Hidden Markov Models

He He

New York University

2021-10-13

Table of Contents

Viterbi decoding: setup

Goal: find the highest-scoring sequence under the pairwise scoring function

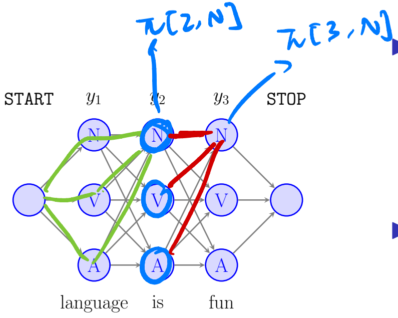
Application: inference in structured prediction (e.g., POS tagging)

Challenge: exponential time complexity using brute force

$$\max_{y \in \mathcal{Y}^m} \sum_{i=1}^{m} s(y_i, y_{i-1})$$

Key idea: dynamic programming

Viterbi decoding: algorithm



$m|Y||Y|$

$O(m|Y|^2)$

- Maximum score of length-j sequences that end at tag t

$$\pi[j, t] \stackrel{\text{def}}{=} \max_{y \in \mathcal{Y}^j, y_j = t} \sum_{i=1}^{j} s(y_i, y_{i-1})$$

len   last tag

- Fill in the chart π recursively

$$\pi[j, t] = \max_{t' \in \mathcal{Y}} \pi[j-1, t'] + s(y_j = t, y_{j-1} = t')$$

  - Backtracking: save argmax in $p[j, t]$

Exponential to polynomial time with exact inference!

Why are we able to do this?

Viterbi decoding: derivation

$$
\begin{aligned}
\pi[j, t] &\stackrel{\text{def}}{=} \max_{y \in \mathcal{Y}^j, y_j = t} \sum_{i=1}^{j} s(y_i, y_{i-1}) \\
&= \max_{y \in \mathcal{Y}^{j-1}} \sum_{i=1}^{j-1} s(y_i, y_{i-1}) + s(y_j = t, y_{j-1}) \\
&= \max_{t' \in \mathcal{Y}} \max_{y \in \mathcal{Y}^{j-2}, y_{j-1} = t'} \sum_{i=1}^{j-1} s(y_i, y_{i-1}) + s(y_j = t, y_{j-1} = t')
\end{aligned}
$$

$$\boxed{\max_{a \in \mathcal{A}} (a + c) = c + \max_{a \in \mathcal{A}} a}$$

$$
\begin{aligned}
&= \max_{t' \in \mathcal{Y}} s(y_j = t, y_{j-1} = t') + \max_{y \in \mathcal{Y}^{j-2}, y_{j-1} = t'} \sum_{i=1}^{j-1} s(y_i, y_{i-1}) \\
&= \max_{t' \in \mathcal{Y}} s(y_j = t, y_{j-1} = t') + \pi[j-1, t']
\end{aligned}
$$

Forward algorithm: setup

CRF learning objective (MLE):

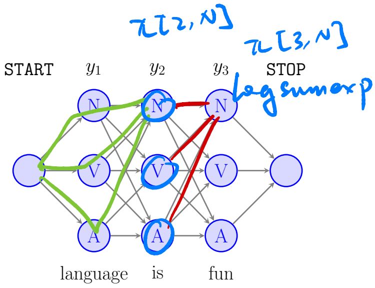
$$
\begin{aligned}
\ell(\theta) &= \sum_{(x,y)\in\mathcal{D}} \log p(y \mid x; \theta) \\
&= \sum_{(x,y)\in\mathcal{D}} \log \frac{\exp\left(\theta \cdot \Phi(x,y)\right)}{\sum_{y'\in\mathcal{Y}^m} \exp\left(\theta \cdot \Phi(x,y)\right)}
\end{aligned}
$$

Goal: compute $\ell(\theta)$ (the forward pass) so that we can do backpropogation

Challenge: exponential time complexity using brute force

If we can compute $\ell(\theta)$ efficiently, computing $\nabla_\theta \ell(\theta)$ will also be efficient. (backpropogation)

Forward decoding: algorithm



- Log of the sum of exponentiated (logsumexp) scores of length-j sequences that end at tag t

$$\pi[j, t] \stackrel{\text{def}}{=} \log \operatorname{sum} \exp_{y \in \mathcal{Y}^j, y_j = t} \sum_{i=1}^{j} s(y_i, y_{i-1})$$

- Fill in the chart π recursively

$$\pi[j, t] = \log \operatorname{sum} \exp_{t' \in \mathcal{Y}} \pi[j-1, t'] + s(y_j = t, y_{j-1} = t')$$

Exponential to polynomial time with exact inference!

Replace max in Viterbi decoding by log sum exp.

Forward decoding: derivation

$$\pi[j, t] \stackrel{\text{def}}{=} \operatorname*{log\,sum\,exp}_{y \in \mathcal{Y}^j, y_j = t} \sum_{i=1}^{j} s(y_i, y_{i-1})$$

$$= \operatorname*{log\,sum\,exp}_{y \in \mathcal{Y}^{j-1}} \sum_{i=1}^{j-1} s(y_i, y_{i-1}) + s(y_j = t, y_{j-1})$$

$$\left[\operatorname*{log\,sum\,exp}_{a \in \mathcal{A}, b \in \mathcal{B}} (a + b) = \operatorname*{log\,sum\,exp}_{a \in \mathcal{A}} \left[\operatorname*{log\,sum\,exp}_{b \in \mathcal{B}} (a + b) \right] \right.$$

$$= \operatorname*{log\,sum\,exp}_{t' \in \mathcal{Y}} \operatorname*{log\,sum\,exp}_{y \in \mathcal{Y}^{j-2}, y_{j-1} = t'} \sum_{i=1}^{j-1} s(y_i, y_{i-1}) + s(y_j = t, y_{j-1} = t')$$

$$\left[\operatorname*{log\,sum\,exp}_{a \in \mathcal{A}} (a + c) = c + \operatorname*{log\,sum\,exp}_{a \in \mathcal{A}} a \right.$$

$$= \operatorname*{log\,sum\,exp}_{t' \in \mathcal{Y}} s(y_j = t, y_{j-1} = t') + \operatorname*{log\,sum\,exp}_{y \in \mathcal{Y}^{j-2}, y_{j-1} = t'} \sum_{i=1}^{j-1} s(y_i, y_{i-1})$$

$$= \operatorname*{log\,sum\,exp}_{t' \in \mathcal{Y}} s(y_j = t, y_{j-1} = t') + \pi[j - 1, t']$$

Table of Contents

Bi-LSTM CRF for sequence labeling

Bi-LSTM tagger: use LSTM as feature extractor

$$p(y_i \mid x) \propto \exp\left(s_{\text{unigram}}(x, y_i, i)\right)$$
$$s_{\text{unigram}}(x, y_i, i) = \theta_{y_i} \cdot \text{Bi-LSTM}(x, i)$$

y_{i-1}

- Learning and inference are similar to MEMM.

Add CRF layer: introduce dependence between neighboring labels

$$p(y \mid x) \propto \exp\left(\sum_{i=1}^{n} s(x, y_i, y_{i-1}, i)\right)$$
$$s(x, y_i, y_{i-1}, i) = s_{\text{unigram}}(x, y_i, i) + s_{\text{bigram}}(y_i, y_{i-1})$$

- Learning and inference: forward and viterbi algorithms

Does it worth it?

Typical neural sequence models:

$$p(y \mid x; \theta) = \prod_{i=1}^{m} p(y_i \mid x, y_{i-1}; \theta)$$

Exposure bias: a learning problem

- Conditions on gold y_{i-1} during training but predicted $\hat{y}_{i-1}$ during test
- Solution: search-aware training

Label bias: a model problem

- Locally normalized models are strictly less expressive than globally normalized *given partial inputs* [Andor+ 16] $p(y_i \mid x_{1:i})$
- Solution: globally normalized models or better encoder

Does it worth it?

Empirical results from [Goyal+ 19]

	Unidirectional	Bidirectional
pretrain-greedy	76.54	92.59
pretrain-beam	77.76	93.29
locally normalized	83.9	**93.76**
globally normalized	**83.93**	93.73

search

Table 2: **Accuracy results on CCG supertagging when initialized with a regular teacher-forcing model.** Reported using *Unidirectional* and *Bidirectional* encoders respectively with fixed attention tagging decoder. *pretrain-greedy* and *pretrain-beam* refer to the output of decoding the initializer model. *locally normalized* and *globally normalized* refer to search-aware soft-beam models

- Partial inputs (unidirectional) + MLE results in poor performance
- Using bidirectional encoder significantly improves results

Table of Contents

Generative vs discriminative models

Generative modeling: $p(x, y)$
Discriminative modeling: $p(y \mid x)$

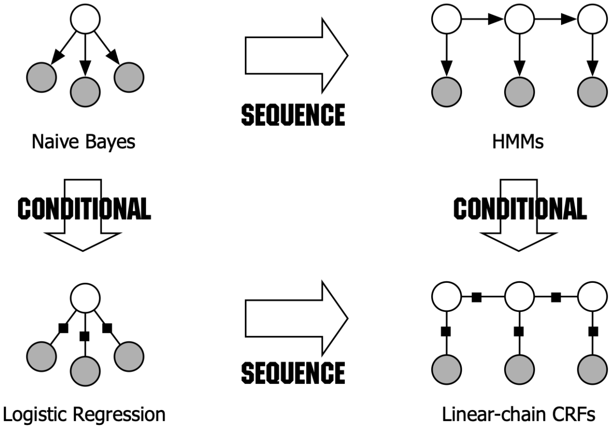


Figure from "An Introduction to Conditional Random Fields for Relational Learning"

Generative modeling for sequence labeling

DT	NN	VBD	IN	DT	NN
the	fox	jumped	over	the	dog

Task: given $x = (x_1, \ldots, x_m) \in \mathcal{X}^m$, predict $y = (y_1, \ldots, y_m) \in \mathcal{Y}^m$

Three questions:

- Modeling: how to define a parametric joint distribution $p(x, y; \theta)$?
- Learning: how to estimate the parameters θ given observed data?
- Inference: how to efficiently find the mostly likely sequence $\arg\max_{y \in \mathcal{Y}^m} p(x, y; \theta)$ given x?

Decompose the joint probability

START $\rightarrow$ DT $\rightarrow$ NN $\rightarrow$ VBD $\rightarrow$ IN $\rightarrow$ DT $\rightarrow$ NN

$\downarrow$ the, fox, jumped, over, the, dog

DT	NN	VBD	IN	DT	NN
the	fox	jumped	over	the	dog

$$
\begin{aligned}
p(x, y) &= p(x \mid y)p(y) \\
&= p(x_1, \ldots, x_m \mid y)p(y) \\
&= \prod_{i=1}^{m} p(x_i \mid y)p(y) \quad \text{Naive Bayes assumption} \\
&= \prod_{i=1}^{m} p(x_i \mid y_i)p(y_1, \ldots, y_m) \quad \text{a word only depends its own tag} \\
&= \prod_{i=1}^{m} p(x_i \mid y_i) \prod_{i=1}^{m} p(y_i \mid y_{i-1}) \quad \text{Markov assumption}
\end{aligned}
$$

Hidden Markov models

Hidden Markov models (HMM):

- Discrete-time, discrete-state Markov chain
- Hidden states $z_i \in \mathcal{Y}$ (e.g. POS tags)
- Observations $x_i \in \mathcal{X}$ (e.g. words)

DT $\downarrow$ the

DT $\to$ NN

$$p(x_{1:m}, y_{1:m}) = \prod_{i=1}^{m} \underbrace{p(x_i \mid y_i)}_{\text{emission probability}} \prod_{i=1}^{m} \underbrace{p(y_i \mid y_{i-1})}_{\text{transition probability}}$$

START

STOP

Model parameters:

- Transition probabilities: $p(y_i = t \mid y_{i-1} = t') = \theta_{t|t'}$ (# params: $|\mathcal{Y}|^2 + 2|\mathcal{Y}|$)
- Emission probabilities: $p(x_i = w \mid y_i = t) = \gamma_{w|t}$ (# params: $|\mathcal{X}| \times |\mathcal{Y}|$)
- $y_0 = *, y_m = \mathtt{STOP}$

Learning: MLE

Data: $\mathcal{D} = \{(x, y)\}$ ($x \in \mathcal{X}^m, y \in \mathcal{Y}^m$) (labeled dataset)
Task: estimate transition probabilities $\theta_{t|t'}$ and emission probabilities $\gamma_{w|t}$

Likelihood: $$\ell(\theta, \gamma) = \sum_{(x,y)\in\mathcal{D}} \left(\sum_{i=1}^{m} \log p(x_i \mid y_i) + \sum_{i=1}^{m} \log p(y_i \mid y_{i-1}) \right)$$

$$\max_{\theta,\gamma} \sum_{(x,y)\in\mathcal{D}} \left(\sum_{i=1}^{m} \log \gamma_{x_i|y_i} + \sum_{i=1}^{m} \log \theta_{y_i|y_{i-1}} \right)$$

$$\text{s.t.} \quad \sum_{w\in\mathcal{X}} \gamma_{w|t} = 1 \quad \forall w \in \mathcal{X}$$

$$\sum_{t\in\mathcal{Y}\cup\{\texttt{STOP}\}} \theta_{t|t'} = 1 \quad \forall t' \in \mathcal{Y} \cup \{*\}$$

MLE solution

$$\begin{array}{cccccccccccc}
\text{DT} & \rightarrow & \text{NN} & \rightarrow & \text{VBD} & \rightarrow & \text{IN} & \rightarrow & \text{DT} & \rightarrow & \text{NN} \\
\downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\
\text{the} & & \text{fox} & & \text{jumped} & & \text{over} & & \text{the} & & \text{dog}
\end{array}$$

Count the occurrence of certain transitions and emissions in the labeled data.

Transition probabilities:

$$\theta_{t|t'} = \frac{\text{count}(t' \rightarrow t)}{\sum_{a \in \mathcal{Y} \cup \{\texttt{STOP}\}} \text{count}(t' \rightarrow a)} \qquad \text{DT} \rightarrow ?$$

Emission probabilities:

$$\gamma_{w|t} = \frac{\text{count}(w, t)}{\sum_{w' \in \mathcal{X}} \text{count}(w', t)}$$

Example: $\theta_{\text{NN}|\text{DT}} = \frac{2}{2} = 1$ $\qquad$ $\gamma_{\text{fox}|\text{NN}} = \frac{1}{2}$

Inference

Task: given model parameters, observe $x \in \mathcal{X}^m$, find the most likely $y \in \mathcal{Y}^m$

$$\begin{aligned}
&\arg\max_{y \in \mathcal{Y}^m} \log p(x, y) \\
&= \arg\max_{y \in \mathcal{Y}^m} \sum_{i=1}^{m} \log p(x_i \mid y_i) + \sum_{i=1}^{m} \log p(y_i \mid y_{i-1})
\end{aligned}$$

HMM

Viterbi + backtracking:

$$s(y) = \sum_{i=1}^{m} s(y_i, y_{i-1}) = \sum_{i=1}^{m} \log p(x_i \mid y_i) + \log p(y_i \mid y_{i-1})$$

$$\pi[j, t] = \max_{t' \in \mathcal{Y}} \underbrace{\log p(x_j \mid t) + \log p(t \mid t')}_{s(y_i, y_{i-1})} + \pi[j-1, t']$$

Table of Contents

Naive Bayes with missing labels

Task:

- Assume data is generated from a Naive Bayes model.
- Observe $\left\{x^{(i)}\right\}_{i=1}^{N}$ without labels.
- Estimate model parameters and the most likely labels.

ID	US	government	gene	lab	label
1	1	1	0	0	?
2	0	1	0	0	?
3	0	0	1	1	?
4	0	1	1	1	?
5	1	1	0	0	?

A chicken and egg problem

If we know the model parameters, we can predict labels easily.

If we know the labels, we can estiamte the model parameters easily.

Idea: start with guesses of labels, then iteratively refine it.

ID	US	government	gene	lab	label
1	1	1	0	0	
2	0	1	0	0	
3	0	0	1	1	
4	0	1	1	1	
5	1	1	0	0	

	US	government	gene	lab
$p(\cdot \mid 0)$				
$p(\cdot \mid 1)$				

$$p(y = 0) = \qquad , p(y = 1) =$$

Iteration 0

Randomly label the data, then estimate parameters given the pseudolabels.

ID	US	government	gene	lab	label
1	1	1	0	0	0
2	0	1	0	0	0
3	0	0	1	1	0
4	0	1	1	1	1
5	1	1	0	0	1

random

	US	government	gene	lab
$p(\cdot \mid 0)$	$1/3$	$2/3$	$1/3$	$1/3$
$p(\cdot \mid 1)$	$1/2$	1	$1/2$	$1/2$

$$p(y = 0) = 3/5, \quad p(y = 1) = 2/5$$

Iteration 1

Given parameters from the last iteration, update the pseudolabels.

ID	US	government	gene	lab	label $y=0$	label $y=1$
1	1	1	0	0	2/5	3/5
2	0	1	0	0		
3	0	0	1	1		
4	0	1	1	1		
5	1	1	0	0		

soft counts

$p(y=0 \mid x_1)$

$\propto p(x_1 \mid y=0)\, p(y=0)$

$= p(US \mid y=0)$

$\times p(gov \mid y=0)$

$\times p(y=0)$

$p(y=1 \mid x_1)$

	US	government	gene	lab
$p(\cdot \mid 0)$	1/3	2/3	1/3	1/3
$p(\cdot \mid 1)$	1/2	1	1/2	1/2

$$p(y=0) = 3/5, \quad p(y=1) = 2/5$$

Algorithm: EM for NB

1. Initialization: $\theta \leftarrow$ random parameters
2. Repeat until convergence:
   (i) Inference:
   $$q(y \mid x^{(i)}) = p(y \mid x^{(i)}; \theta)$$
   soft counts.

   (ii) Update parameters:
   $$\theta_{w|y} = \frac{\sum_{i=1}^{N} q(y \mid x^{(i)}) \mathbb{I}\left[w \text{ in } x^{i}\right]}{\sum_{i=1}^{N} q(y \mid x^{(i)})}$$

- With fully observed data, $q(y \mid x^{(i)}) = 1$ if $y^{(i)} = y$.
- Similar to the MLE solution except that we're using "soft counts".
- What is the algorithm optimizing?

Objective: maximize marginal likelihood

Likelihood: $L(\theta; \mathcal{D}) = \prod_{x \in \mathcal{D}} p(x; \theta)$

Marginal likelihood: $L(\theta; \mathcal{D}) = \prod_{x \in \mathcal{D}} \sum_{z \in \mathcal{Z}} p(x, z; \theta)$

- Introducing latent variables allows us to better model the true generative process
- Marginalize over the (discrete) latent variable $z \in \mathcal{Z}$ (e.g. missing labels)

Maximum marginal log-likelihood estimator:

$$\hat{\theta} = \arg\max_{\theta \in \Theta} \sum_{x \in \mathcal{D}} \log \sum_{z \in \mathcal{Z}} p(x, z; \theta)$$

$\Sigma \Sigma \log p(\cdot\cdot)$

Goal: maximize $\log p(x; \theta)$
Challenge: in general not concave, hard to optimize

Intuition

Problem: marginal log-likelihood is hard to optimize (only observing the words)

Observation: **complete data log-likelihood** is easy to optimize (observing both words and tags)

$$\max_{\theta} \log p(x, z; \theta)$$

Idea: guess a distribution of the latent variables $q(z)$ (soft tags)

Maximize the *expected* complete data log-likelihood:

$$\max_{\theta} \sum_{z \in \mathcal{Z}} q(z) \log p(x, z; \theta)$$

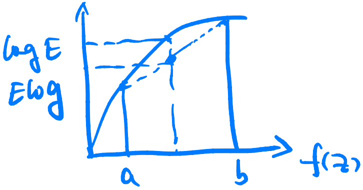
Lower bound of the marginal log-likelihood

marginal log-L

f(z)

a

$$
\begin{aligned}
\log p(x;\theta) &= \log \sum_{z\in\mathcal{Z}} p(x,z;\theta) \\
&= \log \sum_{z\in\mathcal{Z}} q(z) \boxed{\frac{p(x,z;\theta)}{q(z)}} \qquad = \log \mathbb{E}_z\left[f(z)\right] \\
&\geq \boxed{\sum_{z\in\mathcal{Z}} q(z) \log \frac{p(x,z;\theta)}{q(z)}} \qquad = \mathbb{E}_z\left[\log f(z)\right] \qquad \text{Jensen's inequality} \\
&\stackrel{\text{def}}{=} \mathcal{L}(q,\theta)
\end{aligned}
$$



- **Evidence**: $\log p(x;\theta)$
- **Evidence lower bound (ELBO)**: $\mathcal{L}(q,\theta)$
- q: chosen to be a family of tractable distributions
- Idea: Can we maximize the lowerbound instead?

Kullback-Leibler Divergence

- Let $p(x)$ and $q(x)$ be probability mass functions (PMFs) on $\mathcal{X}$.
- How can we measure how "different" p and q are?

- The **Kullback-Leibler** or **"KL" Divergence** is defined by

$$\mathrm{KL}(p\|q) = \sum_{x\in\mathcal{X}} p(x)\log\frac{p(x)}{q(x)}.$$

KL(q||p)

(Assumes $q(x) = 0$ implies $p(x) = 0$.)

- Can also write this as

$$\mathrm{KL}(p\|q) = \mathbb{E}_{x\sim p}\log\frac{p(x)}{q(x)}.$$

Gibbs Inequality ($\mathrm{KL}\left(p \| q\right) \geq 0$ and $\mathrm{KL}\left(p \| q\right) = 0$)

Theorem (Gibbs Inequality)

Let $p(x)$ and $q(x)$ be PMFs on $\mathcal{X}$. Then

$$KL\left(p \| q\right) \geq 0,$$

with equality iff $p(x) = q(x)$ for all $x \in \mathcal{X}$.

- KL divergence measures the "distance" between distributions.
- Note:
  - KL divergence **not a metric**.
  - KL divergence is **not symmetric**.

Gibbs Inequality: Proof

$$
\begin{aligned}
\mathrm{KL}\left(p \| q\right) &= \mathbb{E}_p\left[-\log\left(\frac{q(x)}{p(x)}\right)\right] \\
&\geq -\log\left[\mathbb{E}_p\left(\frac{q(x)}{p(x)}\right)\right] \qquad \text{(Jensen's)} \\
&= -\log\left[\sum_{\{x|p(x)>0\}} p(x)\frac{q(x)}{p(x)}\right] \\
&= -\log\left[\sum_{x\in\mathcal{X}} q(x)\right] \\
&= -\log 1 = 0.
\end{aligned}
$$

- Since $-\log$ is strictly convex, we have strict equality iff $q(x)/p(x)$ is a constant, which implies $q = p$.

Justification for maximizing ELBO

$$\mathcal{L}(q,\theta) \stackrel{\text{def}}{=} \sum_{z\in\mathcal{Z}} q(z) \log \frac{p(x,z;\theta)}{q(z)}$$

$$= \sum_{z\in\mathcal{Z}} q(z) \log \frac{p(z \mid x;\theta)p(x;\theta)}{q(z)}$$

$$= -\sum_{z\in\mathcal{Z}} q(z) \log \frac{q(z)}{p(z \mid x;\theta)} + \underbrace{\sum_{z\in\mathcal{Z}} q(z)}_{=1} \log p(x;\theta)$$

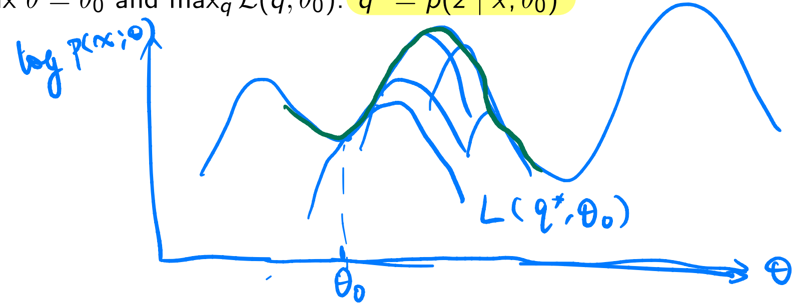
$$= -\mathrm{KL}\left(q(z) \| p(z \mid x;\theta)\right) + \underbrace{\log p(x;\theta)}_{\text{evidence}}$$

- **KL divergence**: measures "distance" between two distributions (not symmetric!)
- $\mathrm{KL}\left(q \| p\right) \geq 0$ with equality iff $q(z) = p(z \mid x)$.
- ELBO $=$ evidence - KL $\leq$ evidence ($\mathrm{KL} \geq 0$)

Justification for maximizing ELBO

$$\mathcal{L}(q, \theta) = -\mathrm{KL}\left(q(z) \| p(z \mid x; \theta)\right) + \log p(x; \theta)$$

Fix $\theta = \theta_0$ and $\max_q \mathcal{L}(q, \theta_0)$: $q^* = p(z \mid x; \theta_0)$



Let θ^*, q^* be the global optimzer of $\mathcal{L}(q, \theta)$, then θ^* is the global optimizer of $\log p(x; \theta)$.

Summary

Latent variable models: clustering, latent structure, missing lables etc.

Parameter estimation: maximum marginal log-likelihood

Challenge: directly maximize the **evidence** $\log p(x;\theta)$ is hard

Solution: maximize the **evidence lower bound**:

$$\text{ELBO} = \mathcal{L}(q,\theta) = -\text{KL}\left(q(z) \| p(z \mid x;\theta)\right) + \log p(x;\theta)$$

Why does it work?

$$\begin{aligned} q^*(z) &= p(z \mid x;\theta) \quad \forall \theta \in \Theta \\ \mathcal{L}(q^*,\theta^*) &= \max_\theta \log p(x;\theta) \end{aligned}$$

EM algorithm

"*Coordinate ascent*" on $\mathcal{L}(q, \theta)$

1. Random initialization: $\theta^{\text{old}} \leftarrow \theta_0$
2. Repeat until convergence

   (i) $q(z) \leftarrow \arg\max_q \mathcal{L}(q, \theta^{\text{old}})$

$$\textbf{Expectation} \text{ (the E-step):} \quad q^*(z) = p(z \mid x; \theta^{\text{old}})$$

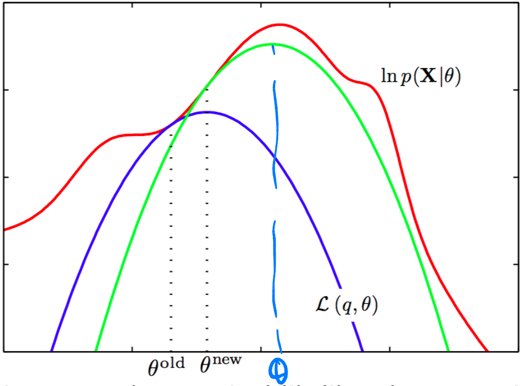
$$\text{ELBO} = \mathcal{L}(q^*, \theta^{\text{old}}) = J(\theta) = \sum_{z \in \mathcal{Z}} q^*(z) \log \frac{p(x, z; \theta)}{q^*(z)}$$

   (ii) $\theta^{\text{new}} \leftarrow \arg\max_\theta \mathcal{L}(q^*, \theta)$

$$\textbf{Maximization} \text{ (the M-step):} \quad \theta^{\text{new}} \leftarrow \arg\max_\theta J(\theta)$$

EM puts no constraint on q in the E-step and assumes the M-step is easy. In general, both steps can be hard.

Monotonically increasing likelihood



HW3: prove that EM increases the marginal likelihood monotonically

$$\log p(x;\theta^{\text{new}}) \geq \log p(x;\theta^{\text{old}}) \ .$$

Does EM converge to a global maximum?

EM for multinomial naive Bayes

Setting: $x = (x_1, \dots, x_m) \in \mathcal{V}^m, z \in \{1, \dots, K\}, \mathcal{D} = \left\{x^{(i)}\right\}_{i=1}^N$

E-step:

$$q^*(z) = p(z \mid x; \theta^{\text{old}}) = \frac{\prod_{i=1}^m p(x_i \mid z; \theta^{\text{old}}) p(z; \theta^{\text{old}})}{\sum_{z' \in \mathcal{Z}} \prod_{i=1}^m p(x_i \mid z'; \theta^{\text{old}}) p(z'; \theta^{\text{old}})}$$

$$J(\theta) = \sum_{x \in \mathcal{D}} \sum_{z \in \mathcal{Z}} q_x^*(z) \log p(x, z; \theta) = \sum_{x \in \mathcal{D}} \sum_{z \in \mathcal{Z}} q_x^*(z) \log \prod_{i=1}^m p(x_i \mid z; \theta) p(z; \theta)$$

M-step:

$$\max_\theta \sum_{x \in \mathcal{D}} \sum_{z \in \mathcal{Z}} q_x^*(z) \left(\sum_{w \in \mathcal{V}} \log \theta_{w|z}^{\text{count}(w|x)} + \log \theta_z \right)$$

$$\text{s.t.} \quad \sum_{w \in \mathcal{V}} \theta_{w|z} = 1 \quad \forall w \in \mathcal{V}, \quad \sum_{z \in \mathcal{Z}} \theta_z = 1 \,,$$

where $\text{count}(w \mid x) \stackrel{\text{def}}{=} \#$ occurrence of w in x

EM for multinomial naive Bayes

M-step has closed-form solution:

$$\theta_z = \frac{\sum_{x \in \mathcal{D}} q_x^*(z)}{\sum_{z \in \mathcal{Z}} \sum_{x \in \mathcal{D}} \underbrace{q_x^*(z)}_{\text{soft label count}}}$$

$$\theta_{w|z} = \frac{\sum_{x \in \mathcal{D}} q_x^*(z) \text{count}(w \mid x)}{\sum_{w \in \mathcal{V}} \sum_{x \in \mathcal{D}} \underbrace{q_x^*(z) \text{count}(w \mid x)}_{\text{soft word count}}}$$

Similar to the MLE solution except that we're using soft counts.

Summary

Expectation maximization (EM) algorithm: maximizing ELBO $\mathcal{L}(q, \theta)$ by coordinate ascent

E-step: Compute the expected complete data log-likelihood $J(\theta)$ using $q^*(z) = p(z \mid x; \theta^{\text{old}})$

M-step: Maximize $J(\theta)$ to obtain θ^{new}

Assumptions: E-step and M-step are easy to compute

Properties: Monotonically improve the likelihood and converge to a stationary point

Table of Contents

HMM recap

Setting:

- Hidden states $z_i \in \mathcal{Y}$ (e.g. POS tags)
- Observations $x_i \in \mathcal{X}$ (e.g. words)

$$p(x_{1:m}, y_{1:m}) = \prod_{i=1}^{m} \underbrace{p(x_i \mid y_i)}_{\text{emission probability}} \prod_{i=1}^{m} \underbrace{p(y_i \mid y_{i-1})}_{\text{transition probability}}$$

Parameters:

- Transition probabilities: $p(y_i = t \mid y_{i-1} = t') = \theta_{t|t'}$
- Emission probabilities: $p(x_i = w \mid y_i = t) = \gamma_{w|t}$
- $y_0 = *, y_m = \texttt{STOP}$

Task: estimate parameters given *incomplete* observations

E-step for HMM

E-step:

$$
\begin{aligned}
q^*(z) &= p(z \mid x; \theta, \gamma) \\
\mathcal{L}(q^*, \theta, \gamma) &= \sum_{x \in \mathcal{D}} \underbrace{\sum_{z \in \mathcal{Z}} q_x^*(z) \log p(x, z; \theta, \gamma)}_{\text{expected complete log-likelihood}} \\
&= \sum_{x \in \mathcal{D}} \sum_{z \in \mathcal{Z}} q_x^*(z) \log \underbrace{\prod_{i=1}^{m} p(x_i \mid z_i) p(z_i \mid z_{i-1})}_{\text{HMM}} \\
&= \sum_{x \in \mathcal{D}} \sum_{z \in \mathcal{Z}} q_x^*(z) \sum_{i=1}^{m} \left(\log \underbrace{p(x_i \mid z_i; \gamma)}_{\gamma_{x_i \mid z_i}} + \log \underbrace{p(z_i \mid z_{i-1}; \theta)}_{\theta_{z_i \mid z_{i-1}}} \right)
\end{aligned}
$$

M-step for HMM

M-step (similar to the NB solution):

$$\max_{\theta,\gamma} \mathcal{L}(q^*, \theta, \gamma) = \sum_{x \in \mathcal{D}} \sum_{z \in \mathcal{Z}} q_x^*(z) \sum_{i=1}^{m} \left(\log \gamma_{x_i|z_i} + \log \theta_{z_i|z_{i-1}} \right)$$

Emission probabilities:

$$\gamma_{w|t} = \frac{\sum_{x \in \mathcal{D}} \sum_{z \in \mathcal{Z}} q_x^*(z) \text{count}(w, t \mid x, z)}{\sum_{w' \in \mathcal{X}} \sum_{x \in \mathcal{D}} \sum_{z \in \mathcal{Z}} q_x^*(z) \text{count}(w', t \mid x, z)}$$

$$\text{count}(w, t \mid x, z) \stackrel{\text{def}}{=} \# \text{ word-tag pairs } (w, t) \text{ in } (x, z)$$

Transition probabilities:

$$\theta_{t|t'} = \frac{\sum_{x \in \mathcal{D}} \sum_{z \in \mathcal{Z}} q_x^*(z) \text{count}(t' \to t \mid z)}{\sum_{a \in \mathcal{Y}} \sum_{x \in \mathcal{D}} \sum_{z \in \mathcal{Z}} q_x^*(z) \text{count}(t' \to a \mid z)}$$

$$\text{count}(t' \to t \mid z) \stackrel{\text{def}}{=} \# \text{ tag bigrams } (t', t) \text{ in } z$$

M-step for HMM

Challenge: $\sum_{z \in \mathcal{Y}^m} q^*_x(z) \text{count}(w, t \mid x, z)$

$$\begin{tikzcd}
\text{DT} \arrow[r] \arrow[d] & \text{NN} \arrow[r] \arrow[d] & \text{VBD} \arrow[r] \arrow[d] & \text{IN} \arrow[r] \arrow[d] & \text{DT} \arrow[r] \arrow[d] & \text{NN} \arrow[d] \\
\text{the} & \text{fox} & \text{jumped} & \text{over} & \text{the} & \text{dog}
\end{tikzcd}$$

Group sequences where $z_i = t$:

$$\sum_{z \in \mathcal{Y}^m} q^*_x(z) \text{count}(w, t \mid x, z) = \sum_{i=1}^{m} \mu_x(z_i = t) \mathbb{I}[x_i = w]$$

$$\mu_x(z_i = t) = \sum_{\{z \in \mathcal{Y}^m \mid z_i = t\}} q^*_x(z)$$

M-step for HMM

Challenge: $\sum_{z \in \mathcal{Y}^m} q_x^*(z) \mathrm{count}(t' \to t \mid z)$

$$\begin{tikzcd}
\mathrm{DT} \arrow[r] \arrow[d] & \mathrm{NN} \arrow[r] \arrow[d] & \mathrm{VBD} \arrow[r] \arrow[d] & \mathrm{IN} \arrow[r] \arrow[d] & \mathrm{DT} \arrow[r] \arrow[d] & \mathrm{NN} \arrow[d] \\
\text{the} & \text{fox} & \text{jumped} & \text{over} & \text{the} & \text{dog}
\end{tikzcd}$$

Group sequences where $z_i = t, z_{i-1} = t'$:

$$\sum_{z \in \mathcal{Y}^m} q_x^*(z) \mathrm{count}(t' \to t \mid z) = \sum_{i=1}^{m} \mu_x(z_i = t, z_{i-1} = t')$$

$$\mu_x(z_i = t, z_{i-1} = t') = \sum_{\{z \in \mathcal{Y}^m \mid z_i = t, z_{i-1} = t'\}} q_x^*(z)$$

Compute tag marginals

$\mu_x(z_i = t)$: probability of the i-th tag being t given observed words x

$$\mu_x(z_i = t) = \sum_{z:z_i=t} q^*_x(z) \propto \sum_{z:z_i=t} \prod_{j=1}^{m} \underbrace{q(x_i \mid z_i) q(z_i \mid z_{i-1})}_{\psi(z_i, z_{i-1})}$$

$$= \sum_{z:z_i=t} \prod_{j=1}^{i-1} \psi(z_j, z_{j-1}) \prod_{j=i}^{m} \psi(z_j, z_{j-1})$$

$$= \sum_{t'} \sum_{z:z_i=t, z_{i-1}=t'} \prod_{j=1}^{i-1} \psi(z_j, z_{j-1}) \prod_{j=i}^{m} \psi(z_j, z_{j-1})$$

$$= \sum_{t'} \left(\sum_{\substack{z_{1:i-1} \\ z_{i-1}=t'}} \prod_{j=1}^{i-1} \psi(z_j, z_{j-1}) \right) \psi(t, t') \left(\sum_{\substack{z_{i+1:m} \\ z_i=t}} \prod_{j=i+1}^{m} \psi(z_j, z_{j-1}) \right)$$

$$= \sum_{t'} \alpha[i-1, t] \psi(t, t') \beta[i, t] = \alpha[i, t] \beta[i, t]$$

Compute tag marginals

Forward probabilities: probability of tag sequence prefix ending at $z_i = t$.

$$\alpha[i,t] \stackrel{\text{def}}{=} q(x_1, \dots, x_i, z_i = t)$$

$$\alpha[i,t] = \sum_{t' \in \mathcal{Y}} \alpha[i-1, t'] \psi(t', t)$$

Backward probabilities: probability of tag sequence suffix starting from z_{i+1} give $z_i = t$.

$$\beta[i,t] \stackrel{\text{def}}{=} q(x_{i+1}, \dots, x_m \mid z_i = t)$$

$$\beta[i,t] = \sum_{t' \in \mathcal{Y}} \beta[i+1, t'] \psi(t, t')$$

Compute tag marginals

1. Compute forward and backward probabilities

$$\begin{aligned}
&\alpha[i, t] \quad \forall i \in \{1, \ldots, m\}, t \in \mathcal{Y} \cup \{\texttt{STOP}\} \\
&\beta[i, t] \quad \forall i \in \{m, \ldots, 1\}, t \in \mathcal{Y} \cup \{*\}
\end{aligned}$$

2. Comptute the tag unigram and bigram marginals

$$\begin{aligned}
\mu_x(z_i = t) &\stackrel{\text{def}}{=} q(z_i = t \mid x) \\
&= \frac{\alpha[i, t]\beta[i, t]}{q(x)} = \frac{\alpha[i, t]\beta[i, t]}{\alpha[m, \texttt{STOP}]} \\
\mu_x(z_{i-1} = t', z_i = t) &\stackrel{\text{def}}{=} q(z_{i-1} = t', z_i = t \mid x) \\
&= \frac{\alpha[i-1, t']\psi(t', t)\beta[i, t]}{q(x)}
\end{aligned}$$

In practice, compute in the *log space*.

Updated parameters

Emission probabilities:

$$
\begin{aligned}
\gamma_{w|t} &= \frac{\sum_{x \in \mathcal{D}} \sum_{z \in \mathcal{Z}} q_x^*(z)\text{count}(w, t \mid x, z)}{\sum_{w' \in \mathcal{X}} \sum_{x \in \mathcal{D}} \sum_{z \in \mathcal{Z}} q_x^*(z)\text{count}(w', t \mid x, z)} \\
&= \frac{\sum_{x \in \mathcal{D}} \sum_{i=1}^{m} \mu_x(z_i = t)\mathbb{I}\left[x_i = w\right]}{\sum_{w' \in \mathcal{X}} \sum_{x \in \mathcal{D}} \sum_{i=1}^{m} \mu_x(z_i = t)\mathbb{I}\left[x_i = w'\right]}
\end{aligned}
$$

Transition probabilities:

$$
\begin{aligned}
\theta_{t|t'} &= \frac{\sum_{x \in \mathcal{D}} \sum_{z \in \mathcal{Z}} q_x^*(z)\text{count}(t' \to t \mid z)}{\sum_{a \in \mathcal{Y}} \sum_{x \in \mathcal{D}} \sum_{z \in \mathcal{Z}} q_x^*(z)\text{count}(t' \to a \mid z)} \\
&= \frac{\sum_{x \in \mathcal{D}} \sum_{i=1}^{m} \mu_x(z_{i-1} = t', z_i = t)}{\sum_{a \in \mathcal{Y}} \sum_{x \in \mathcal{D}} \sum_{i=1}^{m} \mu_x(z_{i-1} = t', z_i = a)}
\end{aligned}
$$

Summary

EM for HMM:

1. Randomly initialize the emission and transition probabilities
2. Repeat until convergence
   (i) Compute forward and backward probabilities
   (ii) Update the emission and transition probabilities using expected counts
3. If the solution is bad, re-run EM with a different random seed.

General EM:

- One example of variational methods (use a tractable q to approximate p)
- May need approximation in both the E-step and the M-step