primary benefits offered by graphical models include representational ability and efficient algorithms for fast inference. This section briefly outlines them both.

2.1. Representation

A graphical model is a graph that represents certain properties about sets of random variables. The nodes in the graph correspond to random variables, and the edges encode a set of conditional independence properties. These properties may be used to obtain a number of valid factorizations of the joint probability distribution. There are many different types of graphical models, such as Bayesian networks (a type of directed graphical model), Markov random fields (undirected models), causal models, chain graphs, and so on. Each type has its own formal semantics [14] for specifying conditional independence relations. Only along with its agreed upon semantics does a GM precisely specify conditional independence properties. Variables in a GM may either be observed (their values are known), or hidden. The name hidden Markov model, for example, results from there being a Markov chain consisting only of hidden variables.

The first version of GMTK uses the semantics of Bayesian networks (BNs) [15, 13]. This means that the graphs are directed, and conditional independence properties are determined by the notion of "d-separation" [15]. Using d-separation one may read off conditional independence statements from the graph which hold for all distributions represented by the graph. Further, the joint probability distribution may be factored as the product of the probability of each variable's value, given the values of its parents in the graph.

Speech is a time signal, and any GM intending to model speech must somehow take this into account. Accordingly, dynamic Bayesian networks (DBNs) [11] are Bayesian networks which include directed edges pointing in the direction of time. Other than the existence of time-edges, DBNs have the same semantics as other BNs.

The structure of a graphical model represents the way in which a set of random variables probabilistically represents natural and artificial processes (such as the existence of articulatory or noise-type variables), and how these variables interact (such as an n^{th} order Markov chain, tree-based dependencies, and so on). The structure also represents constraints on variable values and sequences of values (such as valid phone-sequences in a speech-recognizer). These representational aspects of graphical modeling are detailed in [18, 6].

Lastly, ASR is inherently a problem of pattern classification, and requires statistical models to discriminate between different speech utterances. Apart from discriminatively learned model parameters (such as means, variances, or transition matrices), graphical models are ideally suited for experimenting with discriminative structures [8, 19].

2.2. Computation

Probabilistic inference, such as evaluating (or computing the most likely value of) a conditional distribution, is the foundation behind all statistical computing. Graphical models have an associated set of algorithms which perform inference as efficiently as possible. Apart from describing the structure of a domain, conditional independence can lead to enormous computational savings when doing inference. For example, computing the quantity $p(y|x)$ could be done the hard way $p(y|x) = \sum_{a,b} p(y,a,b|x)$ or the easy way $p(y|x) = \sum_a p(y|a) \sum_b (a|b)p(b|x)$, the latter case assuming independence relations making the computation probabilistically valid. Graphical models use inference algorithms (e.g., the junction-tree algorithm [15, 13] or the generalized distributed law [5]) that provably correspond to valid calculations on probabilistic equations. These algorithms essentially distribute summations to the right into products as efficiently as possible, as above. There are many ways of doing this, one of which is used in GMTK (and described in Section 3.3). Other approaches forfeit exact inference for the sake of speed, and resort instead to approximate methods. In any case, when efficient and accurate probabilistic inference is required, GMs provide numerous possibilities.

3. TOOLKIT FEATURES

GMTK has a number of features that support a wide array of statistical models suitable for speech recognition and other time-series data. GMTK may be used to produce a complete ASR system for both small- and large-vocabulary domains. The graphs themselves may represent everything from N-gram language models down to Gaussian components, and the probabilistic inference mechanism supports first-pass decoding in these cases.

3.1. Explicit vs. Implicit Modeling

In general, there are two representational extremes one may employ when using GMTK for an ASR system. On the one hand, a graph may explicitly represent all the underlying variables and control mechanisms (such as sequencing) that are required in an ASR system [18]. We call this approach an "explicit representation" where variables can exist for such purposes as word identification, numerical word position, phone or phoneme identity, the occurrence of a phoneme transition, and so on. In this case, the structure of the graph explicitly represents the interesting hidden structure underlying an ASR system. On the other hand, one can instead place most or all of this control information into a single hidden Markov chain, and use a single integer state to encode all contextual information and control the allowable sequencing. We call this approach an "implicit" representation.

As an example of these two extremes, consider the word "yamaha" with pronunciation /y aa m aa hh aa/. The phoneme /aa/ occurs three times, each in different contexts, first preceding an /m/, then preceding an /hh/, and finally preceding a word boundary. In an ASR system, it must somewhere be specified that the same phoneme /aa/ may be followed only by one of /m/, /h/, or a word boundary depending on the context — /aa/, for example, may not be followed by a word boundary if it is the first /aa/ of the word. In the explicit GM approach, the graph and associated conditional probabilities unambiguously represent these constraints. In an implicit approach, all of the contextual information is encoded into an expanded single-variable hidden state space, where multiple HMM states correspond to the same phoneme /aa/ but in different contexts.

The explicit approach is useful when modeling the detailed and intricate structures of ASR. It is our belief, moreover, that such an approach will yield improved results when combined with a discriminative structure [6, 8, 19], because it directly exposes events such as word-endings and phone-transitions for use as switching parents (see Section 3.4). The implicit approach is further useful in tempering computational and/or memory requirements. In any case, GMTK supports both extremes and everything in between — a user of GMTK is therefore free to experiment with quite a diverse and intricate set of graphs. It is the task of the toolkit to derive an efficient inference procedure for each such system.

3.2. The GMTKL Specification Language

A standard DBN [11] is typically specified by listing a collection of variables along with a set of intra- and inter-dependencies which are used to unroll the network over time. GMTK generalizes this ability via dynamic GM templates. The template defines a collection of (speech) frames and a chunk specifier. Each frame declares an arbitrary set of random variables and includes attributes such as parents, type (discrete, continuous), parameters to use (e.g. discrete probability tables or Gaussian mixtures) and parameter sharing. At the end of a template is a chunk specifier (two integers, $N : M$) which divides the template into a prologue (the first $N-1$ frames), a repeating chunk, and an epilogue (the last $T-M$ frames, where T is the frame-length of the template). The middle chunk of frames is "unrolled" until the dynamic network is long enough for a specific utterance.

GMTK uses a simple textual language (GMTKL) to define GM templates. Figure 1 shows the template of a basic HMM in GMTKL. It consists of two frames each with a hidden and an observed variable, and dependences between successive hidden and between observed and hidden variables.

A template chunk may consist of several frames, where each frame contains a different set of variables. Using this feature, one can easily specify multi-rate GM networks where variables occur over time at rates which are fractionally but otherwise arbitrarily related to each other.

3.3. Inference

GMTK supports a number of operations for computing with arbitrary graph structures, the four main ones being:

1. Integrating over hidden variables to compute the observation probability: $P(\mathbf{o}) = \sum_{\mathbf{h}} P(\mathbf{o}, \mathbf{h})$
2. Finding the likeliest hidden variable values: $argmax_{\mathbf{h}} P(\mathbf{o}, \mathbf{h})$
3. Sampling from the joint distribution $P(\mathbf{o}, \mathbf{h})$

```
frame: 0 {
   variable : state {
      type : discrete hidden cardinality 4000;
      switchingparents : nil;
      conditionalparents : nil using MDCPT("pi");
   }
   variable : observation {
      type : continuous observed 0:38;
      switchingparents : nil;
      conditionalparents : state(0)
         using mixGaussian mapping("state2obs");
   }
}

frame: 1 {
   variable : state {
      type : discrete hidden cardinality 4000;
      switchingparents : nil;
      conditionalparents : state(-1)
         using MDCPT("transitions");
   }
   variable : observation {
      type : continuous observed 0:38;
      switchingparents : nil;
      conditionalparents : state(0)
         using mixGaussian mapping("state2obs");
   }
}

chunk 1:1;
```

Fig. 1. GMTKL specification of an HMM structure. The feature vector in this case is 39 dimensional, and there are 4000 hidden states. Frame 1 can be duplicated or "unrolled" to create an arbitrarily long network.

4. Parameter estimation given training data $\{\mathbf{o}_k\}$ via EM/GEM: $argmax_\theta \prod_k P(\mathbf{o}_k|\theta)$

A critical advantage of the graphical modeling framework derives from the fact that these algorithms work with *any* graph structure, and a wide variety of conditional probability representations. GMTK uses the *Frontier Algorithm*, detailed in [18, 21], which converts arbitrary graphs into equivalent chain-structured ones, and then executes a forwards-backwards recursion. The chain structure is particularly advantageous because it supports beam-pruning in a very natural way, allows deterministic relationships between variables to be immediately identified and exploited, and, as we see in the next section, allows for exact inference in logarithmic space.

3.3.1. Logarithmic Space Computation

In many speech applications, observation sequences can be thousands of frames long. When there are a dozen or so variables per frame (as in an articulatory network), the resulting unrolled network might have tens of thousands of nodes, and cliques may have millions of possible values. A naive implementation of exact inference, which stores all clique values for all time, would result in (an obviously prohibitive) gigabytes of required storage

To avoid this problem, GMTK implements a recently developed procedure [10, 20] that reduces memory requirements exponentially from $O(T)$ to $O(\log T)$. This reduction has a truly dramatic effect on memory usage, and can additionally be combined with GMTK's beam-pruning procedure for further memory savings. The key to this method is recursive divide-and-conquer. With k-way splits, the total memory usage is $O(k \log_k T)$, and the runtime is $O(T \log_k T)$. The constant of proportionality is related to the number of entries in each clique, and becomes smaller with pruning. For algorithmic details, the reader is referred to [20].

3.3.2. Generalized EM

GMTK supports both EM and generalized EM (GEM) training, and automatically determines which to use based on the parameter sharing currently in use. GEM training is distinctive because it provides a provably convergent method for parameter estimation, even when there is an arbitrary degree of tying, even down to the level of Gaussian means, covariances, or factored covariance matrices (see Section 3.6).

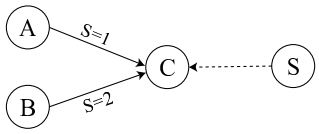

Fig. 2. When $S = 1$, A is B's parent, when $S = 2$, B is C's parent. S is called a switching parent, and A and B conditional parents.

3.3.3. Sampling

Drawing variable assignments according to the joint probability distribution is useful in a variety of areas ranging from approximate inference to speech synthesis, and GMTK supports sampling from arbitrary structures. The sampling procedure is computationally inexpensive, and can thus be run many times to get a good distribution over hidden (discrete or continuous) variable values.

3.4. Switching Parents

GMTK supports another novel feature rarely found in GM toolkits, namely switching parent functionality (also called Bayesian multinets [8]). Normally, a variable has only one set of parents. GMTK, however, allows a variable's parents to change (or switch) conditioned on the current values of other parents. The parents that may change are called conditional parents, and the parents which control the switching are called switching parents. Figure 2 shows the case where variable S switches the parents of C between A and B, corresponding to the probability distribution: $P(C|A,B) = P(C|A,S=1)P(S=1) + P(C|B,S=2)P(S=2)$. This can significantly reduce the number of parameters required to represent a probability distribution, for example, $P(C|A,S=1)$ needs only a 2-dimensional table whereas $P(C|A,B)$ requires a three dimensional table. Switching functionality has found particular utility in representing certain language models, as experiments during the JHU2001 workshop demonstrated.

3.5. Discrete Conditional Probability Distributions

GMTK allows the dependency between discrete variables to be specified in one of three ways. First, they may be deterministically related using flexible n-ary decision trees. This provides a sparse and memory-efficient representation of such dependencies. Alternatively, fully random relationships may be specified using dense conditional probability tables (CPTs). In this case, if a variable of cardinality N has M parents of the same cardinality, the table has size N^{M+1}. Since this can get large, GMTK supports a third sparse method to specify random dependencies. This method combines sparse decision trees with sparse CPTs so that zeros in a CPT simply do not exist. The method also allows flexible tying of discrete distributions from different portions of a CPT.

3.6. Graphical Continuous Conditional Distributions

GMTK supports a variety of continuous observation densities for use as acoustic models. Continuous observation variables for each frame are declared as vectors in GMTKL, and each observation vector variable can have an arbitrary number of conditional and switching parents. The current values of the parents jointly determine the distribution used for the observation vector. The mapping

	clean	20	15	10	5	0	-5
GMTK-WWM	99.2	98.5	97.8	96.0	89.2	66.4	21.5
GMTK-PH	99.1	98.3	97.2	94.9	86.4	54.9	2.80
HP	98.5	97.3	96.2	93.6	85.0	57.6	24.0

Table 1. Word recognition rates: baseline GMTK emulating an HMM system as function of SNR. HP is from [12].

from parent values to child distribution is specified using a decision tree, allowing a sparse representation of this mapping. A vector observation variable spans over a region of the feature vector at the current time. GMTK thereby supports multi-stream speech recognition, where each stream may have its own set of observation distributions and sets of discrete parents.

The observation distributions themselves are mixture models. GMTK uses a splitting and vanishing algorithm during training to learn the number of mixture components. Two thresholds are defined, a mixture-coefficient vanishing ratio (mcvr), and a mixture-coefficient splitting ratio (mcsr). Under a K-component mixture, with component probabilities p_k, if $p_k < 1/(K \times \text{mcvr})$, then the k^{th} component will vanish. If $p_k > \text{mcsr}/K$, that component will split. GMTK also supports forced splitting (or vanishing) of the N most (or least) probable components at each training iteration. Sharing portions of a Gaussian such as means and covariances can be specified either by-hand via parameter files, or via a split (e.g., the split components may share an original covariance).

Each component of a mixture is a general conditional Gaussian. In particular, the c-component probability is $p(x|z_c, c) = \mathcal{N}(x|B_c z_c + f_c(z_c) + \mu_c, D_c)$ where x is the current observation vector, z_c is a c-conditioned vector of continuous observation variables from any observation stream and from the past, present, or future, B_c is an arbitrary sparse matrix, $f_c(z_c)$ is a multi-logistic non-linear regressor, μ_c is a constant mean residual, and D_c is a diagonal covariance matrix. Any of the above components may be tied across multiple distributions, and trained using the GEM algorithm.

GMTK treats Gaussians as directed graphical models, and can thereby represent all possible Gaussian factorization orderings, and all subsets of parents in any of these factorizations. Under this framework, GMTK supports diagonal, full, banded, and semi-tied factored sparse inverse covariance matrices [9]. GMTK can also represent arbitrary switching dependencies between individual elements of successive observation vectors. GMTK thus supports both linear and non-linear buried Markov models [7]. All in all, GMTK supports an extremely rich set of observation distributions.

4. EXPERIMENTAL VALIDATION

This section validates GMTK by producing a GMTK-based ASR system for the Aurora2.0 noisy digits task [12]. While many more results are reported in a companion paper [19], we wish to demonstrate here that GMTK can produce competitive performance on a standard ASR task. For all the results presented here, GMTK emulated an HMM using the explicit modeling approach, as mentioned in Section 3. Table 1 presents results for systems using both whole-word models (WWM), and shared-phone models (PH). The whole-word system was similar to that outlined in [12], except that it used 4 Gaussians per state rather that 3 for a total of 715 Gaussians. Our phone based system used 20 3-state phones, and a total of 710 Gaussians.

5. CONCLUSIONS

This paper introduces GMTK, a flexible open-source toolkit for working with graphical models on speech and other time series data. The toolkit was used at the recent JHU2001 workshop in concert with algorithms for producing discriminative graph structures. GMTK supports an enormous variety of statistical models, and these can be rapidly specified and integrated into a speech recognition system.

The features that are described in this paper correspond to version 1.0 of the toolkit. Our intention is to continue to improve both GMTK features and computational efficiency. Later versions will be released which incorporate and support many new features.

This work was partially supported by NSF Grant 0093430, NSF Grant IIS-0097467, and DARPA contracts N66001-99-2-8916 and N66001-99-2-892403. We thank Peng Xu and Karen Livescu for compiling the baseline results.

6. REFERENCES

[1] The BUGS project. http://www.mrc-bsu.cam.ac.uk/bugs/Welcome.html.

[2] CMU Sphinx: Open source speech recognition. http://www.speech.cs.cmu.edu/sphinx/Sphinx.html.

[3] The ISIP public domain speech to text system. http://www.isip.msstate.edu/projects/speech/software/index.html.

[4] The Matlab bayesian network toolbox. http://www.cs.berkeley.edu/~murphyk/Bayes/bnsoft.html.

[5] S. M. Aji and R. J. McEliece. The generalized distributive law. *IEEE Transactions in Information Thoery*, 46:325–343, March 2000.

[6] J. Bilmes. *Natural Statistical Models for Automatic Speech Recognition.* PhD thesis, U.C. Berkeley, Dept. of EECS, CS Division, 1999.

[7] J.A. Bilmes. Buried Markov models for speech recognition. In *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, Phoenix, AZ, March 1999.

[8] J.A. Bilmes. Dynamic Bayesian Multinets. In *Proceedings of the 16th conf. on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 2000.

[9] J.A. Bilmes. Factored sparse inverse covariance matrices. In *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, Istanbul, Turkey, 2000.

[10] J. Binder, K. Murphy, and S. Russell. Space-efficient inference in dynamic probabilistic networks. *Int'l, Joint Conf. on Artificial Intelligence*, 1997.

[11] T. Dean and K. Kanazawa. Probabilistic temporal reasoning. *AAAI*, pages 524–528, 1988.

[12] H. G. Hirsch and D. Pearce. The aurora experimental framework for the performance evaluations of speech recognition systems under noisy conditions. *ICSA ITRW ASR2000*, September 2000.

[13] F.V. Jensen. *An Introduction to Bayesian Networks.* Springer, 1996.

[14] S.L. Lauritzen. *Graphical Models.* Oxford Science Publications, 1996.

[15] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, 2nd printing edition, 1988.

[16] P. Smyth, D. Heckerman, and M.I. Jordan. Probabilistic independence networks for hidden Markov probability models. Technical Report A.I. Memo No. 1565, C.B.C.L. Memo No. 132, MIT AI Lab and CBCL, 1996.

[17] S. Young, J. Jansen, J. Odell, D. Ollason, and P. Woodland. *The HTK Book*. Entropic Labs and Cambridge University, 2.1 edition, 1990's.

[18] G. Zweig. *Speech Recognition with Dynamic Bayesian Networks.* PhD thesis, U.C. Berkeley, 1998.

[19] G. Zweig, J. Bilmes, T. Richardson, K. Filali, K. Livescu, P. Xu, K. Jackson, Y. Brandman, E. Sandness, E. Holtz, J. Torres, and B. Byrne. Structurally discriminative graphical models for automatis speech recognition -results from the 2001 johns hopkins summer workshop. *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, 2002.

[20] G. Zweig and M. Padmanabhan. Exact alpha-beta computation in logarithmic space with application to map word graph construction. *Int. Conf. on Spoken Lanugage Processing*, 2000.

[21] G. Zweig and S. Russell. Probabilistic modeling with bayesian networks for automatic speech recognition. *Australian Journal of Intelligent Information Processing*, 5(4):253–260, 1999.