HIDDEN MARKOV MODELS

Sergey Nikolenko

Harbour Space University, Barcelona, Spain
March 28, 2017

MARKOV CHAINS AND HIDDEN MARKOV MODELS

MARKOV CHAINS

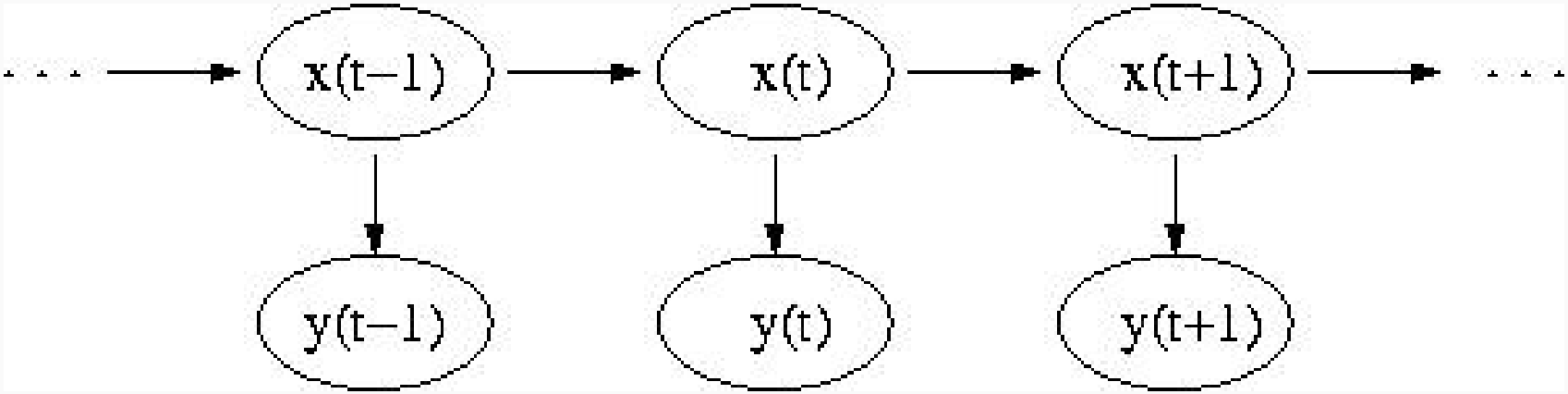
- A Markov chain is defined by initial probability distribution $p^0(x)$ and transition probabilities $T(x'; x)$.
- $T(x'; x)$ is the distribution of the next element in the chain depending on the previous one; distribution on step $(t+1)$ is

$$p^{t+1}(x') = \int T(x'; x) p^t(x) dx.$$

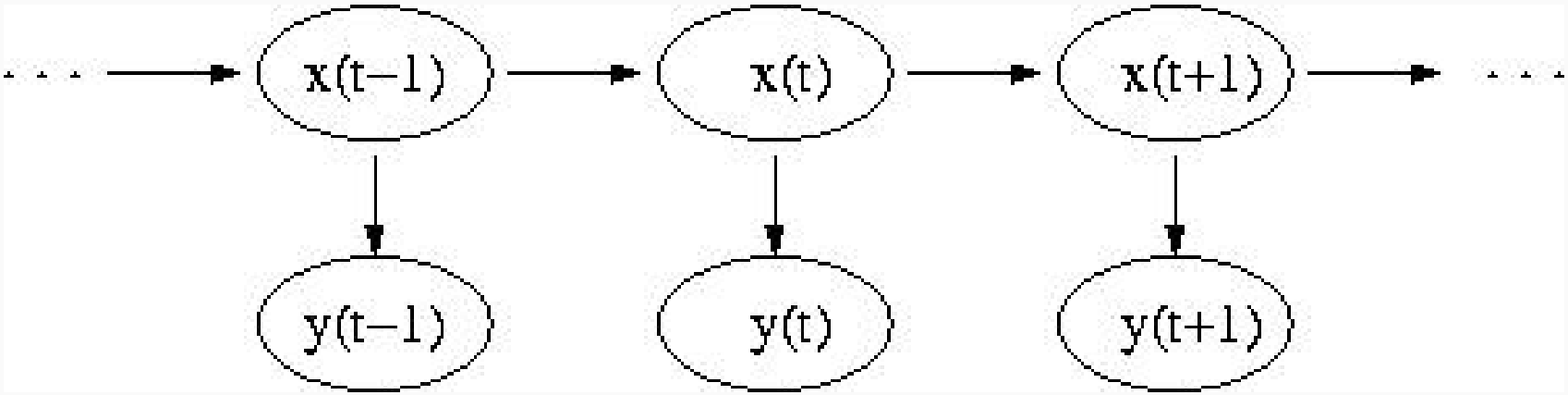
- In the discrete case, $T(x'; x)$ is a matrix of probabilities $p(x' = i | x = j)$.

DISCRETE MARKOV CHAINS

- We are in the discrete case.
- A Markov model is when we can observe certain functions of a Markov chain.



- Here $x(t)$ is the process (chain states) itself, and $y(t)$ are observables.
- The problem is to find hidden parameters of the process.



- Markov property: next state does not depend on the history, only on the previous state:

$$p(x(t) = x_j | x(t-1) = x_{j_{t-1}}, \dots, x(1) = x_{j_1}) = \\ = p(x(t) = x_j | x(t-1) = x_{j_{t-1}}).$$

- Moreover, these probabilities $a_{ij} = p(x(t) = x_j | x(t-1) = x_i)$ do not depend on t.
- These probabilities comprise the transition matrix $A = (a_{ij})$, with natural properties $a_{ij} \geq 0$, $\sum_j a_{ij} = 1$.

DIRECT PROBLEM

- Natural problem: what is the probability to get a certain sequence of events?
- I.e., for a sequence $Q = q_{i_1} \dots q_{i_k}$ find

$$p(Q|\text{model}) = p(q_{i_1})p(q_{i_2}|q_{i_1}) \dots p(q_{i_k}|q_{i_{k-1}}).$$

- Looks trivial. What's hard in the real world?

HIDDEN MARKOV MODELS

- In the real world we do not know the model.
- And, moreover, we do not observe $x(t)$, i.e., real model states, but rather $y(t)$, i.e., observe functions of them (data).
- Example: speech recognition.

PROBLEMS IN HIDDEN MARKOV MODELS

- First: find the probability of a sequence of observations in a given model.
- Second: find the “optimal” sequence of states in a given model and a given sequence of observations.
- Third: find the maximum likelihood model (model parameters).

STATES AND OBSERVABLES

- $X = \{x_1, \dots, x_n\}$ — set of states.
- $V = \{v_1, \dots, v_m\}$ — alphabet from which we choose observables y (set of values of y).
- q_t — state at time t, y_t — observable at time t.

DISTRIBUTIONS

- $a_{ij} = p(q_{t+1} = x_j | q_t = x_i)$ — transition probability from i to j.
- $b_j(k) = p(v_k | x_j)$ — probability to get data v_k in state j.
- Initial distribution $\pi = \{\pi_j\}$, $\pi_j = p(q_1 = x_j)$.
- We denote the data by $D = d_1 \ldots d_T$ (sequence of observables, d_i take values from V).

LEARNING HIDDEN MARKOV MODELS

PROBLEM

- We can now formalize the problem setting.
- First problem: for a given model $\lambda = (A, B, \pi)$ and sequence D, find $p(D|\lambda)$. By itself it simply shows how well the model fits this data.
- Second problem: for a given model λ and sequence D find the "optimal" sequence of states $Q = q_1 \ldots q_T$. Two kinds of optimality: "bitwise" and general.
- Third problem: optimize model parameters $\lambda = (A, B, \pi)$ in order to maximize $p(D|\lambda)$ for a given D (find the maximum likelihood model). This is the main problem, training hidden Markov models.

FIRST PROBLEM

- Formally the first problem looks like

$$p(D|\lambda) = \sum_{Q} p(D|Q,\lambda)p(D|\lambda) =$$

$$= \sum_{q_1,\dots,q_T} b_{q_1}(d_1)\dots b_{q_T}(d_T)\pi_{q_1} a_{q_1 q_2}\dots a_{q_{T-1}q_T}.$$

FIRST PROBLEM

- This is a marginalization problem.
- We use the so-called *forward–backward procedure*, in essence dynamical programming on a lattice.
- We will sequentially compute intermediate values of the form

$$\alpha_t(i) = p(d_1 \dots d_t, q_t = x_i | \lambda),$$

  i.e., the required probabilities with account for current state.

SOLVING THE FIRST PROBLEM

- Initialize $\alpha_1(i) = \pi_i b_i(d_1)$.
- Induction step:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{n} \alpha_t(i) a_{ij}\right] b_j(d_{t+1}).$$

- After we get to step T, we can compute what we need:

$$p(D|\lambda) = \sum_{i=1}^{n} \alpha_T(i).$$

- This is simply the forward pass, we did not need a backward pass here.
- What would it compute?

BACKWARD PASS

- It would compute conditional probabilities $\beta_t(i) = p(d_{t+1} \dots d_T | q_t = x_i, \lambda)$.
- We can initialize $\beta_T(i) = 1$ and proceed by induction:

$$\beta_t(i) = \sum_{j=1}^{n} a_{ij} b_j(d_{t+1}) \beta_{t+1}(j).$$

- We'll need it later to solve the second and third problems.

TWO VERSIONS OF THE SECOND PROBLEM

- There are two versions for the second problem.
- First, solve it “bit by bit”: “what is the most probable state at time j?”
- Second, solve it “in general”: “what is the most probable sequence of states?”.

BITWISE SOLUTION

- Consider auxiliary variables

$$\gamma_t(i) = p(q_t = x_i | D, \lambda).$$

- The problem is to find

$$q_t = \arg\max_{1 \leq i \leq n} \gamma_t(i), \quad 1 \leq t \leq T.$$

- How can we do it?

BITWISE SOLUTION

- We express them via α and β:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{p(D|\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{n} \alpha_t(i)\beta_t(i)}.$$

- The denominator does not matter since we need $\arg\max$.

SOLVING FOR THE SEQUENCE

- To find the most probable sequence, we will use the so-called *Viterbi algorithm* (that is, dynamic programming).
- Now auxiliary variables are

$$\delta_t(i) = \max_{q_1,\cdots,q_{t-1}} p\left(q_1 q_2 \ldots q_t = x_i, d_1 d_2 \ldots d_t | \lambda\right).$$

SOLVING FOR THE SEQUENCE

- That is, $\delta_t(i)$ is the maximal probability to reach state x_i on step t among all paths with given observables.
- By induction:

$$\delta_{t+1}(j) = \left[\max_i \delta_t(i) a_{ij}\right] b_j(d_{t+1}).$$

- Note that we also need to remember the arguments, not only values; $\psi_t(j)$ on the next slide.

SOLVING FOR THE SEQUENCE: ALGORITHM

- Initialize $\delta_1(i) = \pi_i b_i(d_1)$, $\psi_1(i) = []$.
- Induction:

$$\delta_t(j) = \max_{1 \le i \le n} \left[\delta_{t-1}(i) a_{ij}\right] b_j(d_t),$$

$$\psi_t(j) = \arg\max_{1 \le i \le n} \left[\delta_{t-1}(i) a_{ij}\right].$$

- When we reach step T, final step:

$$p^* = \max_{1 \le i \le n} \delta_T(i), \qquad q_T^* = \arg\max_{1 \le i \le n} \delta_T(i).$$

- And the sequence follows: $q_t^* = \psi_{t+1}(q_{t+1}^*)$.

THIRD PROBLEM

- We cannot find a global maximum of $p(D|\lambda)$ analytically.
- We will use local optimization.
- The *Baum–Welch algorithm*: a special case of EM.

AUXILIARY VARIABLES

- Now auxiliary variables are probabilities of the event that at time t we are in state x_i, and at time $t+1$ — in state x_j:

$$\xi_t(i,j) = p(q_t = x_i, q_{t+1} = x_j | D, \lambda).$$

- Rewriting via already familiar variables:

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(d_{t+1}) \beta_{t+1}(j)}{p(D|\lambda)} = \frac{\alpha_t(i) a_{ij} b_j(d_{t+1}) \beta_{t+1}(j)}{\sum_i \sum_j \alpha_t(i) a_{ij} b_j(d_{t+1}) \beta_{t+1}(j)}.$$

- Note also that $\gamma_t(i) = \sum_j \xi_t(i,j)$.

IDEA

- $\sum_t \gamma_t(i)$ is the expected number of transitions from states x_i; $\sum_t \xi_t(i,j)$, from x_i to x_j.
- On the M-step we will reestimate the probabilities:

$$\bar{\pi}_i = \text{expected frequency of } x_i \text{ on step } 1 = \gamma_1(i),$$

$$\bar{a}_{ij} = \frac{\text{no. of transitions from } x_i \text{ to } x_j}{\text{no. of transitions from } x_i} = \frac{\sum_t \xi_t(i,j)}{\sum_t \gamma_t(i)}.$$

$$\bar{b}_j(k) = \frac{\text{no. of times in } x_i \text{ observing } v_k}{\text{no. of times in } x_i} = \frac{\sum_{t:d_t=v_k} \gamma_t(i)}{\sum_t \gamma_t(i)}.$$

- EM-algorithm: start with $\lambda = (A, B, \pi)$, compute $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$, recompute the parameters again, and so on.

Kullback--Leibler divergence

- Kullback–Leibler divergence is an information theoretic measure of how different two distributions are:

$$D_{KL}(p_1, p_2) = \sum_x p_1(x) \log \frac{p_1(x)}{p_2(x)}.$$

- It is nonnegative and equal to zero only if $p_1 \equiv p_2$ (with probability 1).

- We define

$$p_1(Q) = \frac{p(Q, D|\lambda)}{p(D|\lambda)}, \quad p_2(Q) = \frac{p(Q, D|\lambda')}{p(D|\lambda')}.$$

- Then p_1 and p_2 are distributions, and the Kullback–Leibler divergence is

$$0 \leq D_{LK}(\lambda, \lambda') = \sum_Q \frac{p(Q, D|\lambda)}{p(D|\lambda)} \log \frac{p(Q, D|\lambda)p(D|\lambda')}{p(Q, D|\lambda')p(D|\lambda)} =$$

$$= \log \frac{p(D|\lambda')}{p(D|\lambda)} + \sum_Q \frac{p(Q, D|\lambda)}{p(D|\lambda)} \log \frac{p(Q, D|\lambda)}{p(Q, D|\lambda')}.$$

AUXILIARY FUNCTION

- We introduce the auxiliary function

$$Q(\lambda, \lambda') = \sum_Q p(Q|D, \lambda) \log p(Q|D, \lambda').$$

- Then the inequality implies that

$$\frac{Q(\lambda, \lambda') - Q(\lambda, \lambda)}{p(D|\lambda)} \leq \log \frac{p(D|\lambda')}{p(D|\lambda)}.$$

- That is, if $Q(\lambda, \lambda') > Q(\lambda, \lambda)$ then $p(D|\lambda') > p(D|\lambda)$.
- That is, if we maximize $Q(\lambda, \lambda')$ w.r.t. λ', we will be moving in the right direction.

FUNCTION q

- We need to maximize $Q(\lambda, \lambda')$. We rewrite

$$Q(\lambda, \lambda') = \sum_Q p(Q|D, \lambda) \log p(Q|D, \lambda') =$$

$$= \sum_Q p(Q|D, \lambda) \log \pi_{q_1} \prod_t a_{q_{t-1} q_t} b_{q_t}(d_t) =$$

$$= \sum_Q p(Q|D, \lambda) \log \pi_{q_1} + \sum_Q p(Q|D, \lambda) \sum_t \log a_{q_{t-1} q_t} b_{q_t}(d_t).$$

- The latter expression is easy to differentiate w.r.t. a_{ij}, $b_i(k)$, and π_i, add the corresponding Lagrange multipliers, and solve.
- We'll get exactly the Baum–Welch algorithm (check it!).

EXTENSIONS OF MARKOV MODELS

CONTINUOUS DENSITIES OF OBSERVABLES

- We had discrete observables with probabilities $B = (b_j(k))$.
- In reality we often observe continuous signals, and they are hard to discretize.
- But the chain itself can remain discrete, i.e., we pass to $b_j(D)$.

SPECIAL FORM OF THE DENSITY

- Not for all densities we know Baum–Welch generalizations.
- But for many, e.g. if $b_j(D)$ can be represented as a convex combination

$$b_j(D) = \sum_{m=1}^{M} c_{jm} \mathcal{P}(D, \mu_{jm}, \sigma_{jm}),$$

  where c_{jm} are mixture coefficients ($\sum_m c_{jm} = 1$), and $\mathcal{P}$ is a convex distribution with mean μ and variance σ (e.g., Gaussian).
- This can serve as a good approximation for nearly all distributions.

AUXILIARY VARIABLES

- $\gamma_t(j, m)$ — probability to be in state j at time t, and D was generated from the mth component of the mixture.
- Formally,

$$\gamma_t(j, m) = \left[\frac{\alpha_t(j)\beta_t(j)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)} \right] \left[\frac{c_{jm}\mathcal{P}(d_t, \mu_{jm}, \sigma_{jm})}{\sum_{m=1}^{M} c_{jm}\mathcal{P}(d_t, \mu_{jm}, \sigma_{jm})} \right].$$

- If $M = 1$, it's the same as $\gamma_t(j)$.

- We need to recompute $b_j(D)$, i.e., recompute c_{jm}, μ_{jm}, and σ_{jm}.
- Very natural formulas:

$$\bar{c}_{jm} = \frac{\sum_{t=1}^{T} \gamma_t(j,m)}{\sum_{t=1}^{T} \sum_{m=1}^{M} \gamma_t(j,m)},$$

$$\bar{\mu}_{jm} = \frac{\sum_{t=1}^{T} \gamma_t(j,m) \cdot d_t}{\sum_{t=1}^{T} \gamma_t(j,m)},$$

$$\bar{\sigma}_{jm} = \frac{\sum_{t=1}^{T} \gamma_t(j,m) \cdot (d_t - \mu_{jm})(d_t - \mu_{jm})^t}{\sum_{t=1}^{T} \gamma_t(j,m)}.$$

PROBLEM

- How do we model how long we stay at a certain state?
- In the discrete case the probability to spend d steps at state i is

$$p_i(d) = a_{ii}^{d-1}(1 - a_{ii}).$$

- But for most signals this exponential distribution is wrong!
- We'd like to specify the distribution $p_i(d)$ explicitly.

AUXILIARY VARIABLES

- We introduce variables

$$\alpha_t(i) = p(d_1 \dots d_t, x_i \text{ ends at time } t|\lambda).$$

- In total, over the first t steps we have visited r states $q_1 \dots q_r$, spending $d_1, \dots, d_r$ steps in them, where

$$q_r = x_i, \qquad \sum_{s=1}^{r} d_s = t.$$

COMPUTING $\alpha_t(i)$

- Then we get

$$\alpha_t(i) = \sum_q \sum_d \pi_{q_1} p_{q_1}(d_1) p(d_1 d_2 \ldots d_{d_1} | q_1)$$
$$a_{q_1 q_2} p_{q_2}(d_2) p(d_{d_1+1} \ldots d_{d_1+d_2} | q_2) \ldots$$
$$\ldots a_{q_{r-1} q_r} p_{q_r}(d_r) p(d_{d_1+\ldots+d_{r-1}+1} \ldots d_t | q_r).$$

COMPUTING $\alpha_t(i)$

- By induction

$$\alpha_t(j) = \sum_{i=1}^{n} \sum_{d=1}^{D} \alpha_{t-d}(j) a_{ij} p_j(d) \prod_{s=t-d+1}^{t} b_j(d_s),$$

  where D is the maximal time spent at any state.

- Then, as before,

$$p(d|\lambda) = \sum_{i=1}^{n} \alpha_T(i).$$

AUXILIARY VARIABLES

- We will need three more variables for recomputation:

$$\alpha_t^*(i) = p(d_1 \dots d_t, x_i \text{ starts at time } t+1|\lambda),$$

$$\beta_t(i) = p(d_{t+1} \dots d_T|x_i \text{ ends at time } t, \lambda),$$

$$\beta_t^*(i) = p(d_{t+1} \dots d_T|x_i \text{ starts at time } t+1, \lambda).$$

- Relations between them:

$$\alpha_t^*(j) = \sum_{i=1}^{n} \alpha_t(i) a_{ij},$$

$$\alpha_t(i) = \sum_{d=1}^{D} \alpha_{t-d}^*(i) p_i(d) \prod_{s=t-d+1}^{t} b_i(d_s),$$

$$\beta_t(i) = \sum_{j=1}^{n} a_{ij} \beta_t^*(j),$$

$$\beta_t^*(i) = \sum_{d=1}^{D} \beta_{t+d}(i) p_i(d) \prod_{s=t+1}^{t+d} b_i(d_s).$$

- Formulas:
  - π_i is the probability that x_i was the first state:

$$\hat{\pi}_i = \frac{\pi_i \beta_0^*(i)}{p(d|\lambda)}.$$

  - a_{ij} — the usual formula, but we have both α and β that says that the new state begins at the next step:

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T} \alpha_t(i) a_{ij} \beta_t^*(j)}{\sum_{k=1}^{n} \sum_{t=1}^{T} \alpha_t(i) a_{ik} \beta_t^*(k)}.$$

- $b_i(k)$ — ratio of the expectation of the number of events $d_t = v_k$ in state x_i to the expectation of any v_j in state x_i:

$$\hat{b}_i(k) = \frac{\sum_{t=1, d_t = v_k}^{T} \left(\sum_{\tau < t} \alpha_\tau^*(i) \beta_\tau^*(i) - \sum_{\tau < t} \alpha_\tau(i) \beta_\tau(i) \right)}{\sum_{k=1}^{m} \sum_{t=1, d_t = v_k}^{T} \left(\sum_{\tau < t} \alpha_\tau^*(i) \beta_\tau^*(i) - \sum_{\tau < t} \alpha_\tau(i) \beta_\tau(i) \right)}.$$

- $p_i(d)$ — ratio of the expectation of the number of times x_i has occurred with duration d to the number of times x_i occurred at all:

$$\hat{p}_i(d) = \frac{\sum_{t=1}^{T} \alpha_t^*(i) p_i(d) \beta_{t+d}(i) \prod_{s=t+1}^{t+d} b_i(d_s)}{\sum_{d=1}^{D} \sum_{t=1}^{T} \alpha_t^*(i) p_i(d) \beta_{t+d}(i) \prod_{s=t+1}^{t+d} b_i(d_s)}.$$

PROS AND CONS

- Very useful approach when $p_i(d)$ is far from exponential.
- But it significantly increases the computational cost (by a factor of D^2).
- And, more importantly, increases the number of parameters, i.e., we need more data to estimate them.

PARAMETRIC STATE DURATION

- To reduce the number of parameters, sometimes we can assume that $p_i(d)$ is a classical distribution with a small number of parameters.
- E.g., $p_i(d)$ can be uniform, or Gaussian ($p_i(d) = \mathcal{N}(d, \mu_i, \sigma_i^2)$), or a Gamma distribution:

$$p_i(d) = \frac{\eta_i^{\nu_i} d^{\nu_i - 1} e^{-\eta_i d}}{\Gamma(\nu_i)}.$$

THANK YOU!

Thank you for your attention!