Hidden Markov Models

Computational Foundations of Artificial Intelligence

Professor: Michael Degorgio

FAU
FLORIDA ATLANTIC UNIVERSITY

2020

1 Introduction

An hidden Markov model (HMM) is a mathematical model which describes the possible sequencings of discrete random variables, otherwise known as state variable sequences, under the assumption that a given sequence of observations is generated via some unobserved state sequence. To simplify the mathematics, the transitions between states are assumed to be a Markov process, that is to say they follow the Markov assumption. Before examining the latent dimension of these models, a reduction of this problem is worth exploring.

2 Markov Chains

A Markov model, or Markov chain, models the probabilities of particular state sequences valued from some finite set. Markov chains model a Markov process; they follow the assumption that when predicting the future only the present matters, not the past, also known as the Markov assumption or property. Formally, considering a sequence of state variables $q_1, q_2, ..., q_i$:

$$\textbf{Markov Assumption: } P(q_i = a|q_1...q_{i-1}) = P(q_i = a|q_i = q_{i-1}) \tag{1}$$

Similarities can be drawn between automaton, language models and Markov chains. As shown in Figure 1, Markov models are merely finite state automaton with stochastic transition behavior described by probabilities. With their dependence solely on the previous state, Markov chains would model a bigram language from the state vocabulary or alphabet $q_1, q_2, ..., q_i$ which could produce the probabilities for particular letter sequences representing a word or word sequences [JM09].

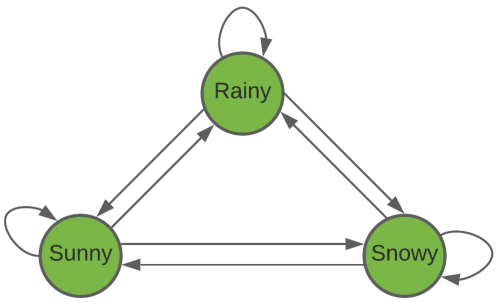

Figure 1: Markov Chain

Looking closer, Markov chains are composed of the following:

$Q = q_1, q_2, ..., q_N$	a set of *N* states
$A = a_{11}, a_{12}, ..., a_{n1}, ..., a_{nn}$	transition probability matrix A, each a_{ij} representing the probability of moving from state *i* to state *j*
$\Pi = \pi_1, \pi_2, ..., \pi_N$	initial probability distribution over the *N* states

Given these components one can find the probabilities for every sequence Q as a member of the set of all possible sequences Q^*.

3 Hidden Markov Models

Markov chains effectively demonstrate probabilities for sequences of observable events. What if those states aren't observed directly (hidden)? To use an example from language models: part-of-speech tags aren't what's observed in a document, the words are. The POS tags are inferred from the sequence of words [JM09]. To handle this an augmentation of the Markov chain is made: enter the hidden Markov model, which handles both *observed* and *hidden* events of importance to the model. By producing observations from underlying hidden states represented within the model, more complexity in problem structure can be modeled.

Formally, a hidden Markov model is composed as:

$Q = q_1, q_2, ..., q_N$	a set of *N* states
$A = a_{11}, a_{ij}, ..., a_{NN}$	transition probability matrix A, each a_{ij} representing the probability of moving from state *i* to state *j*
$O = o_1, o_2, ..., o_T$	A sequence of T observations
$B = b_i(o_t)$	a sequence of emission probabilities expressing the probability of an observation o_t being generated from a state i
$\Pi = \pi_1, \pi_2, ..., \pi_N$	initial probability distribution over the *N* states

Unsurprisingly, the Markov Assumption holds for hidden Markov models. Important is another assumption for hidden Markov models, output independence. The probability of an output observation o_i depends solely on the state producing the observation q_i:

Output Independence: $P(o_i|q_1, ...q_i, ..., q_T, o_1, ..., o_i, ..., o_T) = P(o_i|q_i)$ (2)

To illustrate Markov models, their state transitions, emission probabilities and their use of automata, consider a toy problem where a local amusement park makes available a ledger containing the average number of rides purchased per person for a waterslide on any given day. Some days were rainy while others were sunny, which likely influences the average number of purchased rides for a day. Let these two types of days be our set of states: $Q = \{sunny, rainy\}$. For the sake of this problem, only one, two, or three rides were purchased on average per person. This is our observation set (or alphabet): $O = \{1, 2, 3\}$, where each observation corresponds to the average number of rides per person for a given day which is solely explained by a single hidden state as dictated by assumption 3, output independence.

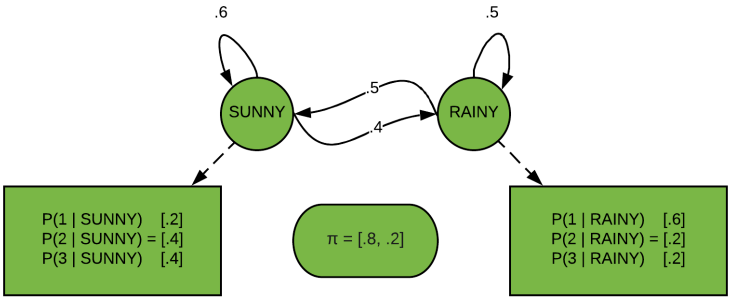

Figure 2: Example HMM

Figure 2 shows an example HMM for the waterslide ride task where the two rectangles correspond to their respective state's emission probabilities A and each edge corresponds to a transition probability as part of the transition matrix B. It can be seen how traversing this automaton $T - 1$ times would produce two T-tuples of states and observations, where the states are unknown to the observer and determining said hidden state sequence is desirable such that the observed process (average number of waterslide rides) can be explained.

Before delving further into examples of hidden Markov models, an important algorithm must be introduced. As the length of sequences O and Q are equivalent under assumptions 2 and 3, the hidden Markov model one-to-one maps the sequence of hidden states and the sequence of observations to each other [JM09]. This mapping allows for a convenient property of hidden Markov models. The probability for any hidden state sequence $Q = q_0, q_1, ..., q_T$ and an observation sequence $O = o_0, o_1, ..., o_T$ with *known* hidden states can be calculated as follows:

$$P(O|Q) = \prod_{i=1}^{T} P(o_i|q_i) \tag{3}$$

While succinct and interpretable, this conditional probability assumes known hidden state, which of course is not available in this class of problems. By weighting each conditional probability as calculated above with the probability that that particular state was *transitioned to* $P(q_i|q_{i-1})$, learned is the joint probability of being in a particular state sequence $Q = q_0, q_1, ..., q_T$ generating a particular observation sequence $O = o_0, o_1, ..., o_T$:

$$P(O, Q) = P(O|Q) * P(Q) = \prod_{i=1}^{T} P(o_i|q_i) * \prod_{i=1}^{T} P(q_i|q_{i-1}) \tag{4}$$

Unfortunately equation 5 only provides the probability of an observation sequence O given only a single hidden sequence Q. Considering that there are N possible hidden states and each hidden sequence is of length T, there are a total of N^T possible hidden sequences that must be considered before producing the actual observation sequence's O likelihood, as every possible hidden state sequence *must* be considered to produce an observation sequence's O likelihood. This exponential time complexity is highly undesirable for finding the probability of an observation sequence O. Fortunately finding this likelihood is tractable with the efficient solution called the forward algorithm, although the algorithm does not produce a probability distribution as the exponential approach would, it merely efficiently calculates the likelihood of an observation sequence O. Calculating observation sequence O's likelihood is central to solving most problems within hidden Markov models.

4 HMM Canonical Problems

According to Rabiner, hidden Markov models are distinguished by three primary problems [Rab89]. Solving these three problems motivates the use of the hidden Markov model:

1. Likelihood	Given A & B (written as $\lambda = (A, B)$) and a sequence of observations O, find the likelihood $P(O\|\lambda)$.
2. Decoding	Given $\lambda = (A, B)$ and a sequence of observations O, find the state sequence which best explains O.
3. Learning	Given a sequence of observations O and the set of states in the HMM Q, learn A and B.

The inefficiency problem introducing the forward algorithm above was problem 1, calculating the likelihood. The forward algorithm, which is not explored further in this paper, produces the likeli-

hood of a given observation sequence O given the HMM parameters $\lambda = (A, B)$ which can be written as $P(O|\lambda)$.

For problem 2 decoding, the distribution of hidden state sequences Q for an observation sequence O could be used to to determine the state sequence which best explains O, however this is computationally inefficient. If the exponential time complexity calculation of each hidden state sequence's probability as described in problem 1 were somehow made instead of using the forward algorithm, a state sequence probability distribution would be produced. This would obsolete the need for decoding in problem 2 as the most probable state sequence given by $\lambda = (A, B)$ and a sequence of observations O could be chosen from the most probable hidden state sequence Q of the distribution. Of course, the algorithms exponential time complexity is computationally inefficient and therefore not used. If instead the forward algorithm were used, which is efficient in time complexity $O(N^2T)$ for producing a likelihood for a state sequence Q given an observation sequence O, one would still have to consider N^T hidden state sequences to produce the likelihood for Q across all state sequences, which would still be inefficient.

Enter the Viterbi algorithm, which is used to efficiently calculate the most probable state sequences across all possible observation sequences. Algorithmically, the Viterbi algorithm and the forward algorithm differ in only one aggregate function: max versus sum when considering previous path probabilities. It can be seen that these functions would properly associate with their use: max for determining the most likely path through the automaton and sum for determining the actual probability of the path through the automaton [JM09]. On top of this the Viterbi algorithm not only calculates a probability value but must also produce a most likely sequence. The algorithm incorporates backpointers so a backtrace may be performed after running the algorithm such that the most like sequence may be retrieved.

This leaves problem 3, which could be considered the most important. Empirically the emission and transition probabilities are highly obfuscated from the observed process, and must be learned in some way. The parameters of an HMM are learned through the tandem work of the forward and the backward algorithm, known as the forward-backward algorithm or as the Baum-Welch algorithm. The forward-backward algorithm is an expectation-maximization algorithm iteratively estimating the probability for hidden states (using the forward and backward algorithms) and subsequently the constituent model parameters $\lambda = (A, B)$ as used by the forward and backwards algorithms. While the forward algorithm has been motivated by the canonical problems of determining a state sequence's likelihood (problem 1) and decoding the most probable hidden state sequence from a given observation sequence

through the Viterbi variant (problem 2), the backwards algorithm's utility might be less clear. The backward algorithm calculates the probability of encountering observations from time $t+1$ to the end, given an automaton λ and a state i at time t:

$$\beta_t(i) = P(o_{t+1}, o_{t+2}, ..., o_T | q_t = i, \lambda) \tag{5}$$

It is comprehensible, then, that both the forwards and backwards algorithm are essential in producing likelihoods for the purpose of learning an hidden Markov model's parameters $\lambda = (A, B)$.

5 Hidden Markov Model Variants & Applications

Hidden Markov models are used in numerous real-world applications in various adaptations. This section details these fields and their variant uses within them.

Variants

Higher-Order HMM

The higher order hidden Markov model (HO-HMM) is a generalization of the standard (first-order) hidden Markov model relaxing assumption 3, output independence. The dependency on the previous state is extended to n states [LJ16]. As such, transition probabilities and observation probabilities are also dependent on n states. The second-order hidden Markov model (SOHMM) is an HOMM where $n = 2$.

Hidden Semi-Markov Model

Hidden semi-Markov models (HSMM) provide a method to handle the duration of state explicitly. The Markov assumption on unobservable processes is relaxed to be semi-Markov such that the probability of a change in hidden state is contingent not only on current state but time elapsed since reaching that state [Mor20]. More formally, a hidden state q_t will remain in that state for some time duration d whilst emitting d observations. This is in contrast to the first-order hidden Markov model's constant probability of state change regardless of time duration d.

Factorial HMM

The factorial HMM (FHMM) generalizes the state structure into a multi-layer state configuration, relaxing the multimodal assumption over state variables into a distributed state representation, first explored by Williams and Hinton [WH91]. Instead of a single Markov chain for state variables, the FHMM

introduces a collection of M Markov chains independent of each other. Each Markov chain typically contains its own transition matrix A and emission matrix B. Thus, for any time step t, the current state is represented by M state variables $q_t = \{q_t^1, ..., q_t^M\}$. Interestingly the implementations of a forward-backward algorithm are computationally inefficient such that learning is done through approximative Markov chain Monte Carlo sampling processes like Gibbs sampling or through variational Bayesian methods [GJ97].

Layered HMM

A layered HMM (LHMM) is several HMMs composed together such that they run parallel together. For each of the L layers, K HMMs exist, producing a likelihood for each class k. Shown in Figure 3, each layer's observations originate from computations performed by the previous layer, such that the produced sequence of observations at layer L can be classified into one of K_L classes. The one exception is the lowest layer which produces k likelihoods from the observed process.

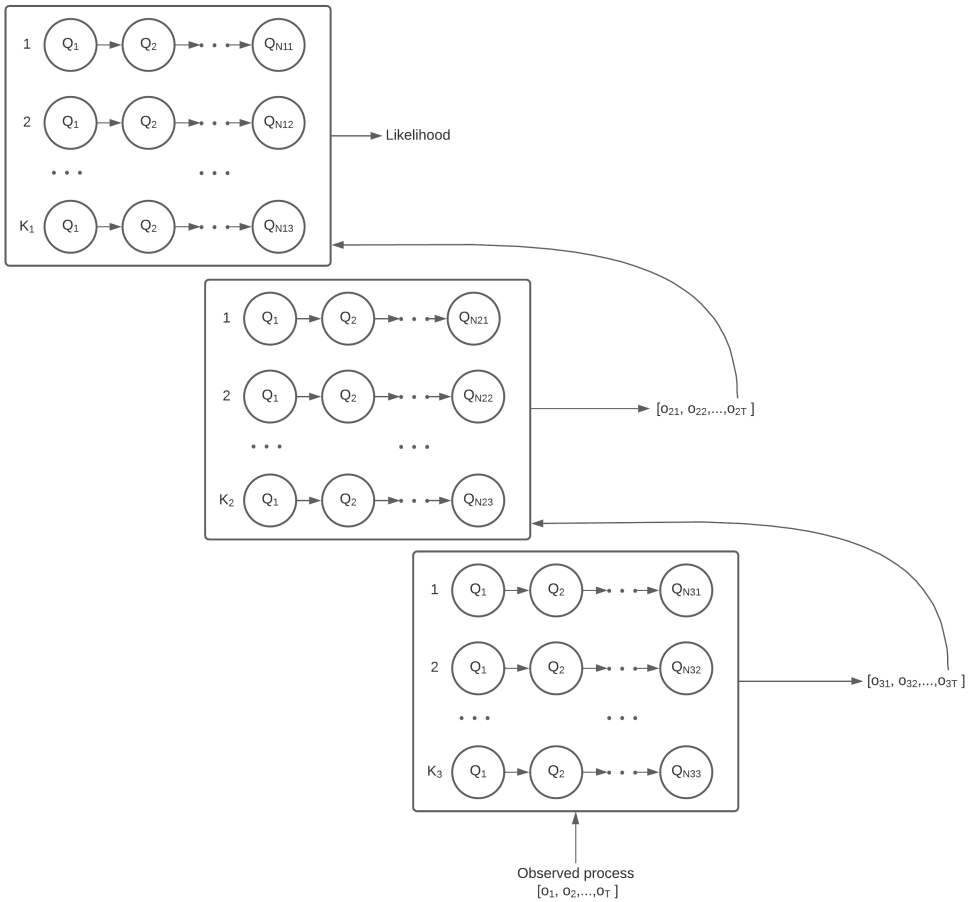

Figure 3: Structure of LHMM

Autoregressive HMM

Autoregressive HMMs (AR-HMM) model temporal structures in time series data through combining autoregressive time series modelling with HMM architecture. By introducing direct stochastic dependence between observations the ARHMM can model long-term correlations of sequential data. In some implementations only the dependence between the consecutive observations is considered while others consider larger observation chains [SWF14].

Non-Stationary HMM

The non-stationary HMM (NS-HMM) is a generalization of the hidden semi-Markov model, constructed to address the HMM's poor ability in modeling state duration behavior. This model variant introduces dynamic state transition probabilities as a function of time duration t to the HMM architecture, similar to HSMM, while also producing state *duration* probabilities as a function of time [SK95].

Hierarchical HMM

The hierarchical HMM (HHMM) introduces multi-level states in a tree-like structure such that sequences can be described at various degrees of granularity. Within the HHMM (shown in Figure 4) each state is modeled by sub-HHMMs which generate internal states. Thus each state of the HHMM does not produce a single observation but a sequence of observations, where the last state in the Markov chain represents terminal nodes, determining the vertical transition behavior for the sub-tree [FST98].

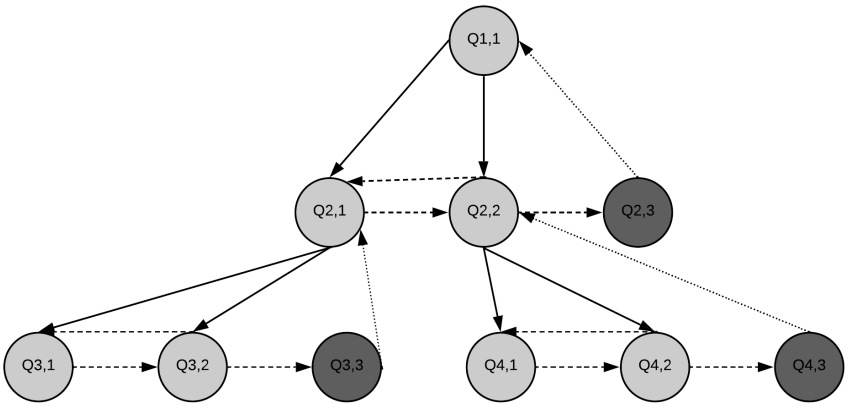

Figure 4: Structure of HHMM

Applications

With the myriad of HMM variants established in analyzing or generating sequential data over the decades, HMM applications are even more diverse across their fields, as shown in Figure 5. By examining papers across numerous fields, from signal processing to tool condition monitoring, a high-level view of HMM variants and their applications is made.

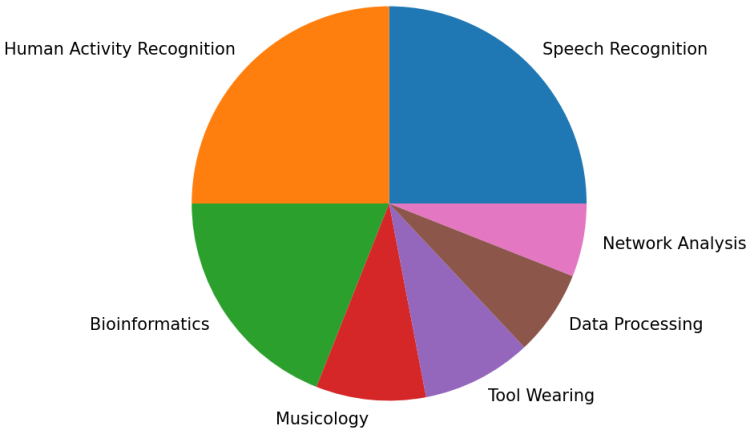

Figure 5: HMM Applications by Field

Figure 6: Model Variant Usage

Speech Recognition

Modern general-purpose speech recognition applications make use of hidden Markov models for their simplicity and success. Their computational efficiency and automatic learning through online-modified EM algorithms make for fast, constantly learning processing of speech signal. Yang et al. demonstrated lexical tone recognition for Mandarin speech using both vector quantization and a first-order hidden Markov model. The observation sequence was produced via vector quantization upon the logarithmic pitch interval and its first derivative across 72 monosyllabic utterances, demonstrating success in classifying tones into four categories [Yan+88].

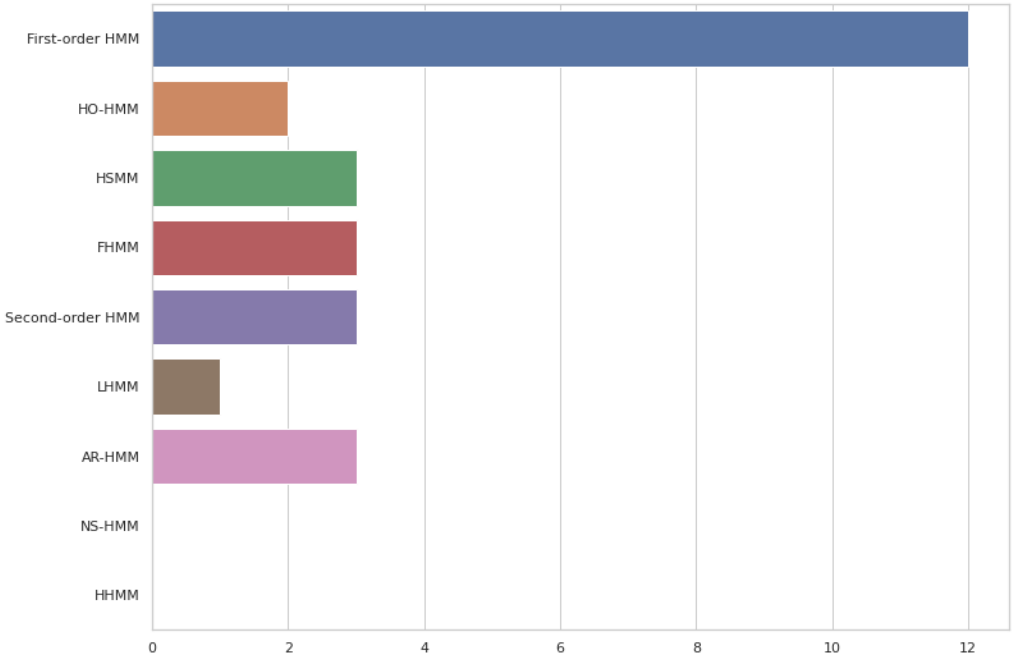

Figure 7: Model Applications in Speech Recognition

Shown in Figure 7 is the distribution of model variants across a number of publications in the field of speech recognition. Of note is the high proportion of first order hidden Markov models within the distribution. While other models are used, typically the most simple Markov model as described in Section 3 is chosen. This could be attributed to the non-stationary nature of speech signal.

Tool Condition Monitoring

Tool wear in machining processes is highly undesirable as it negatively impacts tool life. Tool life is important due to its correlation with quality and accuracy of machined surfaces [EDN06]. A natural extension of modeling such behavior to mitigate wear is the economic value its performance offers manufacturers. Also of importance in machining is early fault detection. Li et al. demonstrated early

fault detection in helicopter gearboxes via hidden semi-Markov models [Li+17]. Tool wear is analyzed through many mediums including visual, auditory and even vibrational signal. As shown in Figure 8, tool condition monitoring is the field with the highest proportion of hidden semi-Markov model usage across papers analyzed. This correlates with the high proportion of hidden semi-Markov models shown in Figure 6.

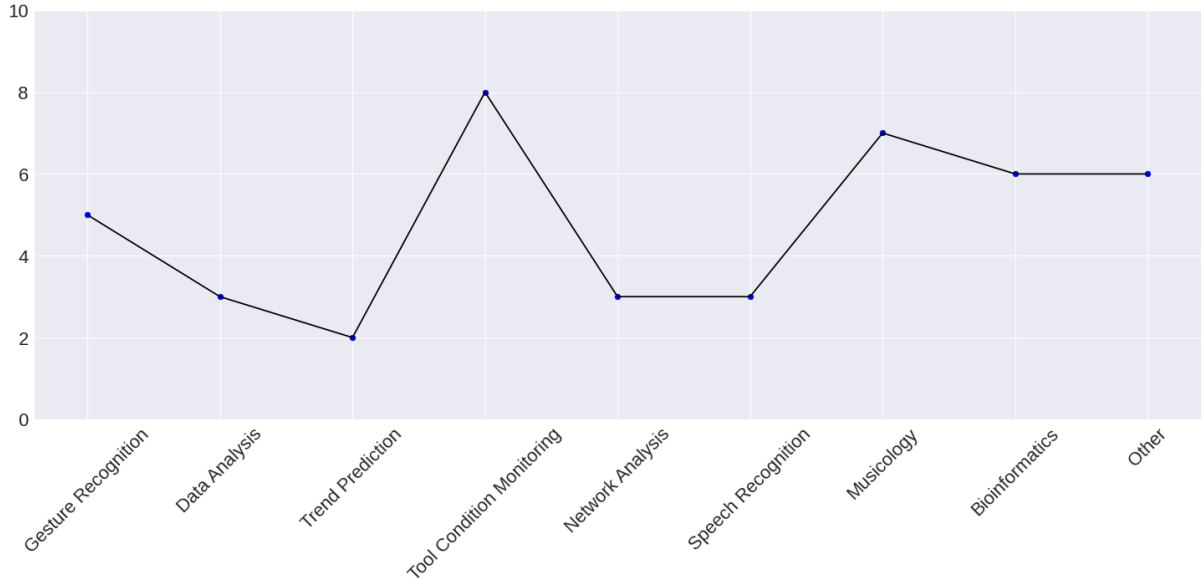

Figure 8: HSMM Applications by Field

Bioinformatics

Human genome sequence data is a natural fit for hidden Markov models as the sheer volume of sequences is quickly processed by their computational efficiency. The noise produced by physical detection methods used in sub-fields like onceogenomics (cancer-associated genes) or even non-human genomics can be mitigated by hidden Markov models via learning copy number changes and then similarity searching them along the sequence [Sei+12].

6 Conclusions

Hidden Markov model's stochasticity introduces a degree of complexity to its processing, however its widespread use across fields shows it is favored for modeling latent state of autonomous systems. The ideas broached in this paper merely touch on the basics of hidden Markov models, however even a foundational understanding can provide a new perspective on modelling systems. This is corroborated

by the breakthroughs made in fields like speech recognition and bioinformatics dating from decades ago to today.

References

[EDN06] Elbestawi, M. A., Dumitrescu, M. and Ng, E.-G. 'Tool Condition Monitoring in Machining'. In: *Condition Monitoring and Control for Intelligent Manufacturing*. Ed. by Wang, L. and Gao, R. X. London: Springer London, 2006, pp. 55–82.

[FST98] Fine, S., Singer, Y. and Tishby, N. 'The Hierarchical Hidden Markov Model: Analysis and Applications'. In: *Machine Learning* vol. 32, no. 1 (July 1998), pp. 41–62.

[GJ97] Ghahramani, Z. and Jordan, M. I. 'Factorial Hidden Markov Models'. In: *Machine Learning* vol. 29, no. 2 (Nov. 1997), pp. 245–273.

[JM09] Jurafsky, D. and Martin, J. H. *Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition*. Upper Saddle River, N.J.: Pearson Prentice Hall, 2009.

[Li+17] Li, X. et al. 'Optimal Cost-Effective Maintenance Policy for a Helicopter Gearbox Early Fault Detection under Varying Load'. In: *Mathematical Problems in Engineering* vol. 2017 (Mar. 2017), pp. 1–16.

[LJ16] Lee, L.-M. and Jean, F.-R. 'High-order hidden Markov model for piecewise linear processes and applications to speech recognition'. In: *The Journal of the Acoustical Society of America* vol. 140, no. 2 (2016), EL204–EL210. eprint: https://doi.org/10.1121/1.4960107.

[Mor20] Mor, B. 'A Systematic Review of Hidden Markov Models and Their Applications'. In: *Archives of Computational Methods in Engineering* vol. OnlineFirst (May 2020), pp. 1–20.

[Rab89] Rabiner, L. R. 'A tutorial on hidden Markov models and selected applications in speech recognition'. In: *Proceedings of the IEEE* vol. 77, no. 2 (1989), pp. 257–286.

[Sei+12] Seifert, M. et al. 'Parsimonious Higher-Order Hidden Markov Models for Improved Array-CGH Analysis with Applications to Arabidopsis thaliana'. In: *PLOS Computational Biology* vol. 8, no. 1 (Jan. 2012), pp. 1–15.

[SK95] Sin, B. and Kim, J. H. 'Nonstationary hidden Markov model'. In: *Signal Processing* vol. 46, no. 1 (1995), pp. 31–46.

[SWF14] Stanculescu, I., Williams, C. K. I. and Freer, Y. 'Autoregressive Hidden Markov Models for the Early Detection of Neonatal Sepsis'. In: *IEEE Journal of Biomedical and Health Informatics* vol. 18, no. 5 (2014), pp. 1560–1570.

[WH91] Williams, C. and Hinton, G. 'Mean field networks that learn to discriminate temporally distorted strings'. English. In: *Connectionist Models*. 1991, pp. 18–22.

[Yan+88] Yang, W. .-. et al. 'Hidden Markov model for Mandarin lexical tone recognition'. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* vol. 36, no. 7 (1988), pp. 988–992.