Inference and Representation

David Sontag

New York University

Lecture 6, Oct. 14, 2015

Approximate marginal inference

- Given the joint $p(x_1, \ldots, x_n)$ represented as a graphical model, how do we perform **marginal inference**, e.g. to compute $p(x_1 \mid e)$?
- We showed in Lecture 4 that doing this exactly is NP-hard
- Nearly all *approximate inference* algorithms are either:
  1. **Monte-carlo methods (e.g., Gibbs sampling, likelihood reweighting, MCMC)**
  2. Variational algorithms (e.g., mean-field, loopy belief propagation)

Generating samples from a Bayesian network

Algorithm 12.1 Forward Sampling in a Bayesian network

Procedure Forward-Sample (
  $\mathcal{B}$ // Bayesian network over $\mathcal{X}$
)

  Let $X_1, \ldots, X_n$ be a topological ordering of $\mathcal{X}$
  for $i = 1, \ldots, n$
    $\boldsymbol{u}_i \leftarrow \boldsymbol{x}\langle \mathrm{Pa}_{X_i} \rangle$ // Assignment to Pa_{X_i} in $x_1, \ldots, x_{i-1}$
    Sample x_i from $P(X_i \mid \boldsymbol{u}_i)$
  return $(x_1, \ldots, x_n)$

(Koller & Friedman, *Probabilistic Graphical Models*, MIT Press 2009)

Monte-Carlo algorithms

- Given a joint distribution $p(x_1, \ldots, x_n)$, how do we compute marginals?

$$
\begin{aligned}
p[X_1 = x_1] &= E_{\mathbf{x} \sim p}[f(\mathbf{x})], \text{ where } f(\mathbf{x}) = 1[X_1 = x_1] \\
&= \sum_{\mathbf{x}} p(\mathbf{x}) f(\mathbf{x}).
\end{aligned}
$$

- Rather than explicitly enumerating *all* assignments, consider the following Monte-Carlo estimate of the expectation:

$$
\begin{aligned}
\mathbf{x}^1 &\sim p(\mathbf{x}) \\
\mathbf{x}^2 &\sim p(\mathbf{x}) \\
&\vdots \\
\mathbf{x}^M &\sim p(\mathbf{x})
\end{aligned}
$$

- Then, our *estimate* is $\hat{E}_p[f(x)] = \frac{1}{M} \sum_{m=1}^{M} f(\mathbf{x}^m)$. How good is it?

Monte-Carlo algorithms

- Let $\mathcal{D} = \{\mathbf{x}^1, \ldots, \mathbf{x}^M\}$. Since $\mathcal{D}$ was drawn randomly from $p(\mathbf{x})$, the estimate is itself a random variable
- The estimate is *unbiased* because

$$
\begin{aligned}
E_{\mathbf{x}^1,\ldots,\mathbf{x}^M \sim p(\mathbf{x})}\left[\hat{E}[f(\mathbf{x})]\right] &= E_{\mathbf{x}^1,\ldots,\mathbf{x}^M \sim p(\mathbf{x})}\left[\frac{1}{M}\sum_{m=1}^{M} f(\mathbf{x}^m)\right] \\
&= \frac{1}{M}\sum_{m=1}^{M} E_{\mathbf{x}^m \sim p(\mathbf{x})}\left[f(\mathbf{x}^m)\right] \\
&= E_{\mathbf{x} \sim p(\mathbf{x})}[f(\mathbf{x})].
\end{aligned}
$$

- How quickly does the estimate converge to the true expectation?

Law of large numbers

- There are two general results we can use, depending on whether we care about additive or multiplicative error
- **Hoeffding bound** says that:

$$\mathrm{Pr}_{\mathcal{D}\sim p(\mathbf{x})}\Big[E_p[f(\mathbf{x})] - \epsilon \le \hat{E}_{\mathcal{D}}[f(\mathbf{x})] \le E_p[f(\mathbf{x})] + \epsilon\Big] \ge 1 - 2e^{-2M\epsilon^2}$$

- **Chernoff bound** says that (assuming $f(\mathbf{x}) \in [0,1]$):

$$\mathrm{Pr}_{\mathcal{D}\sim p(\mathbf{x})}\Big[E_p[f(\mathbf{x})](1-\epsilon) \le \hat{E}_{\mathcal{D}}[f(\mathbf{x})] \le E_p[f(\mathbf{x})](1+\epsilon)\Big] \ge 1 - 2e^{\frac{-M\epsilon^2}{3} E_p[f(\mathbf{x})]}$$

- Estimating *single-variable* marginals for a BN is easy: just forward sample!
- What about computing *conditional* queries such as $p(\mathbf{X} = \mathbf{x} \mid \mathbf{E} = \mathbf{e})$?
- Computing denominator of $p(\mathbf{X} = \mathbf{x}, \mathbf{E} = \mathbf{e})/p(\mathbf{E} = \mathbf{e})$ needs $\Omega(1/p(\mathbf{E} = \mathbf{e}))$ samples, by Chernoff bound.

"Normalized" Importance Sampling

- If we could instead directly sample from $p(\mathbf{X} \mid \mathbf{E} = \mathbf{e})$, we would be in business – but this is hard!
- For the same reason, sampling from an undirected graphical model $p(\mathbf{x}) = \frac{1}{Z} \prod_{c \in C} \phi_c(\mathbf{x}_c)$ – even without evidence – is hard, because we don't know Z
- Suppose we instead had a simpler-to-sample-from distribution $q(\mathbf{x})$, called the "proposal distribution"
- Let $\tilde{p}(\mathbf{x})$ be an unnormalized version of the distribution, e.g.

$$\begin{aligned} \tilde{p}(\mathbf{x}) &= p(\mathbf{x}, E = \mathbf{e}) \quad \text{(BN with evidence)} \\ \tilde{p}(\mathbf{x}) &= \prod_{c \in C} \phi_c(\mathbf{x}_c) \quad \text{(MRF)} \end{aligned}$$

Note that we can efficiently *evaluate* $\tilde{p}(\mathbf{x})$ for any $\mathbf{x}$

"Normalized" Importance Sampling

- Consider the following estimate (now using $\mathbf{x}^1, \ldots \mathbf{x}^M \sim q(\mathbf{x})$):

$$\hat{E}_{\mathcal{D}}[f(x)] = \frac{\frac{1}{M}\sum_{m=1}^{M} f(\mathbf{x}^m)\tilde{w}(\mathbf{x}^m)}{\frac{1}{M}\sum_{m=1}^{M} \tilde{w}(\mathbf{x}^m)}, \quad \text{where } \tilde{w}(\mathbf{x}) = \frac{\tilde{p}(\mathbf{x})}{q(\mathbf{x})}$$

- This is *not* an unbiased estimate! E.g., for $M = 1$, we have

$$\begin{aligned} E_{\mathbf{x}_1 \sim q(\mathbf{x})}\left[\hat{E}_{\mathcal{D}}[f(x)]\right] = E_{\mathbf{x}_1 \sim q(\mathbf{x})}\left[\frac{f(\mathbf{x}^1)\tilde{w}(\mathbf{x}^1)}{\tilde{w}(\mathbf{x}^1)}\right] &= E_{\mathbf{x} \sim q(\mathbf{x})}[f(\mathbf{x})] \\ &\neq E_{\mathbf{x} \sim p(\mathbf{x})}[f(\mathbf{x})]. \end{aligned}$$

- However, the estimate is asymptotically correct (i.e., as $M \to \infty$)

"Normalized" Importance Sampling

- Consider the following estimate (now using $\mathbf{x}^1, \ldots \mathbf{x}^M \sim q(\mathbf{x})$):

$$\hat{E}_{\mathcal{D}}[f(x)] = \frac{\frac{1}{M}\sum_{m=1}^{M} f(\mathbf{x}^m)\tilde{w}(\mathbf{x}^m)}{\frac{1}{M}\sum_{m=1}^{M} \tilde{w}(\mathbf{x}^m)}, \quad \text{where } \tilde{w}(\mathbf{x}) = \frac{\tilde{p}(\mathbf{x})}{q(\mathbf{x})}$$

- Letting $\tilde{p}(\mathbf{x}) = p(\mathbf{x})Z$, the expectation of the numerator is:

$$\begin{aligned} E_{\mathcal{D}\sim q(\mathbf{x})}\left[\frac{1}{M}\sum_{m=1}^{M} f(\mathbf{x}^m)\tilde{w}(\mathbf{x}^m)\right] &= \frac{1}{M}\sum_{m=1}^{M} E_{\mathbf{x}^m\sim q(\mathbf{x})}[f(\mathbf{x}^m)\tilde{w}(\mathbf{x}^m)] \\ &= \frac{1}{M}\sum_{m=1}^{M}\sum_{\mathbf{x}} q(\mathbf{x})\left[f(\mathbf{x})\frac{\tilde{p}(\mathbf{x})}{q(\mathbf{x})}\right] \\ &= \frac{1}{M}\sum_{m=1}^{M}\sum_{\mathbf{x}} \tilde{p}(\mathbf{x})f(\mathbf{x}) = ZE_p[f(\mathbf{x})]. \end{aligned}$$

"Normalized" Importance Sampling

- Consider the following estimate (now using $\mathbf{x}^1, \ldots \mathbf{x}^M \sim q(\mathbf{x})$):

$$\hat{E}_{\mathcal{D}}[f(x)] = \frac{\frac{1}{M}\sum_{m=1}^{M} f(\mathbf{x}^m)\tilde{w}(\mathbf{x}^m)}{\frac{1}{M}\sum_{m=1}^{M} \tilde{w}(\mathbf{x}^m)}, \quad \text{where } \tilde{w}(\mathbf{x}) = \frac{\tilde{p}(\mathbf{x})}{q(\mathbf{x})}$$

- Letting $\tilde{p}(\mathbf{x}) = p(\mathbf{x})Z$, the expectation of the numerator is $ZE_p[f(\mathbf{x})]$.
- The expectation of the denominator is Z!

$$\begin{aligned} E_{\mathcal{D}\sim q(\mathbf{x})}\left[\frac{1}{M}\sum_{m=1}^{M} \tilde{w}(\mathbf{x}^m)\right] &= \frac{1}{M}\sum_{m=1}^{M} E_{\mathbf{x}^m\sim q(\mathbf{x})}[\tilde{w}(\mathbf{x}^m)] \\ &= \frac{1}{M}\sum_{m=1}^{M}\sum_{\mathbf{x}} q(\mathbf{x})\left[\frac{\tilde{p}(\mathbf{x})}{q(\mathbf{x})}\right] \\ &= \frac{1}{M}\sum_{m=1}^{M}\sum_{\mathbf{x}} \tilde{p}(\mathbf{x}) = Z. \end{aligned}$$

Likelihood weighting

- What should we use for $q(\mathbf{x})$? For a Bayesian network, we can sample sample from the latent variables, keeping the evidence fixed

Algorithm 12.2 Likelihood-weighted particle generation

Procedure LW-Sample (
  $\mathcal{B}$, // Bayesian network over $\mathcal{X}$
  $\boldsymbol{Z} = \boldsymbol{z}$ // Event in the network
)
  Let $X_1, \ldots, X_n$ be a topological ordering of $\mathcal{X}$
  $w \leftarrow 1$
  for $i = 1, \ldots, n$
    $\boldsymbol{u}_i \leftarrow \boldsymbol{x}\langle \mathrm{Pa}_{X_i} \rangle$ // Assignment to Pa_{X_i} in $x_1, \ldots, x_{i-1}$
    if $X_i \notin \boldsymbol{Z}$ **then**
      Sample x_i from $P(X_i \mid \boldsymbol{u}_i)$
    else
      $x_i \leftarrow \boldsymbol{z}\langle X_i \rangle$ // Assignment to X_i in $\boldsymbol{z}$
      $w \leftarrow w \cdot P(x_i \mid \boldsymbol{u}_i)$ // Multiply weight by probability of desired value
  return $(x_1, \ldots, x_n), w$

(Koller & Friedman, *Probabilistic Graphical Models*, MIT Press 2009)

- Corresponds to importance sampling using:

$$q(\mathbf{x}) = \prod_{t \notin \mathbf{E}} p(x_t \mid \mathbf{x}_{pa(t)}) \prod_{t \in \mathbf{E}} 1[x_t = \mathbf{e}_t], \text{ so } \tilde{w}(x) = \frac{\tilde{p}(\mathbf{x})}{q(\mathbf{x})} = \prod_{t \in \mathbf{E}} p(x_t \mid \mathbf{x}_{pa(t)}).$$

Problem Set 4

Problem Set 4 will explore Gibbs sampling for topic models

Latent Dirichlet allocation (LDA)

- **Topic models** are powerful tools for exploring large data sets and for making inferences about the content of documents

Documents

Topics

politics	religion	sports
president	hindu	baseball
obama	judiasm	soccer
washington	ethics	basketball
religion	buddhism	football
...	...	...

- Many applications in information retrieval, document summarization, and classification

New document

Words $w_1, \ldots, w_N$

What is this document about?

weather	.50
finance	.49
sports	.01

Distribution of topics θ

- LDA is one of the simplest and most widely used topic models

Generative model for a document in LDA

1. Sample the document's **topic distribution** θ (aka topic vector)

$$\theta \sim \mathrm{Dirichlet}(\alpha_{1:T})$$

where the $\{\alpha_t\}_{t=1}^T$ are fixed hyperparameters. Thus θ is a distribution over T topics with mean $\theta_t = \alpha_t / \sum_{t'} \alpha_{t'}$

2. For $i = 1$ to N, sample the **topic** z_i of the i'th word

$$z_i | \theta \sim \theta$$

3. ... and then sample the actual **word** w_i from the z_i'th topic

$$w_i | z_i \sim \beta_{z_i}$$

where $\{\beta_t\}_{t=1}^T$ are the *topics* (a fixed collection of distributions on words)

Generative model for a document in LDA

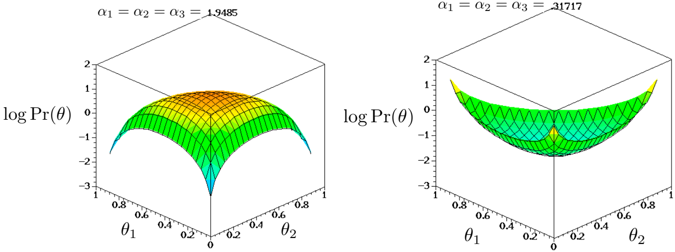
1. Sample the document's **topic distribution** θ (aka topic vector)

$$\theta \sim \mathrm{Dirichlet}(\alpha_{1:T})$$

where the $\{\alpha_t\}_{t=1}^{T}$ are hyperparameters.The Dirichlet density, defined over $\Delta = \{\vec{\theta} \in \mathbb{R}^T : \forall t\ \theta_t \geq 0, \sum_{t=1}^{T} \theta_t = 1\}$, is:

$$p(\theta_1, \ldots, \theta_T) \propto \prod_{t=1}^{T} \theta_t^{\alpha_t - 1}$$

For example, for T=3 ($\theta_3 = 1 - \theta_1 - \theta_2$):



Generative model for a document in LDA

3. … and then sample the actual **word** w_i from the z_i'th topic

$$w_i | z_i \sim \beta_{z_i}$$

where $\{\beta_t\}_{t=1}^{T}$ are the *topics* (a fixed collection of distributions on words)

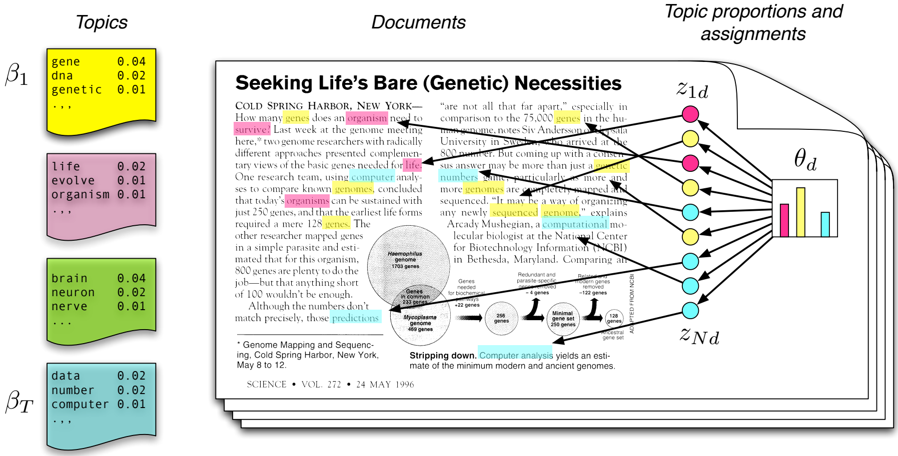
Documents

Topics

politics .0100	religion .0500	sports .0105
president .0095	hindu .0092	baseball .0100
obama .0090	judiasm .0080	soccer .0055
washington .0085	ethics .0075	basketball .0050
religion .0060	buddhism .0016	football .0045
…	…	…

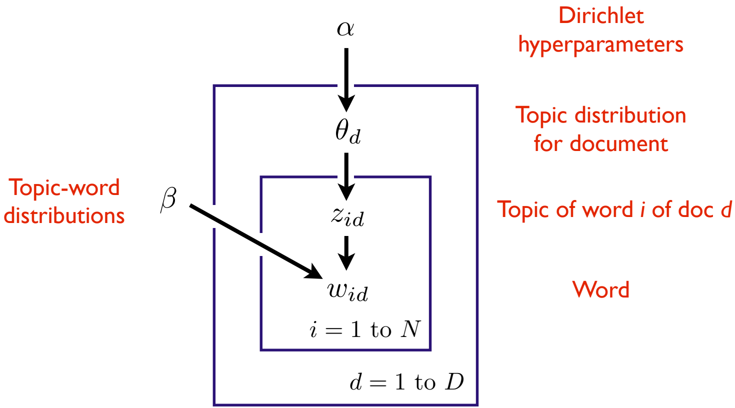
$$\beta_t = \{\, p(w \mid z = t) \,\}$$

Example of using LDA



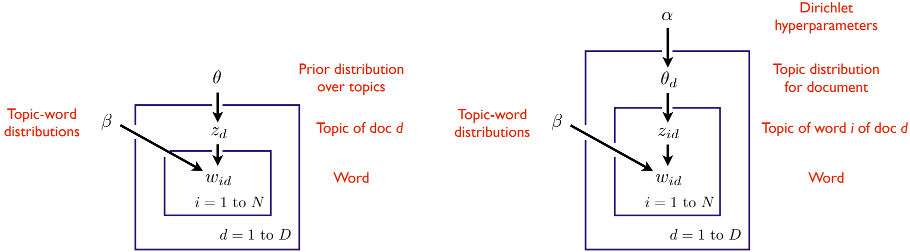
(Blei, *Introduction to Probabilistic Topic Models*, 2011)

"Plate" notation for LDA model



Variables within a plate are replicated in a conditionally independent manner

Comparison of mixture and admixture models



- Model on left is a **mixture model**
  - Called *multinomial* naive Bayes (a word can appear multiple times)
  - Document is generated from a single topic
- Model on right (LDA) is an **admixture model**
  - Document is generated from a distribution over topics