Variational Inference w/Exponential Families

+ Learning Hidden-State CRFs
Reminders

• Homework 4: MCMC
  – Out: Wed, Mar. 24
  – Due: Wed, Apr. 7 at 11:59pm

• Project Midway Milestones:
  – Midway Poster Session:
    Wed, Apr. 14 at 6:30pm – 8:30pm
  – Midway Executive Summary
    Due: Wed, Apr. 14 at 11:59pm
MEAN FIELD VARIATIONAL INFERENCE
Variational Inference

Whiteboard

– Coordinate Ascent Variational Inference (CAVI) Algorithm
  • Connecting CAVI to BP and Gibbs sampling
  • Computing marginals from a trained mean field approximation

– CAVI algorithm derivation
  • Chain rule decomposition of $\log p(x, z)$
  • Decomposing the entropy
  • Decomposing the ELBO
  • Derivatives and closed form solution
CAVI Algorithm

Coordinate Ascent Variational Inference (CAVI)

- here we assume a **mean field** approximation
- application of **coordinate ascent** to maximization of ELBO
- converges to a **local optimum** of the **nonconvex** ELBO objective

```
1: procedure CAVI(p_α)
2:    Let q_θ(z) = \prod_{t=1}^{T} q_t(z_t)  \quad \triangleright Mean field approx.
3:    while ELBO(q_θ) has not converged do
4:        for t ∈ {1, ..., T} do \quad \triangleright For each variable
5:            Set q_t(z_t) ∝ \exp(E_{q_{−t}}[\log p_α(z_t | z_{−t}, x)])
6:                while keeping all \{q_s(·)\}_{s \neq t} fixed
7:        Compute ELBO(q_θ) = E_{q_θ(z)} [\log p_α(x, z)] - E_{q_θ(z)} [\log q_θ(z)]
8:    return q_θ
```
**Variational Inference**

**Whiteboard**

- Computing the CAVI update
  - Multinomial full conditionals
- Example: two variable factor graph
  - Joint distribution
  - Mean Field Variational Inference
  - Gibbs Sampling
EXPONENTIAL FAMILY DISTRIBUTION
Exponential family, a basic building block

- For a numeric random variable $X$
  
  $$p(x \mid \eta) = h(x) \exp \left\{ \eta^T T(x) - A(\eta) \right\}$$
  
  $$= \frac{1}{Z(\eta)} h(x) \exp \{ \eta^T T(x) \}$$

  is an exponential family distribution with natural (canonical) parameter $\eta$

- Function $T(x)$ is a sufficient statistic.
- Function $A(\eta) = \log Z(\eta)$ is the log normalizer.
- Examples: Bernoulli, multinomial, Gaussian, Poisson, gamma,...
Example: Multivariate Gaussian Distribution

- For a continuous vector random variable \( X \in \mathbb{R}^k \):
  \[
p(x|\mu, \Sigma) = \frac{1}{(2\pi)^{k/2} |\Sigma|^{1/2}} \exp\left\{ -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right\}
  = \frac{1}{(2\pi)^{k/2}} \exp\left\{ -\frac{1}{2} \text{tr}(\Sigma^{-1}xx^T) + \mu^T \Sigma^{-1} x - \frac{1}{2} \mu^T \Sigma^{-1} \mu - \log|\Sigma| \right\}
  \]

- Exponential family representation
  \[
  \eta = \left[ \Sigma^{-1} \mu; -\frac{1}{2} \text{vec}(\Sigma^{-1}) \right] = [\eta_1, \text{vec}(\eta_2)], \quad \eta_1 = \Sigma^{-1} \mu \quad \text{and} \quad \eta_2 = -\frac{1}{2} \Sigma^{-1}
  
  T(x) = \left[ x; \text{vec}(xx^T) \right]
  
  A(\eta) = \frac{1}{2} \mu^T \Sigma^{-1} \mu + \log|\Sigma| = -\frac{1}{2} \text{tr}(\eta_2 \eta_1^T) - \frac{1}{2} \log(-2\eta_2)
  
  h(x) = (2\pi)^{-k/2}
  \]

- Note: a \( k \)-dimensional Gaussian is a \((d + d^2)\)-parameter distribution with a \((d + d^2)\)-element vector of sufficient statistics (but because of symmetry and positivity, parameters are constrained and have lower degree of freedom)
Example: Multinomial distribution

- For a binary vector random variable $\mathbf{x} \sim \text{multi}(\mathbf{x} \mid \pi)$,

$$p(x \mid \pi) = \pi_1^{x_1} \pi_2^{x_2} \cdots \pi_K^{x_K} = \exp\left\{ \sum_k x_k \ln \pi_k \right\}$$

$$= \exp\left\{ \sum_{k=1}^{K-1} x_k \ln \pi_k + \left( 1 - \sum_{k=1}^{K-1} x_K \right) \ln \left( 1 - \sum_{k=1}^{K-1} \pi_k \right) \right\}$$

$$= \exp\left\{ \sum_{k=1}^{K-1} x_k \ln \left( \frac{\pi_k}{1 - \sum_{k=1}^{K-1} \pi_k} \right) + \ln \left( 1 - \sum_{k=1}^{K-1} \pi_k \right) \right\}$$

- Exponential family representation

$$\eta = \left[ \ln \left( \frac{\pi_k}{\pi_K} \right); 0 \right]$$

$$T(x) = [x]$$

$$A(\eta) = -\ln \left( 1 - \sum_{k=1}^{K-1} \pi_k \right) = \ln \left( \sum_{k=1}^{K} e^{\eta_k} \right)$$

$$h(x) = 1$$
Examples

- Gaussian:
  \[ \eta = \left[ \Sigma^{-1} \mu; -\frac{1}{2} \text{vec}(\Sigma^{-1}) \right] \]
  \[ T(x) = [x; \text{vec}(xx^T)] \]
  \[ A(\eta) = \frac{1}{2} \mu^T \Sigma^{-1} \mu + \frac{1}{2} \log|\Sigma| \]
  \[ h(x) = (2\pi)^{-k/2} \]
  \[ \Rightarrow \mu_{\text{MLE}} = \frac{1}{N} \sum_n T_1(x_n) = \frac{1}{N} \sum_n x_n \]

- Multinomial:
  \[ \eta = \left[ \ln \left( \frac{\pi_k}{\pi_k} \right); 0 \right] \]
  \[ T(x) = [x] \]
  \[ A(\eta) = -\ln \left( 1 - \sum_{k=1}^{K-1} \pi_k \right) = \ln \left( \sum_{k=1}^K e^{\eta_k} \right) \]
  \[ h(x) = 1 \]
  \[ \Rightarrow \mu_{\text{MLE}} = \frac{1}{N} \sum_n x_n \]

- Poisson:
  \[ \eta = \log \lambda \]
  \[ T(x) = x \]
  \[ A(\eta) = \lambda = e^\eta \]
  \[ h(x) = \frac{1}{x!} \]
  \[ \Rightarrow \mu_{\text{MLE}} = \frac{1}{N} \sum_n x_n \]
Why exponential family?

- Moment generating property

\[
\frac{dA}{d\eta} = \frac{d}{d\eta} \log Z(\eta) = \frac{1}{Z(\eta)} \frac{d}{d\eta} Z(\eta)
\]

\[
= \frac{1}{Z(\eta)} \frac{d}{d\eta} \int h(x) \exp \{\eta^T T(x)\} dx
\]

\[
= \int T(x) \frac{h(x) \exp \{\eta^T T(x)\}}{Z(\eta)} dx
\]

\[
= E[T(x)]
\]

\[
\frac{d^2 A}{d\eta^2} = \int T^2(x) \frac{h(x) \exp \{\eta^T T(x)\}}{Z(\eta)} dx - \int T(x) \frac{h(x) \exp \{\eta^T T(x)\}}{Z(\eta)} dx \frac{1}{Z(\eta)} \frac{d}{d\eta} Z(\eta)
\]

\[
= E[T^2(x)] - E^2[T(x)]
\]

\[
= Var[T(x)]
\]
Moment estimation

- We can easily compute moments of any exponential family distribution by taking the derivatives of the log normalizer $A(\eta)$.
- The $q^{th}$ derivative gives the $q^{th}$ centered moment.

\[
\frac{dA(\eta)}{d\eta} = \text{mean} \\
\frac{d^2 A(\eta)}{d\eta^2} = \text{variance} \\
\ldots
\]

- When the sufficient statistic is a stacked vector, partial derivatives need to be considered.
The moment parameter \( \mu \) can be derived from the natural (canonical) parameter \( \lambda \) since

\[
\frac{dA(\eta)}{d\eta} = E[T(x)] = \mu
\]

\( A(\eta) \) is convex since

\[
\frac{d^2 A(\eta)}{d\eta^2} = Var[T(x)] > 0
\]

Hence we can invert the relationship and infer the canonical parameter from the moment parameter (1-to-1):

\[
\eta = \psi(\mu)
\]

A distribution in the exponential family can be parameterized not only by \( \eta \) – the canonical parameterization, but also by \( \mu \) – the moment parameterization.
MLE for Exponential Family

- For iid data, the log-likelihood is
  \[
  \ell(\eta; D) = \log \prod_n h(x_n) \exp \{ \eta^T T(x_n) - A(\eta) \} 
  \]
  \[
  = \sum_n \log h(x_n) + \left( \eta^T \sum_n T(x_n) \right) - NA(\eta) 
  \]
- Take derivatives and set to zero:
  \[
  \frac{\partial \ell}{\partial \eta} = \sum_n T(x_n) - N \frac{\partial A(\eta)}{\partial \eta} = 0 
  \]
  \[
  \frac{\partial A(\eta)}{\partial \eta} = \frac{1}{N} \sum_n T(x_n) 
  \]
  \[
  \Rightarrow \mu_{MLE} = \frac{1}{N} \sum_n T(x_n) 
  \]
- This amounts to moment matching.
- We can infer the canonical parameters using \( \hat{\eta}_{MLE} = \psi(\mu_{MLE}) \)
Sufficiency

- For $p(x|\theta)$, $\mathcal{T}(x)$ is sufficient for $\theta$ if there is no information in $X$ regarding $\theta$ beyond that in $\mathcal{T}(x)$.
  - We can throw away $X$ for the purpose of inference w.r.t. $\theta$.

- Bayesian view

- Frequentist view

- The Neyman factorization theorem
  - $\mathcal{T}(x)$ is sufficient for $\theta$ if

\[
p(x,T(x),\theta) = \psi_1(T(x),\theta)\psi_2(x,T(x))
\]

\[
\Rightarrow p(x|\theta) = g(T(x),\theta)h(x,T(x))
\]
EXPONENTIAL FAMILY AND MEAN FIELD VARIATIONAL INF.
CAVI Algorithm

Coordinate Ascent Variational Inference (CAVI)

- here we assume a mean field approximation
- application of coordinate ascent to maximization of ELBO
- converges to a local optimum of the nonconvex ELBO objective

```
1: procedure CAVI(\(p_\alpha\))
2:    Let \(q_\theta(z) = \prod_{t=1}^{T} q_t(z_t)\) \(\triangleright\) Mean field approx.
3:    while ELBO(\(q_\theta\)) has not converged do
4:        for \(t \in \{1, \ldots, T\}\) do \(\triangleright\) For each variable
5:            Set \(q_t(z_t) \propto \exp(E_{q_{-t}}[\log p_\alpha(z_t | z_{-t}, x)])\)
6:            while keeping all \(\{q_s(\cdot)\}_{s \neq t}\) fixed
7:        Compute ELBO(\(q_\theta\)) = \(E_{q_\theta(z)}[\log p_\alpha(x, z)] - E_{q_\theta(z)}[\log q_\theta(z)]\)
8:    return \(q_\theta\)
```
Optimizing the ELBO in Mean Field Variational Inference

Notes:

- This coordinate ascent procedure convergences to a local maximum.
- The coordinate ascent update for $q(z_j)$ only depends on the other, fixed approximations $q(z_k)$, $k \neq j$.
- While this determines the optimal $q(z_j)$, we haven’t yet specified the form (i.e. what specific distribution family) of $q$ we aim to use, only the factorization.
- Depending on what form we use, the coordinate update $q^*(z_j)$ might not be easy to work with (and might not be in the same form as $q(z_j)$ …).
  - But in many cases it is!
  - And we will specify what forms yield good coordinate updates.
Optimizing the ELBO in Mean Field Variational Inference

Simple Example: multinomial conditionals

- Suppose we have chosen a model whose conditional distribution is a multinomial, i.e.
  \[ p(z_j | z_{-j}, x) = \pi(z_{-j}, x) \]

- Then the optimal (coordinate update for) \( q(z_j) \) is:
  \[ q^*(z_j) \propto \exp\{\mathbb{E}[\log \pi(z_{-j}, x)]\} \]

- Which is also a multinomial, and is easy to compute. So choosing a multinomial family of approximations for each latent variable gives closed form coordinate ascent updates.
Quick Recap

Quick recap on what we’ve covered:

- We defined a family of approximations called “mean field” approximations, in which there are no dependencies between latent variables (and also a generalized version of this).

- We decomposed the ELBO into a nice form under mean field assumptions.

- We derived coordinate ascent updates to iteratively optimize each local variational approximation under mean field assumptions.

- Next, we will discuss specific forms for the local variational approximations in which we can easily compute (closed-form) coordinate ascent updates.
Is there a general form for models in which the coordinate updates in mean field variational inference are easy to compute and lead to closed-form updates?

Yes: the answer is exponential family conditionals.

I.e. models with conditional densities that are in an exponential family, i.e. of the form:

\[ p(z_j|z_{-j}, x) = h(z_j) \exp \left\{ \eta(z_{-j}, x)^\top t(z_j) - a(\eta(z_{-j}, x)) \right\} \]

where \( h, \eta, t, \) and \( a \) are functions that parameterize the exponential family.

Different choices of these parameters lead to many popular densities (normal, gamma, exponential, Bernoulli, Dirichlet, categorical, beta, Poisson, geometric, etc.).
Exponential Family Conditionals

- We call these “exponential-family-conditional” models.
  - Also known as “conditionally conjugate models”.

- Many popular models fall into this category, including:
  - Bayesian mixtures of exponential family models with conjugate priors.
  - Hierarchical hidden Markov models.
  - Kalman filter models and switching Kalman filters.
  - Mixed-membership models of exponential families.
  - Factorial mixtures / hidden Markov models of exponential families.
  - Bayesian linear regression.
  - Any model containing only conjugate pairs and multinomials.

- Some popular models do not fall into this category, including:
  - Bayesian logistic regression and other nonconjugate Bayesian generalized linear models.
  - Correlated topic model, dynamic topic model.
  - Discrete choice models.
  - Nonlinear matrix factorization models.
Exponential Family Conditionals

- We can derive a general formula for the coordinate ascent update for all exponential-family-conditional models.

- First, we will choose the form of our local variational approximation \( q(z_j) \) to be the same as the conditional distribution (i.e. in an exponential family).

- When we perform our coordinate ascent update, we will see that the update yields an optimal \( q(z_j) \) in the same family.

- Recall from above that we derived the coordinate ascent updates for optimizing the ELBO (under the mean field assumption) as:

\[
q^*(z_j) \propto \exp \left\{ \mathbb{E}_{q_{-j}} \left[ \log p(z_j|z_{-j}, x) \right] \right\}
\]
Exponential Family Conditionals

Coordinate ascent updates for exponential-family-conditional models (under the mean field approximation):

- The log of the conditional:

\[
\log p(z_j|z_{-j}, x) = \log h(z_j) + \eta(z_{-j}, x)\top t(z_j) - a(\eta(z_{-j}, x))
\]

- The expectation of this with respect to \( q(z_{-j}) \) is:

\[
\mathbb{E}_{q_{-j}} [\log p(z_j|z_{-j}, x)] = \log h(z_j) + \mathbb{E}_{q_{-j}} [\eta(z_{-j}, x)]\top t(z_j) - \mathbb{E}_{q_{-j}} [a(\eta(z_{-j}, x))]
\]

- The last term does not depend on \( q(z_j) \), so we have the update:

\[
q^*(z_j) \propto h(z_j) \exp \left\{ \mathbb{E}_{q_{-j}} [\eta(z_{-j}, x)]\top t(z_j) \right\}
\]

- So the optimal \( q(z_j) \) is in the same exponential family as the conditional.
Exponential Family Conditionals

Writing this update in terms of variational parameters $\nu$.

- Give each latent variable a variational parameter $\nu_j$. Under the mean field assumption, we can write the full approximation as:

$$q(z_{1:m}|\nu) = \prod_{j=1}^{m} q(z_j|\nu_j)$$

where each local variational approximation has an exponential family form.

- Then the coordinate ascent algorithm updates each variational parameter, in turn, as:

$$\nu_j^* = \mathbb{E}_{q_{-j}} [\eta(z_{-j}, x)]$$
Quick Recap

Quick recap on what we’ve covered:

- We found a family of models (exponential-family-conditional models) in which we have closed form coordinate ascent updates to optimize the ELBO.
  - And we gave a number of examples (and non-examples) of these models.

- We gave an explicit form for the coordinate ascent update for these exponential-family-conditional models.
  - And also looked at the update in terms of the local variational parameters.
MAP INFERENCE AND VARIATIONAL INFERENC
Suppose: We want a family $Q$ such that the variational inference solution:

$$
\hat{q}(z) = \text{argmin}_{q \in Q} \text{KL}(q \parallel p)
$$

gives back a distribution that is a point mass on the MAP inference solution:

$$
\hat{q}(z) = \begin{cases} 
\hat{z} = \text{argmax}_{z \in \mathcal{Z}(x)} p(z \mid x) & \text{w/prob. 1.0} \\
\text{any other } z \in \mathcal{Z}(x) & \text{w/prob. 0.0}
\end{cases}
$$

Question: What is $Q$?

Answer:
VARIATIONAL INFERENCE RESULTS
Variational Inference & Nonconvexity

• ELBO is a non-convex objective function
• Below shows 10 random initializations of CAVI for Gaussian Mixture Model
• Parameters with higher ELBO are closer to true posterior

Figure 2: Different initializations may lead CAVI to find different local optima of the ELBO.

Figure from Blei, Kucukelbir, and McAuliffe (2018)
Variational Bayesian LDA

- Explicit Variational Inference
Variational Bayesian LDA

• Explicit Variational Inference

Standard VB inference upper bounds the negative log marginal likelihood \(- \log p(x|\alpha, \beta)\) using the variational free energy:

\[
- \log p(x|\alpha, \beta) \leq \tilde{F}(\tilde{q}(z, \theta, \phi)) = E_{\tilde{q}}[- \log p(x, z, \phi, \theta|\alpha, \beta)] - H(\tilde{q}(z, \theta, \phi))
\]  

with \(\tilde{q}(z, \theta, \phi)\) an approximate posterior, \(H(\tilde{q}(z, \theta, \phi)) = E_{\tilde{q}}[- \log \tilde{q}(z, \theta, \phi)]\) the variational entropy, and \(\tilde{q}(z, \theta, \phi)\) assumed to be fully factorized:

\[
\tilde{q}(z, \theta, \phi) = \prod_{ij} \tilde{q}(z_{ij}|\tilde{\gamma}_{ij}) \prod_{j} \tilde{q}(\theta_j|\tilde{\alpha}_j) \prod_{k} \tilde{q}(\phi_k|\tilde{\beta}_k)
\]  

\(\tilde{q}(z_{ij}|\tilde{\gamma}_{ij})\) is multinomial with parameters \(\tilde{\gamma}_{ij}\) and \(\tilde{q}(\theta_j|\tilde{\alpha}_j)\), \(\tilde{q}(\phi_k|\tilde{\beta}_k)\) are Dirichlet with parameters \(\tilde{\alpha}_j\) and \(\tilde{\beta}_k\) respectively. Optimizing \(\tilde{F}(\tilde{q})\) with respect to the variational parameters gives us a set of updates guaranteed to improve \(\tilde{F}(\tilde{q})\) at each iteration and converges to a local minimum:

\[
\tilde{\alpha}_{jk} = \alpha + \sum_i \tilde{\gamma}_{ijk}
\]

\[
\tilde{\beta}_{kw} = \beta + \sum_{ij} \mathbf{1}(x_{ij}=w) \tilde{\gamma}_{ijk}
\]

\[
\tilde{\gamma}_{ijk} \propto \exp \left( \Psi(\tilde{\alpha}_{jk}) + \Psi(\tilde{\beta}_{kx_{ij}}) - \Psi(\sum_w \tilde{\beta}_{kw}) \right)
\]

where \(\Psi(y) = \frac{\partial \log \Gamma(y)}{\partial y}\) is the digamma function and \(\mathbf{1}\) is the indicator function.

Text from Teh et al (2006)
Collapsed Variational Bayesian LDA

• Collapsed Variational Inference

\[
\begin{align*}
\alpha & \\
\theta_m & \rightarrow \alpha \\
\phi_k & \rightarrow \beta \\
x_{mn} & \rightarrow \theta_m \\
N_m & \rightarrow \theta_m \\
M & \\
\end{align*}
\]
Collapsed Variational Bayesian LDA

- **First row**: test set per word log probabilities as functions of numbers of iterations for VB, CVB and Gibbs.
- **Second row**: histograms of final test set per word log probabilities across 50 random initializations.

Slide from Teh et al (2007)
Online Variational Bayes for LDA

Figure 1: Top: Perplexity on held-out Wikipedia documents as a function of number of documents analyzed, i.e., the number of E steps. Online VB run on 3.3 million unique Wikipedia articles is compared with online VB run on 98,000 Wikipedia articles and with the batch algorithm run on the same 98,000 articles. The online algorithms converge much faster than the batch algorithm does.

Bottom: Evolution of a topic about business as online LDA sees more and more documents.

A central research problem for topic modeling is to efficiently fit models to larger corpora [4, 5]. To this end, we develop an online variational Bayes algorithm for latent Dirichlet allocation (LDA), one of the simplest topic models and one on which many others are based. Our algorithm is based on online stochastic optimization, which has been shown to produce good parameter estimates dramatically faster than batch algorithms on large datasets [6]. Online LDA handily analyzes massive collections of documents and, moreover, online LDA need not locally store or collect the documents—each can arrive in a stream and be discarded after one look.

In the subsequent sections, we derive online LDA and show that it converges to a stationary point of the variational objective function. We study the performance of online LDA in several ways, including by fitting a topic model to 3.3M articles from Wikipedia without looking at the same article twice. We show that online LDA finds topic models as good as or better than those found with batch VB, and in a fraction of the time (see figure 1). Online variational Bayes is a practical new method for estimating the posterior of complex hierarchical Bayesian models.

2 Online variational Bayes for latent Dirichlet allocation

Latent Dirichlet Allocation (LDA) [7] is a Bayesian probabilistic model of text documents. It assumes a collection of $K$ "topics." Each topic defines a multinomial distribution over the vocabulary and is assumed to have been drawn from a Dirichlet, $\phi_k \sim \text{Dirichlet}(\alpha)$. Given the topics, LDA assumes the following generative process for each document $d$. First, draw a distribution over topics $\phi_d \sim \text{Dirichlet}(\alpha)$. Then, for each word $i$ in the document, draw a topic index $z_{di} \sim \phi_d$ and draw the observed word $w_{di} \sim \phi(z_{di})$.

For simplicity, we assume symmetric priors on $\phi$ and $\alpha$, but this assumption is easy to relax [8]. Note that if we sum over the topic assignments $z$, then we get $p(w_{di} | \phi_d, \alpha) = \sum_k \phi(z_{di} = k)$. This leads to the "multinomial PCA" interpretation of LDA; we can think of LDA as a probabilistic factorization of the matrix of word counts $n_{dw}$ (where $n_{dw}$ is the number of times word $w$ appears in document $d$) into a matrix of topic weights $\phi$ and a dictionary of topics [9]. Our work can thus...
Online Variational Bayes for LDA

Algorithm 1 Batch variational Bayes for LDA

Initialize $\lambda$ randomly.

while relative improvement in $\mathcal{L}(w, \phi, \gamma, \lambda) > 0.00001$ do

E step:

for $d = 1$ to $D$ do

Initialize $\gamma_{dk} = 1$. (The constant 1 is arbitrary.)

repeat

Set $\phi_{d wk} \propto \exp\{\mathbb{E}_q[\log \theta_{dk}] + \mathbb{E}_q[\log \beta_{kw}]\}$

Set $\gamma_{dk} = \alpha + \sum_w \phi_{d wk} n_{dw}$

until $\frac{1}{K} \sum_k |\text{change in } \gamma_{dk}| < 0.00001$

end for

M step:

Set $\lambda_{kw} = \eta + \sum_d n_{dw} \phi_{d wk}$

end while

Algorithm 2 Online variational Bayes for LDA

Define $\rho_t \triangleq (\tau_0 + t)^{-\kappa}$

Initialize $\lambda$ randomly.

for $t = 0$ to $\infty$ do

E step:

Initialize $\gamma_{tk} = 1$. (The constant 1 is arbitrary.)

repeat

Set $\phi_{tkw} \propto \exp\{\mathbb{E}_q[\log \theta_{tk}] + \mathbb{E}_q[\log \beta_{kw}]\}$

Set $\gamma_{tk} = \alpha + \sum_w \phi_{tkw} n_{tw}$

until $\frac{1}{K} \sum_k |\text{change in } \gamma_{tk}| < 0.00001$

M step:

Compute $\tilde{\lambda}_{kw} = \eta + D n_{tw} \phi_{tkw}$

Set $\lambda = (1 - \rho_t) \lambda + \rho_t \tilde{\lambda}$

end for

Figure 2: Held-out perplexity obtained on the Nature (left) and Wikipedia (right) corpora as a function of CPU time. For moderately large mini-batch sizes, online LDA finds solutions as good as those that the batch LDA finds, but with much less computation. When fit to a 10,000-document subset of the training corpus batch LDA’s speed improves, but its performance suffers.
Fully-Connected CRF

Model

\[ p(x|i) = \frac{1}{Z(i)} \exp(-E(x)) \]

\[ E(x) = \sum_i \psi_u(x_i) + \sum_{i<j} \psi_p(x_i, x_j), \]

Inference

- Can do MCMC, but slow
- Instead use Variational Inference
- Then filter some variables for speed up

Follow-up Work (combine with CNN)

Published as a conference paper at ICLR 2015

SEMANTIC IMAGE SEGMENTATION WITH DEEP CONVOLUTIONAL NETS AND FULLY CONNECTED CRFS

Liang-Chieh Chen
Univ. of California, Los Angeles
icchen@cs.ucla.edu

George Papandreou *
Google Inc.
gpapan@google.com

Iasonas Kokkinos
CentraleSupélec and INRIA
iasonas.kokkinos@epfl.ch

Kevin Murphy
Google Inc.
kmurphy@gmail.com

Alan L. Yuille
Univ. of California, Los Angeles
yuille@stat.ucla.edu

ABSTRACT

Deep Convolutional Neural Networks (DCNNs) have recently shown state of the art performance in high level vision tasks, such as image classification and object detection. This work brings together methods from DCNNs and probabilistic graphical models for addressing the task of pixel-level classification (also called “semantic image segmentation”). We show that responses at the final layer of DCNNs are not sufficiently localized for accurate object segmentation. This is due to the very invariance properties that make DCNNs good for high level tasks. We overcome this poor localization property of deep networks by combining the responses at the final DCNN layer with a fully connected Conditional Random Field (CRF). Qualitatively, our “DeepLab” system is able to localize segment boundaries at a level of accuracy which is beyond previous methods. Quantitatively, our method sets the new state-of-art at the PASCAL VOC-2012 semantic image segmentation task, reaching 71.6% IOU accuracy in the test set. We show how these results can be obtained efficiently: Careful network re-purposing and a novel application of the ‘hole’ algorithm from the wavelet community allow dense computation of neural net responses at 8 frames per second on a modern GPU.

Figures from Krähenbühl & Koltun (2011)
Fully-Connected CRF

Model

\[ p(x|i) = \frac{1}{Z(i)} \exp(-E(x)) \]

\[ E(x) = \sum_{i} \psi_u(x_i) + \sum_{i<j} \psi_p(x_i, x_j), \]

Inference

- Can do MCMC, but slow
- Instead use Variational Inference
- Then filter some variables for speed up

Results

Figure 1: Pixel-level classification with a fully connected CRF. (a) Input image from the MSRC-21 dataset. (b) The response of unary classifiers used by our models. (c) Classification produced by the Robust \( P^n \) CRF [9]. (d) Classification produced by MCMC inference [17] in a fully connected pixel-level CRF model; the algorithm was run for 36 hours and only partially converged for the bottom image. (e) Classification produced by our inference algorithm in the fully connected model in 0.2 seconds.

Figure 2: Convergence analysis. (a) KL-divergence of the mean field approximation during successive iterations of the inference algorithm, averaged across 94 images from the MSRC-21 dataset. (b) Visualization of convergence on distributions for two class labels over an image from the dataset.
Joint Parsing and Alignment with Weakly Synchronized Grammars

Figure 2: An example of a Chinese-English sentence pair with parses, word alignments, and a subset of the full optimal ITG derivation, including one totally unsynchronized bispan ($b_4$), one partially synchronized bispan ($b_7$), and and fully synchronized bispan ($b_8$). The inset provides some examples of active synchronization features (see Section 4.3) on these bispans. On this example, the monolingual English parser erroneously attached the lower PP to the VP headed by established, and the non-syntactic ITG word aligner misaligned 等 to such instead of to etc. Our joint model corrected both of these mistakes because it was rewarded for the synchronization of the two NPs joined by $b_8$. 

Figures from Burkett et al. (2010)
Joint Parsing and Alignment with Weakly Synchronized Grammars

Figures from Burkett & Klein (ACL 2013 tutorial)

Table 1: Parsing results. Our joint model has the highest reported $F_1$ for English-Chinese bilingual parsing.

<table>
<thead>
<tr>
<th></th>
<th>Ch $F_1$</th>
<th>Eng $F_1$</th>
<th>Tot $F_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monolingual</td>
<td>83.6</td>
<td>81.2</td>
<td>82.5</td>
</tr>
<tr>
<td>Reranker</td>
<td>86.0</td>
<td>83.8</td>
<td>84.9</td>
</tr>
<tr>
<td>Joint</td>
<td>85.7</td>
<td>84.5</td>
<td>85.1</td>
</tr>
</tbody>
</table>

Table 2: Word alignment results. Our joint model has the highest reported $F_1$ for English-Chinese word alignment.

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>AER</th>
<th>$F_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM</td>
<td>86.0</td>
<td>58.4</td>
<td>30.0</td>
<td>69.5</td>
</tr>
<tr>
<td>ITG</td>
<td>86.8</td>
<td>73.4</td>
<td>20.2</td>
<td>79.5</td>
</tr>
<tr>
<td>Joint</td>
<td>85.5</td>
<td>84.6</td>
<td>14.9</td>
<td>85.0</td>
</tr>
</tbody>
</table>
HIDDEN STATE CRFS
Case Study: Object Recognition

Data consists of images $x$ and labels $y$. 

pigeon  
leopard  
rhinoceros  
llama
Case Study: Object Recognition

Data consists of images $x$ and labels $y$.

- Preprocess data into “patches”
- Posit a latent labeling $z$ describing the object’s parts (e.g. head, leg, tail, torso, grass)
- Define graphical model with these latent variables in mind
- $z$ is not observed at train or test time
Case Study: Object Recognition

Data consists of images $x$ and labels $y$.

- Preprocess data into “patches”
- Posit a latent labeling $z$ describing the object’s parts (e.g. head, leg, tail, torso, grass)
- Define graphical model with these latent variables in mind
- $z$ is not observed at train or test time
Case Study: Object Recognition

Data consists of images $x$ and labels $y$.

- Preprocess data into “patches”
- Posit a latent labeling $z$ describing the object’s parts (e.g. head, leg, tail, torso, grass)
- Define graphical model with these latent variables in mind
- $z$ is not observed at train or test time
Hidden-state CRFs

Data: \[ \mathcal{D} = \{\mathbf{x}^{(n)}, \mathbf{y}^{(n)}\}_{n=1}^N \]

Joint model: \[ p_\theta(\mathbf{y}, \mathbf{z} \mid \mathbf{x}) = \frac{1}{Z(\mathbf{x}, \theta)} \prod_\alpha \psi_\alpha(\mathbf{y}_\alpha, \mathbf{z}_\alpha, \mathbf{x}) \]

Marginalized model: \[ p_\theta(\mathbf{y} \mid \mathbf{x}) = \sum_\mathbf{z} p_\theta(\mathbf{y}, \mathbf{z} \mid \mathbf{x}) \]
Hidden-state CRFs

Data: \( \mathcal{D} = \{x^{(n)}, y^{(n)}\}_{n=1}^{N} \)

Joint model: \( p_{\theta}(y, z | x) = \frac{1}{Z(x, \theta)} \prod_{\alpha} \psi_{\alpha}(y_\alpha, z_\alpha, x) \)

Marginalized model: \( p_{\theta}(y | x) = \sum_{z} p_{\theta}(y, z | x) \)

We can train using gradient based methods:
(the values \( x \) are omitted below for clarity)

\[
\frac{d\ell(\theta | \mathcal{D})}{d\theta} = \sum_{n=1}^{N} \left( E_{z \sim p_{\theta}(\cdot | y^{(n)})}[f_j(y^{(n)}, z)] - E_{y, z \sim p_{\theta}(\cdot, \cdot)}[f_j(y, z)] \right)
\]

\[
= \sum_{n=1}^{N} \sum_{\alpha} \left( \sum_{z_\alpha} p_{\theta}(z_\alpha | y^{(n)}) f_{\alpha,j}(y^{(n)}_\alpha, z_\alpha) - \sum_{y_\alpha, z_\alpha} p_{\theta}(y_\alpha, z_\alpha) f_{\alpha,j}(y_\alpha, z_\alpha) \right)
\]

Inference on clamped factor graph

Inference on full factor graph