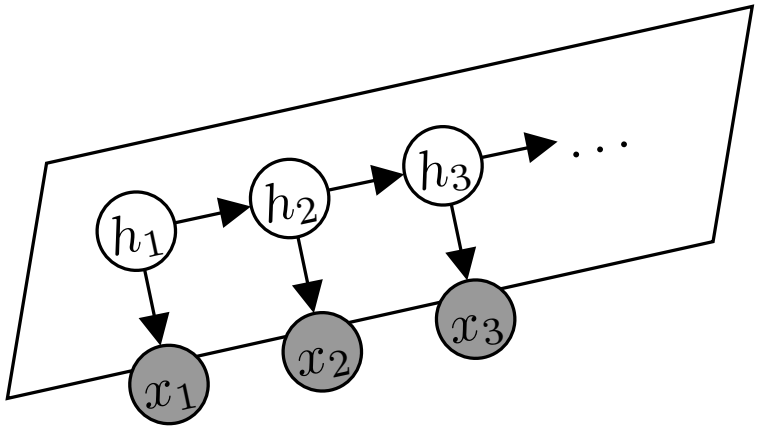
A spectral algorithm for learning hidden Markov models



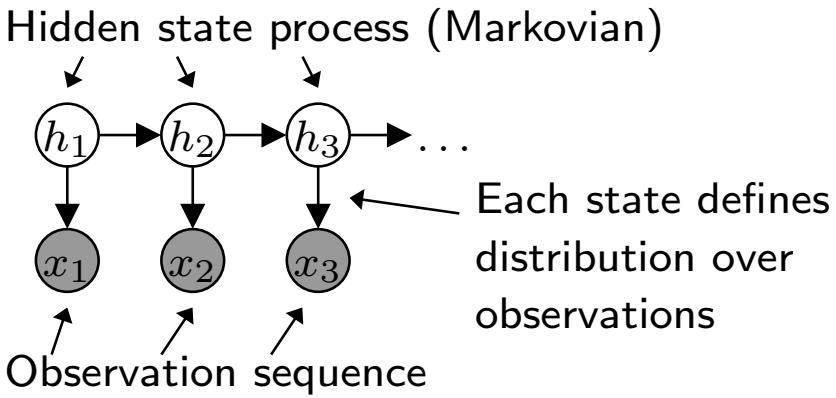
Daniel Hsu
UCSD

Sham M. Kakade
TTI-C

Tong Zhang
Rutgers

Motivation

- Hidden Markov Models (HMMs) – popular model for sequential data (*e.g.* speech, bio-sequences, natural language)



- Hidden state sequence *not* observed; hence, *unsupervised learning*.

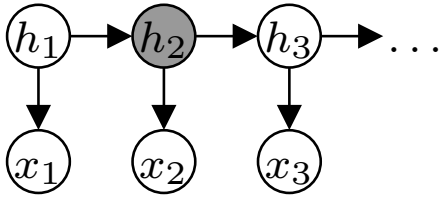
Motivation

- Why HMMs?
  - Handle temporally-dependent data
  - Succinct "factored" representation when state space is low-dimensional (*c.f.* autoregressive model)
- Some uses of HMMs:
  - Monitor "belief state" of dynamical system
  - Infer latent variables from time series
  - Density estimation

Preliminaries: parameters of discrete HMMs

Sequences of hidden states $(h_1, h_2, \ldots)$ and observations $(x_1, x_2, \ldots)$.

- Hidden states $\{1, 2, \ldots, m\}$;
  Observations $\{1, 2, \ldots, n\}$
- Conditional independences



- Initial state distribution $\vec{\pi} \in \mathbb{R}^m$
  $\vec{\pi}_i = \Pr[h_1 = i]$
- Transition matrix $T \in \mathbb{R}^{m \times m}$
  $T_{ij} = \Pr[h_{t+1} = i | h_t = j]$
- Observation matrix $O \in \mathbb{R}^{n \times m}$
  $O_{ij} = \Pr[x_t = i | h_t = j]$

$$\Pr[x_{1:t}] = \sum_{h_1} \Pr[h_1] \cdot \sum_{h_2} \Pr[h_2|h_1] \Pr[x_1|h_1] \cdot \ldots \cdot \sum_{h_{t+1}} \Pr[h_{t+1}|h_t] \Pr[x_t|h_t]$$

Preliminaries: learning discrete HMMs

- Popular heuristic: Expectation-Maximization (EM) (*a.k.a.* Baum-Welch algorithm)
- Computationally hard in general under cryptographic assumptions (Terwijn, '02)
- This work: Computationally efficient algorithm with learning guarantees for *invertible HMMs*:

  Assume $T \in \mathbb{R}^{m \times m}$ and $O \in \mathbb{R}^{n \times m}$ have rank m.

  (here, $n \geq m$).

Our contributions

- **Simple** and **efficient** algorithm for learning invertible HMMs.
- **Sample complexity bounds** for
  - Joint probability estimation (total variation distance):

$$\sum_{x_{1:t}} |\Pr[x_{1:t}] - \widehat{\Pr}[x_{1:t}]| \leq \epsilon$$

    (relevant for density estimation tasks)
  - Conditional probability estimation (KL distance):

$$KL(\Pr[x_t|x_{1:t-1}] \| \widehat{\Pr}[x_t|x_{1:t-1}]) \leq \epsilon$$

    (relevant for next-symbol prediction / "belief states")
- Connects **subspace identification** to **observation operators**.

Outline

1. Motivation and preliminaries
2. Discrete HMMs: key ideas
3. Observable representation for HMMs
4. Learning algorithm and guarantees
5. Conclusions and future work

Outline

1. Motivation and preliminaries
2. **Discrete HMMs: key ideas**
3. Observable representation for HMMs
4. Learning algorithm and guarantees
5. Conclusions and future work

Discrete HMMs: linear model

Let $\vec{h}_t \in \{\vec{e}_1, \ldots, \vec{e}_m\}$ and $\vec{x}_t \in \{\vec{e}_1, \ldots, \vec{e}_n\}$ (coord. vectors in $\mathbb{R}^m$ and $\mathbb{R}^n$)

$$\mathbb{E}[\vec{h}_{t+1}|\vec{h}_t] \;=\; T\;\vec{h}_t \quad \text{and} \quad \mathbb{E}[\vec{x}_t|\vec{h}_t] \;=\; O\;\vec{h}_t$$

In expectation, dynamics and observation process are linear!

e.g. conditioned on $h_t = 2$ (*i.e.* $\vec{h_t} = \vec{e}_2$):

$$\mathbb{E}[\vec{h}_{t+1}|\vec{h}_t] \;=\; \begin{bmatrix} 0.2 & 0.4 & 0.6 \\ 0.3 & 0.6 & 0.4 \\ 0.5 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.4 \\ 0.6 \\ 0 \end{bmatrix}$$

Upshot: Can borrow "subspace identification" techniques from linear systems theory.

Discrete HMMs: linear model

Exploiting linearity

- Subspace identification for general linear models
  - Use SVD to discover subspace containing relevant states, then learn effective transition and observation matrices (Ljung, '87).
  - Analysis typically assumes additive noise (independent of state), *e.g.* Gaussian noise (Kalman filter); not applicable to HMMs.
- This work: Use subspace identification, then learn alternative HMM parameterization.

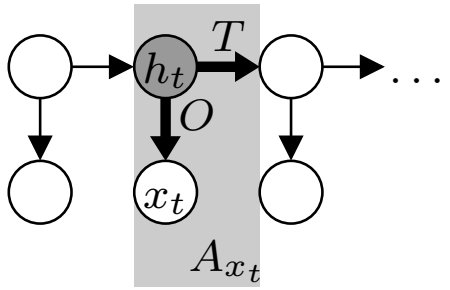
Discrete HMMs: observation operators

For $x \in \{1, \ldots, n\}$: define

$$A_x \triangleq \begin{bmatrix} & T & \end{bmatrix} \begin{bmatrix} O_{x,1} & & 0 \\ & \ddots & \\ 0 & & O_{x,m} \end{bmatrix} \in \mathbb{R}^{m \times m}$$

$$[A_x]_{i,j} = \Pr[\, h_{t+1} = i \ \wedge \ x_t = x \mid h_t = j \,].$$

The $\{A_x\}$ are *observation operators* (Schützenberger, '61; Jaeger, '00).



Discrete HMMs: observation operators

Using observation operators

Matrix multiplication handles "local" marginalization of hidden variables: *e.g.*

$$\begin{aligned}\Pr[x_1, x_2] &= \sum_{h_1} \Pr[h_1] \cdot \sum_{h_2} \Pr[h_2|h_1] \Pr[x_1|h_1] \cdot \sum_{h_3} \Pr[h_3|h_2] \Pr[x_2|h_2] \\ &= \vec{1}_m^\top A_{x_2} A_{x_1} \vec{\pi}\end{aligned}$$

where $\vec{1}_m \in \mathbb{R}^m$ is the all-ones vector.

Upshot: The $\{A_x\}$ contain the same information as T and O.

Discrete HMMs: observation operators

Learning observation operators

- Previous methods face the problem of discovering and extracting the relationship between hidden states and observations (Jaeger, '00).
  - Various techniques proposed (*e.g.* James and Singh, '04; Wiewiora, '05).
  - Formal guarantees were unclear.
- This work: Combine subspace identification with observation operators to yield observable HMM representation that is *efficiently learnable*.

Outline

1. Motivation and preliminaries
2. Discrete HMMs: key ideas
3. **Observable representation for HMMs**
4. Learning algorithm and guarantees
5. Conclusions and future work

Observable representation for HMMs

Key rank condition: require $T \in \mathbb{R}^{m \times m}$ and $O \in \mathbb{R}^{n \times m}$ to have rank m
(rules out pathological cases from hardness reductions)

Define $P_1 \in \mathbb{R}^n$, $P_{2,1} \in \mathbb{R}^{n \times n}$, $P_{3,x,1} \in \mathbb{R}^{n \times n}$ for $x = 1, \ldots, n$ by

$$\begin{aligned} [P_1]_i &= \Pr[x_1 = i] \\ [P_{2,1}]_{i,j} &= \Pr[x_2 = i, x_1 = j] \\ [P_{3,x,1}]_{i,j} &= \Pr[x_3 = i, x_2 = x, x_1 = j] \end{aligned}$$

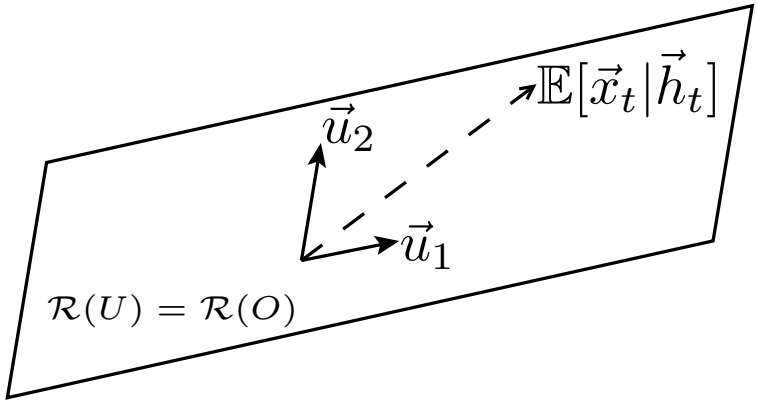
(probabilities of singletons, doubles, and triples).

Claim: Can recover equivalent HMM parameters from P_1, $P_{2,1}$, $\{P_{3,x,1}\}$, and *these quantities can be estimated from data*.

Observable representation for HMMs

"Thin" SVD: $P_{2,1} = U\Sigma V^\top$ where $U = [\vec{u}_1 | \ldots | \vec{u}_m] \in \mathbb{R}^{n \times m}$

Guaranteed m non-zero singular values by rank condition.



New parameters (based on U) implicitly transform hidden states

$$\vec{h}_t \quad \mapsto \quad (U^\top O)\vec{h}_t \quad = \quad U^\top \mathbb{E}[\vec{x}_t|\vec{h}_t]$$

(*i.e.* change to coordinate representation of $\mathbb{E}[\vec{x}_t|\vec{h}_t]$ w.r.t. $\{\vec{u}_1, \ldots, \vec{u}_m\}$).

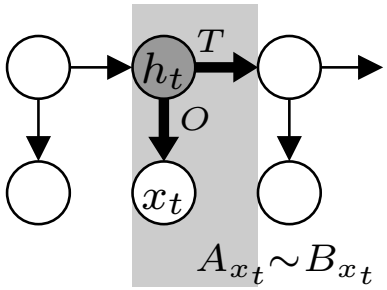
Observable representation for HMMs

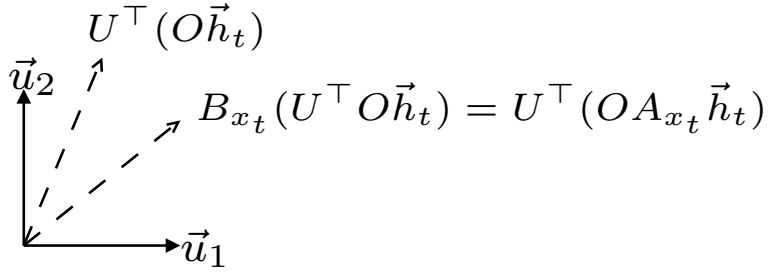
For each $x = 1, \ldots, n$,

$$B_x \triangleq (U^\top P_{3,x,1})\ (U^\top P_{2,1})^+ \qquad (X^+ \text{ is pseudoinv. of } X)$$
$$= (U^\top O)\ A_x\ (U^\top O)^{-1}\ . \qquad \text{(algebra)}$$

The B_x operate in the coord. system defined by $\{\vec{u}_1, \ldots, \vec{u}_m\}$ (columns of U).

$$\Pr[x_{1:t}] \ = \ \vec{1}_m^\top A_{x_t} \ldots A_{x_1} \vec{\pi} \ = \ \vec{1}_m^\top (U^\top O)^{-1} B_{x_t} \ldots B_{x_1} (U^\top O) \vec{\pi}$$





Upshot: Suffices to learn $\{B_x\}$ instead of $\{A_x\}$.

Outline

1. Motivation and preliminaries
2. Discrete HMMs: key ideas
3. Observable representation for HMMs
4. **Learning algorithm and guarantees**
5. Conclusions and future work

Learning algorithm

The algorithm

1. Look at triples of observations (x_1, x_2, x_3) in data;
   estimate frequencies $\widehat{P}_1$, $\widehat{P}_{2,1}$, and $\{\widehat{P}_{3,x,1}\}$
2. Compute SVD of $\widehat{P}_{2,1}$ to get matrix of top m singular vectors $\widehat{U}$
   ("subspace identification")
3. Compute $\widehat{B}_x \triangleq (\widehat{U}^\top \widehat{P}_{3,x,1})(\widehat{U}^\top \widehat{P}_{2,1})^+$ for each x
   ("observation operators")
4. Compute $\widehat{b}_1 \triangleq \widehat{U}^\top \widehat{P}_1$ and $\widehat{b}_\infty \triangleq (\widehat{P}_{2,1}^\top \widehat{U})^+ \widehat{P}_1$

Learning algorithm

- Joint probability calculations:

$$\widehat{\Pr}[x_1, \ldots, x_t] \triangleq \widehat{b}_\infty^\top \widehat{B}_{x_t} \ldots \widehat{B}_{x_1} \widehat{b}_1.$$

- Conditional probabilities: Given $x_{1:t-1}$,

$$\widehat{\Pr}[x_t | x_{1:t-1}] \triangleq \widehat{b}_\infty^\top \widehat{B}_{x_t} \widehat{b}_t$$

where

$$\widehat{b}_t \triangleq \frac{\widehat{B}_{x_{t-1}} \ldots \widehat{B}_{x_1} \widehat{b}_1}{\widehat{b}_\infty^\top \widehat{B}_{x_{t-1}} \ldots \widehat{B}_{x_1} \widehat{b}_1} \approx (U^\top O) \mathbb{E}[\vec{h}_t | x_{1:t-1}].$$

"Belief states" $\widehat{b}_t$ linearly related to conditional hidden states.
(b_t live in hypercube $[-1, +1]^m$ instead simplex Δ^m)

Sample complexity bound

Joint probability accuracy: with probability $\geq 1 - \delta$,

$$O\left(\frac{t^2}{\epsilon^2} \cdot \left(\frac{m}{\sigma_m(O)^2 \sigma_m(P_{2,1})^4} + \frac{m \cdot n_0}{\sigma_m(O)^2 \sigma_m(P_{2,1})^2}\right) \cdot \log \frac{1}{\delta}\right)$$

observation triples sampled from the HMM suffices to guarantee

$$\sum_{x_1,\ldots,x_t} |\Pr[x_1, \ldots, x_t] - \widehat{\Pr}[x_1, \ldots, x_t]| \leq \epsilon.$$

- m: number of states
- n_0: number of observations that account for most of the probability mass
- $\sigma_m(M)$: mth largest singular value of matrix M

Also have a sample complexity bound for conditional probability accuracy.

Conclusions and future work

Summary:

- Simple and efficient learning algorithm for invertible HMMs (sample complexity bounds, streaming implementation, etc.)
- Observable representation lets us avoid estimating T and O (*n.b.* could recover these if we really wanted, but less stable).
- SVD subspace captures dynamics of observable representation.
- Salvages old ideas and techniques from automata and control theory.

Future work:

- Improve efficiency, stability
- General linear dynamical models
- Behavior of EM under the rank condition?

Thanks!

Sample complexity bound

Conditional probability accuracy: with probability $\geq 1 - \delta$,

$$\mathrm{poly}(1/\epsilon, 1/\alpha, 1/\gamma, 1/\sigma_m(O), 1/\sigma_m(P_{2,1})) \cdot (m^2 + mn_0) \cdot \log \frac{1}{\delta}$$

observation triples sampled from the HMM suffices to guarantee

$$KL(\Pr[x_t|x_{1:t-1}] \| \widehat{\Pr}[x_t|x_{1:t-1}]) \leq \epsilon$$

for all t and all history sequences $x_{1:t-1}$.

- n_0: number of observations that account for $1 - \varepsilon$ of total probability mass, where $\varepsilon = \sigma_m(O)\sigma_m(P_{2,1})\epsilon/(4\sqrt{m})$
- $\gamma = \inf_{\vec{v}:\|\vec{v}\|_1=1} \|O\vec{v}\|_1$ "value of observation" (Even-Dar *et al*, '07)
- $\alpha = \min_{x,i,j}[A_x]_{ij}$ (stochasticity requirement)

Computer simulations

- Simple "cycle" HMM: $m = 9$ states, $n = 180$ observations
- Train using 20000 sequences of length 100 (generated by true model)
- Results (average log-loss on 2000 test sequences of length 100):

True model	4.79 nats per symbol
EM, initialized with true model	4.79 nats per symbol
EM, random initializations	5.15 nats per symbol
Our algorithm	4.88 nats per symbol
Our algorithm, followed by EM	4.81 nats per symbol