10-708 Probabilistic Graphical Models

Machine Learning Department
School of Computer Science
Carnegie Mellon University

Variational Inference w/Exponential Families
+
Learning Hidden-State CRFs

Matt Gormley
Lecture 18
Apr. 2, 2021

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

procedure CAVI(p_α)
    Let $q_\theta(\mathbf{z}) = \prod_{t=1}^{T} q_t(z_t)$ ▷ Mean field approx.
    while ELBO(q_θ) has not converged **do**
        for $t \in \{1, \ldots, T\}$ **do** ▷ For each variable
            Set $q_t(z_t) \propto \exp(E_{q_{\neg t}}[\log p_\alpha(z_t \mid z_{\neg t}, x)])$
            while keeping all $\{q_s(\cdot)\}_{s \neq t}$ fixed
        Compute ELBO(q_θ) $= E_{q_\theta(\mathbf{z})}[\log p_\alpha(\mathbf{x}, \mathbf{z})] - E_{q_\theta(\mathbf{z})}[\log q_\theta(\mathbf{z})]$
    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

$$p(x \mid \eta) = h(x)\exp\left\{\eta^T T(x) - A(\eta)\right\}$$
$$= \frac{1}{Z(\eta)} h(x)\exp\left\{\eta^T T(x)\right\}$$

is an **exponential family distribution** with natural (canonical) parameter η

- 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 $X \in \mathbf{R}^k$:

$$p(x|\mu,\Sigma) = \frac{1}{(2\pi)^{k/2}|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right\}$$

Moment parameter

$$= \frac{1}{(2\pi)^{k/2}} \exp\left\{-\tfrac{1}{2}\operatorname{tr}\left(\Sigma^{-1}xx^T\right) + \mu^T\Sigma^{-1}x - \tfrac{1}{2}\mu^T\Sigma^{-1}\mu - \log|\Sigma|\right\}$$

- Exponential family representation

Natural parameter

$$\eta = \left[\Sigma^{-1}\mu; -\tfrac{1}{2}\operatorname{vec}\left(\Sigma^{-1}\right)\right] = \left[\eta_1, \operatorname{vec}(\eta_2)\right],\ \eta_1 = \Sigma^{-1}\mu \text{ and } \eta_2 = -\tfrac{1}{2}\Sigma^{-1}$$

$$T(x) = \left[x; \operatorname{vec}\left(xx^T\right)\right]$$

$$A(\eta) = \tfrac{1}{2}\mu^T\Sigma^{-1}\mu + \log|\Sigma| = -\tfrac{1}{2}\operatorname{tr}(\eta_2\eta_1\eta_1^T) - \tfrac{1}{2}\log\left(-2\eta_2\right)$$

$$h(x) = (2\pi)^{-k/2}$$

  - 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 $x \sim \text{multi}(x \mid \pi)$,

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

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

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

- Exponential family representation

$$\eta = \left[\ln\left(\pi_k / \pi_K\right); 0\right]$$

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

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

$$h(x) = 1$$

Examples

- Gaussian:

$$\eta = \left[\Sigma^{-1}\mu; -\tfrac{1}{2}\operatorname{vec}\left(\Sigma^{-1}\right)\right]$$
$$T(x) = \left[x; \operatorname{vec}\left(xx^T\right)\right]$$
$$A(\eta) = \tfrac{1}{2}\mu^T\Sigma^{-1}\mu + \tfrac{1}{2}\log|\Sigma|$$
$$h(x) = (2\pi)^{-k/2}$$

$$\Rightarrow \mu_{MLE} = \frac{1}{N}\sum_n T_1(x_n) = \frac{1}{N}\sum_n x_n$$

- Multinomial:

$$\eta = \left[\ln\left(\pi_k / \pi_K\right); 0\right]$$
$$T(x) = [x]$$
$$A(\eta) = -\ln\left(1 - \sum_{k=1}^{K-1}\pi_k\right) = \ln\left(\sum_{k=1}^{K} e^{\eta_k}\right)$$
$$h(x) = 1$$

$$\Rightarrow \mu_{MLE} = \frac{1}{N}\sum_n x_n$$

- Poisson:

$$\eta = \log\lambda$$
$$T(x) = x$$
$$A(\eta) = \lambda = e^{\eta}$$
$$h(x) = \frac{1}{x!}$$

$$\Rightarrow \mu_{MLE} = \frac{1}{N}\sum_n x_n$$

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.

$$\frac{dA(\eta)}{d\eta} = \text{mean}$$

$$\frac{d^2 A(\eta)}{d\eta^2} = \text{variance}$$

$$\cdots$$

- When the sufficient statistic is a stacked vector, partial derivatives need to be considered.

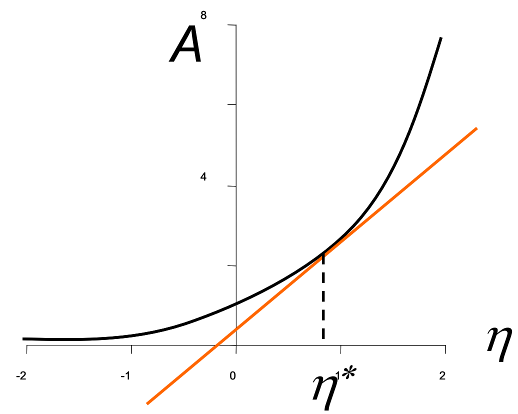
Moment vs canonical parameters

- The moment parameter μ can be derived from the natural (canonical) parameter

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

- $A(\eta)$ 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):

$$\eta \stackrel{\text{def}}{=} \psi(\mu)$$

  - A distribution in the exponential family can be parameterized not only by η – the canonical parameterization, but also by μ – the moment parameterization.

MLE for Exponential Family

- For *iid* data, the log-likelihood is

$$\ell\,(\eta; D) = \log \prod_n h(x_n) \exp\left\{\eta^T T(x_n) - A(\eta)\right\}$$

$$= \sum_n \log h(x_n) + \left(\eta^T \sum_n T(x_n)\right) - NA(\eta)$$

- Take derivatives and set to zero:

$$\frac{\partial \ell}{\partial \eta} = \sum_n T(x_n) - N\frac{\partial A(\eta)}{\partial \eta} = 0$$

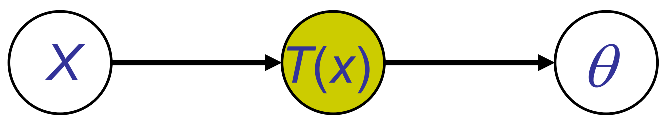
$$\Rightarrow \quad \frac{\partial A(\eta)}{\partial \eta} = \frac{1}{N}\sum_n T(x_n)$$

$$\hat{\mu}_{MLE} = \frac{1}{N}\sum_n T(x_n)$$

- This amounts to **moment matching**.
- We can infer the canonical parameters using $\hat{\eta}_{MLE} = \psi(\hat{\mu}_{MLE})$

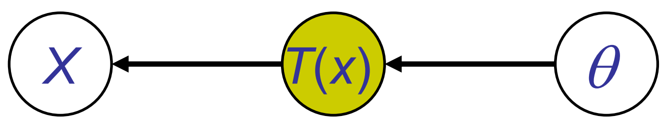
Sufficiency

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



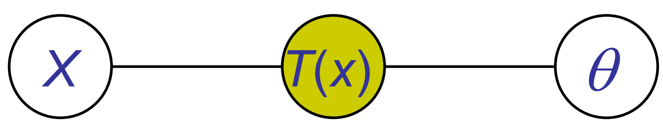
$$p(\theta \mid T(x), x) = p(\theta \mid T(x))$$

  - Frequentist view



$$p(x \mid T(x), \theta) = p(x \mid T(x))$$

  - The Neyman factorization theorem
    - $T(x)$ is *sufficient* for θ if



$$p(x, T(x), \theta) = \psi_1(T(x), \theta)\psi_2(x, T(x))$$

$$\Rightarrow p(x \mid \theta) = g(T(x), \theta)h(x, T(x))$$

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

procedure CAVI(p_α)
    Let $q_\theta(\mathbf{z}) = \prod_{t=1}^{T} q_t(z_t)$ ▷ Mean field approx.
    while ELBO(q_θ) has not converged **do**
        for $t \in \{1, \ldots, T\}$ **do** ▷ For each variable
            Set $q_t(z_t) \propto \exp(E_{q_{\neg t}}[\log p_\alpha(z_t \mid z_{\neg t}, x)])$
            while keeping all $\{q_s(\cdot)\}_{s \neq t}$ fixed
        Compute ELBO(q_θ) $= E_{q_\theta(\mathbf{z})}[\log p_\alpha(\mathbf{x}, \mathbf{z})] - E_{q_\theta(\mathbf{z})}[\log q_\theta(\mathbf{z})]$
    return q_θ

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.

Exponential Family Conditionals

- 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, η, t, and a are functions that parameterize the exponential family.

- Different choices of these parameters lead to many popular densities (normal, gamma, exponential, Bernouilli, 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:

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

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

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

- 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 ν.

- Give each latent variable a variational parameter ν_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:

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

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 INFERENCE

MAP Inference as Variational Inference

Suppose: We want a family $\mathcal{Q}$ such that the variational inference solution:

$$\hat{q}(\mathbf{z}) = \operatorname*{argmin}_{q \in \mathcal{Q}} \mathsf{KL}(q \parallel p)$$

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

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

Question: What is $\mathcal{Q}$?

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

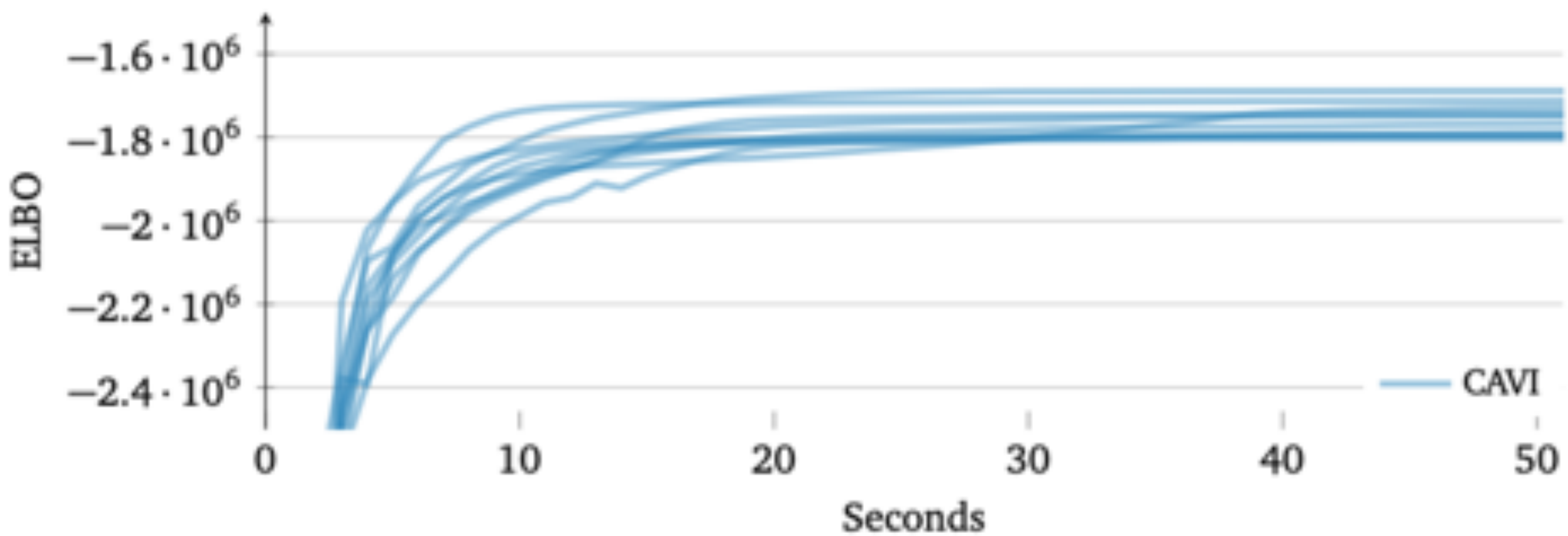
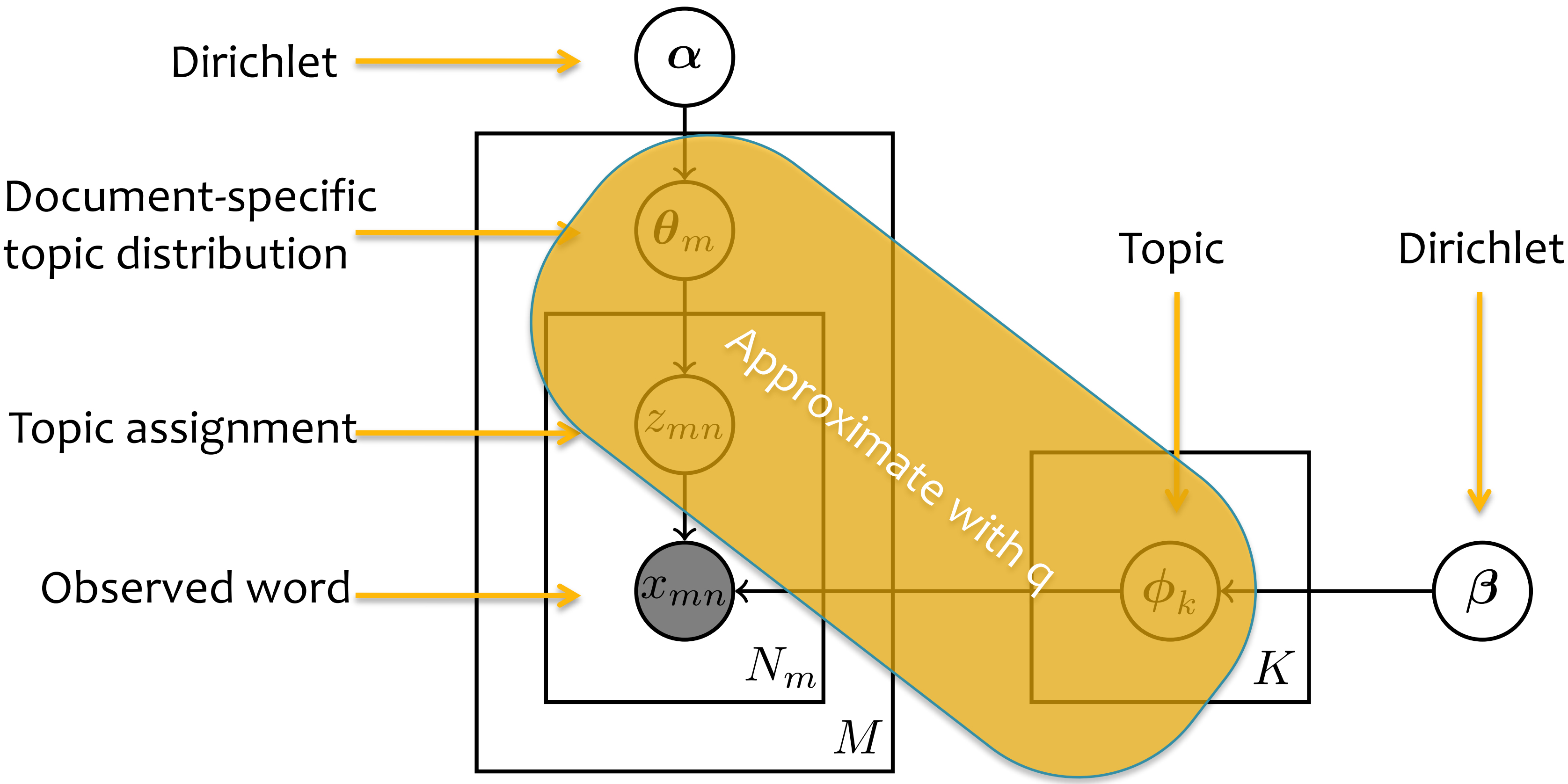


Figure 2: Different initializations may lead CAVI to find different local optima of the ELBO.

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 $-\log p(\mathbf{x}|\alpha,\beta)$ using the variational free energy:

$$-\log p(\mathbf{x}|\alpha,\beta) \leq \widetilde{\mathcal{F}}(\tilde{q}(\mathbf{z},\boldsymbol{\theta},\boldsymbol{\phi})) = E_{\tilde{q}}[-\log p(\mathbf{x},\mathbf{z},\boldsymbol{\phi},\boldsymbol{\theta}|\alpha,\beta)] - \mathcal{H}(\tilde{q}(\mathbf{z},\boldsymbol{\theta},\boldsymbol{\phi})) \tag{2}$$

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

$$\tilde{q}(\mathbf{z},\boldsymbol{\theta},\boldsymbol{\phi}) = \prod_{ij} \tilde{q}(z_{ij}|\tilde{\gamma}_{ij}) \prod_{j} \tilde{q}(\theta_j|\tilde{\alpha}_j) \prod_{k} \tilde{q}(\phi_k|\tilde{\beta}_k) \tag{3}$$

$\tilde{q}(z_{ij}|\tilde{\gamma}_{ij})$ 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 $\widetilde{\mathcal{F}}(\tilde{q})$ with respect to the variational parameters gives us a set of updates guaranteed to improve $\widetilde{\mathcal{F}}(\tilde{q})$ at each iteration and converges to a local minimum:

$$\tilde{\alpha}_{jk} = \alpha + \textstyle\sum_i \tilde{\gamma}_{ijk} \tag{4}$$

$$\tilde{\beta}_{kw} = \beta + \textstyle\sum_{ij} \mathbf{1}(x_{ij}=w)\tilde{\gamma}_{ijk} \tag{5}$$

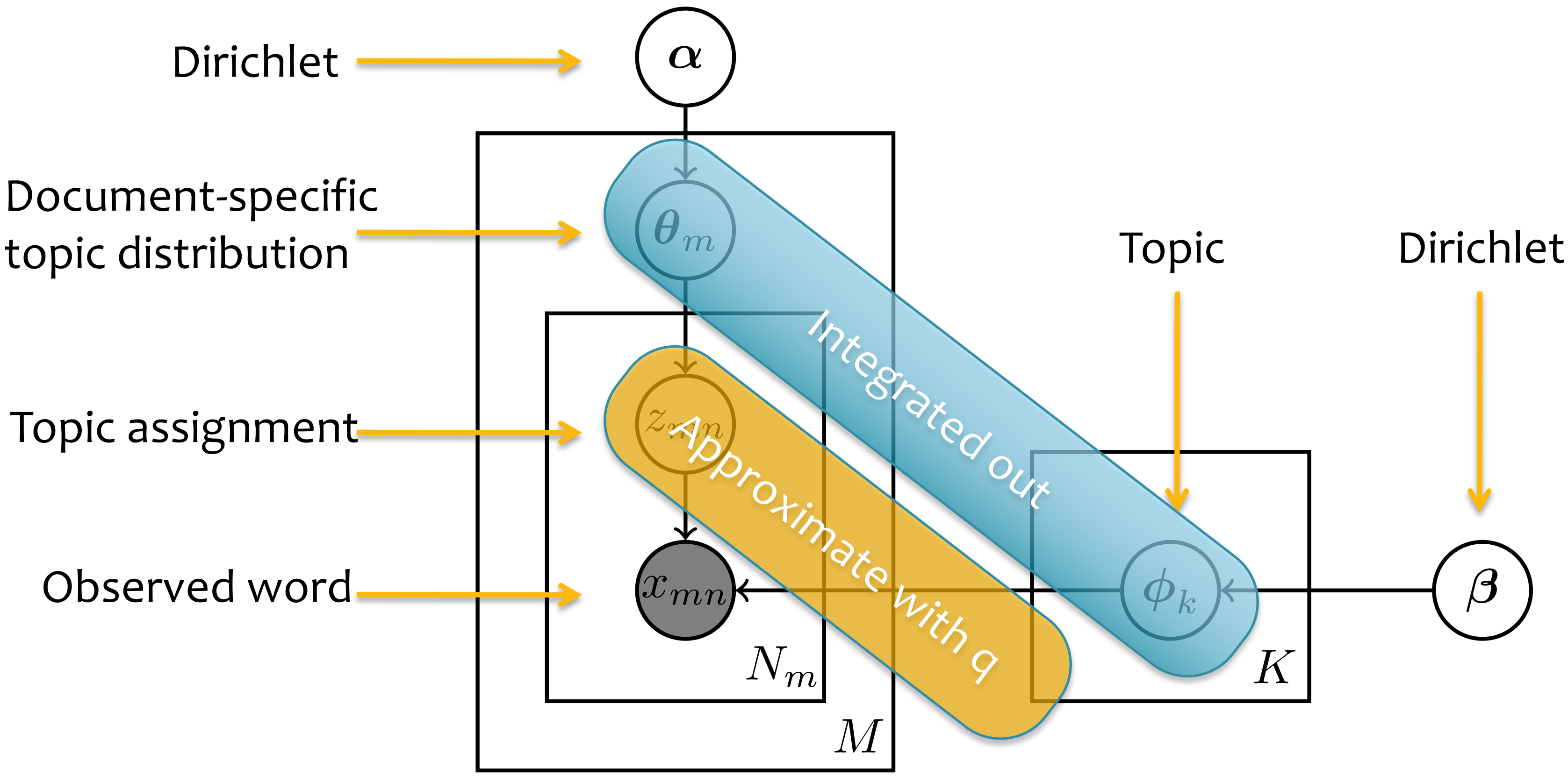
$$\tilde{\gamma}_{ijk} \propto \exp\left(\Psi(\tilde{\alpha}_{jk}) + \Psi(\tilde{\beta}_{kx_{ij}}) - \Psi(\textstyle\sum_w \tilde{\beta}_{kw})\right) \tag{6}$$

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

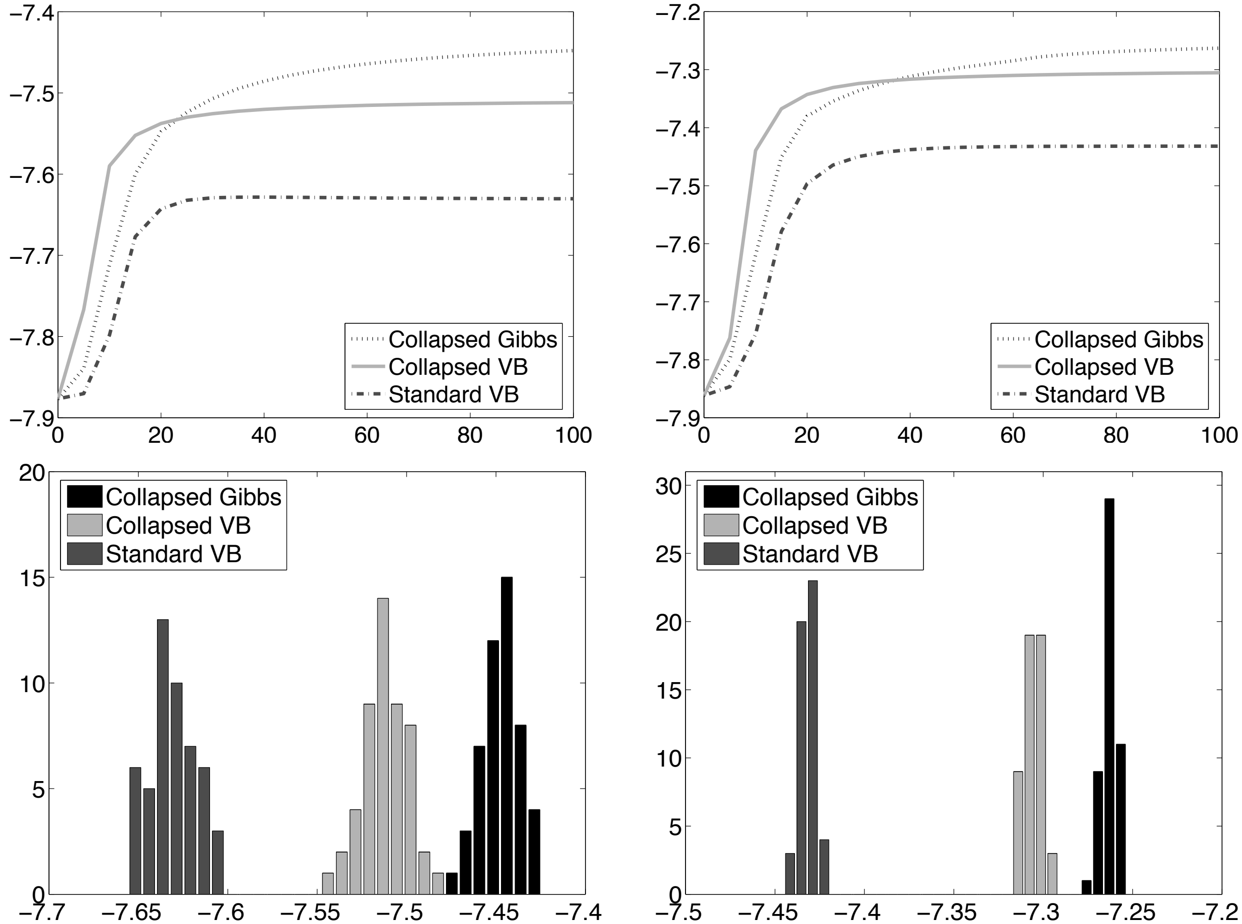


Collapsed Variational Bayesian LDA

- **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.

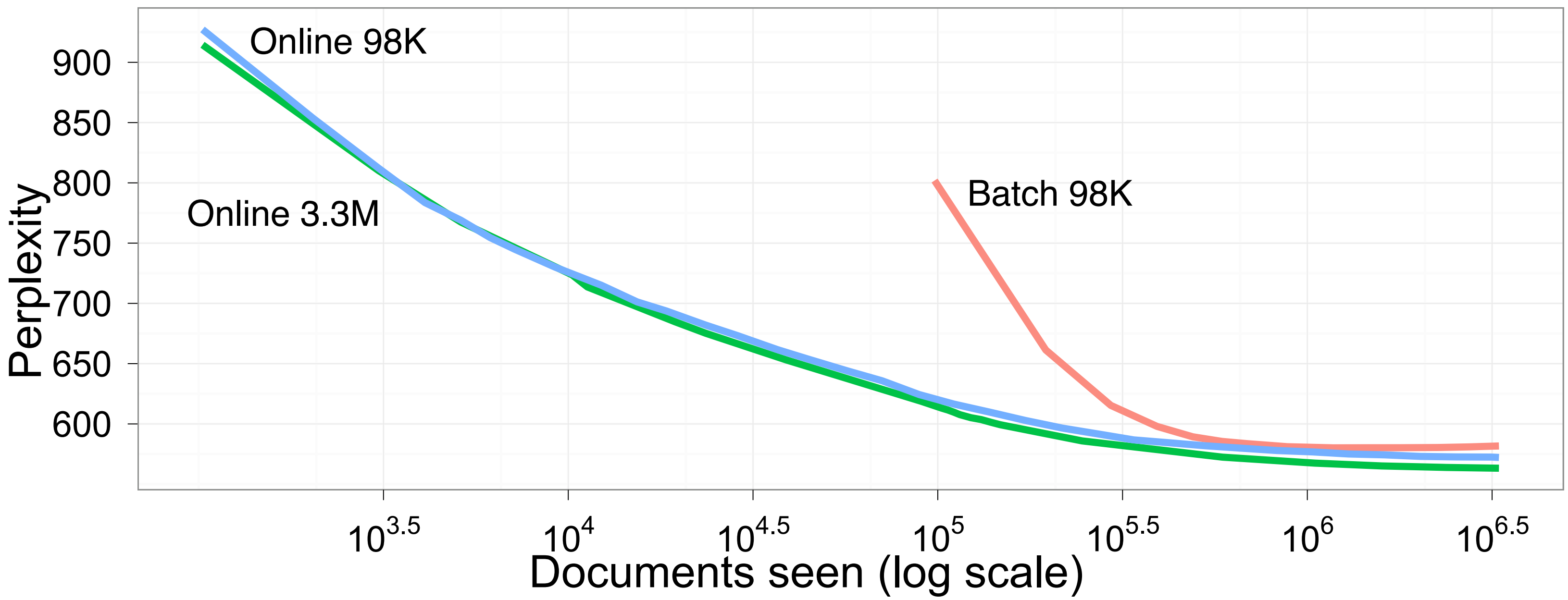
Data from dailykos.com

Data from NeurIPS proceedings



Slide from Teh et al (2007)

Online Variational Bayes for LDA



Documents analyzed	**2048**	**4096**	**8192**	**12288**	**16384**	**32768**	**49152**	**65536**
Top eight words	systems road made service announced national west language	systems health communication service billion language care road	service systems health companies market communication company billion	service systems companies business company billion health industry	service companies systems business company industry market billion	business service companies industry company management systems services	business service companies industry services company management public	business industry service companies services company management public

Figures from Hoffman et al. (2010)

Online Variational Bayes for LDA

Algorithm 1 Batch variational Bayes for LDA

Initialize $\boldsymbol{\lambda}$ randomly.
while relative improvement in $\mathcal{L}(\boldsymbol{w}, \boldsymbol{\phi}, \boldsymbol{\gamma}, \boldsymbol{\lambda}) > 0.00001$ **do**
  E step:
  for $d = 1$ to D **do**
    Initialize $\gamma_{dk} = 1$. (The constant 1 is arbitrary.)
    repeat
      Set $\phi_{dwk} \propto \exp\{\mathbb{E}_q[\log \theta_{dk}] + \mathbb{E}_q[\log \beta_{kw}]\}$
      Set $\gamma_{dk} = \alpha + \sum_w \phi_{dwk} n_{dw}$
    until $\frac{1}{K} \sum_k |\text{change in} \gamma_{dk}| < 0.00001$
  end for
  M step:
  Set $\lambda_{kw} = \eta + \sum_d n_{dw} \phi_{dwk}$
end while

Algorithm 2 Online variational Bayes for LDA

Define $\rho_t \triangleq (\tau_0 + t)^{-\kappa}$
Initialize $\boldsymbol{\lambda}$ randomly.
for $t = 0$ to ∞ **do**
  E step:
  Initialize $\gamma_{tk} = 1$. (The constant 1 is arbitrary.)
  repeat
    Set $\phi_{twk} \propto \exp\{\mathbb{E}_q[\log \theta_{tk}] + \mathbb{E}_q[\log \beta_{kw}]\}$
    Set $\gamma_{tk} = \alpha + \sum_w \phi_{twk} n_{tw}$
  until $\frac{1}{K} \sum_k |\text{change in} \gamma_{tk}| < 0.00001$
  M step:
  Compute $\tilde{\lambda}_{kw} = \eta + D n_{tw} \phi_{twk}$
  Set $\boldsymbol{\lambda} = (1 - \rho_t)\boldsymbol{\lambda} + \rho_t \tilde{\boldsymbol{\lambda}}$.
end for

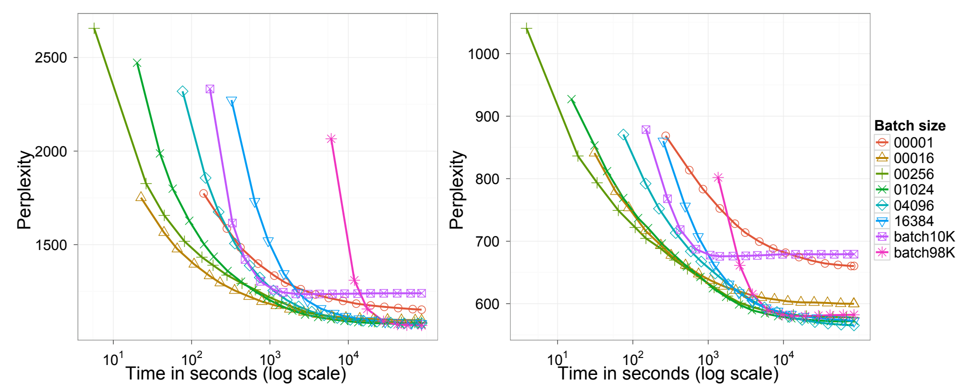


Figure 2: Held-out perplexity obtained on the *Nature* (left) and Wikipedia (right) corpora as a function of CPU time. For moderately large mini-batch sizes, online LDA finds solutions as good as those that the batch LDA finds, but with much less computation. When fit to a 10,000-document subset of the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

Figures from Hoffman et al. (2010)

Fully–Connected CRF

Model

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

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

This is a fully connected graph!

Inference

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

Figures from Krähenbühl & Koltun (2011)

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
lcchen@cs.ucla.edu

George Papandreou *
Google Inc.
gpapan@google.com

Iasonas Kokkinos
CentraleSupélec and INRIA
iasonas.kokkinos@ecp.fr

Kevin Murphy
Google Inc.
kpmurphy@google.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.

Fully–Connected CRF

Model

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

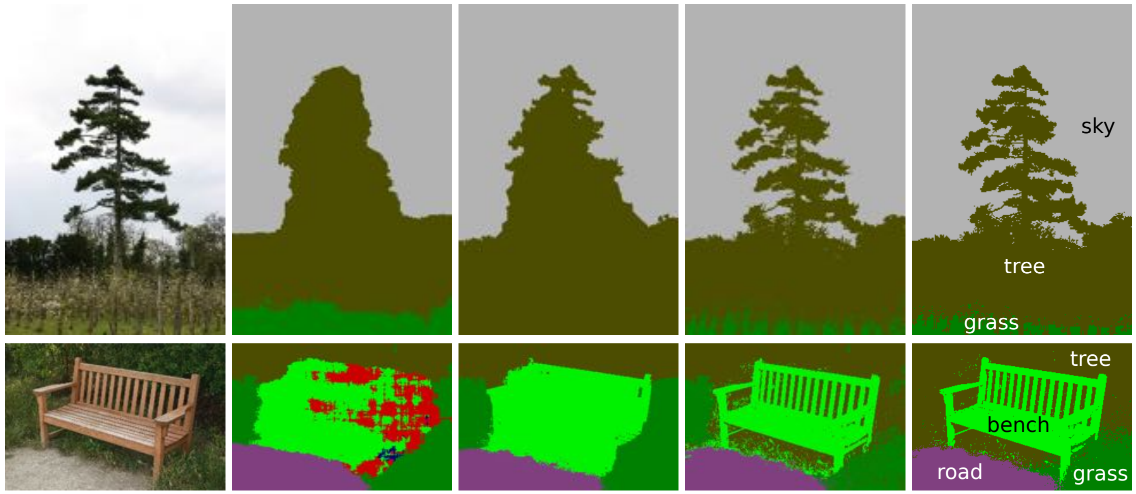
$$E(\mathbf{x}) = \sum_i \psi_u(x_i) + \sum_{i<j} \psi_p(x_i, x_j),$$

This is a fully connected graph!

Inference

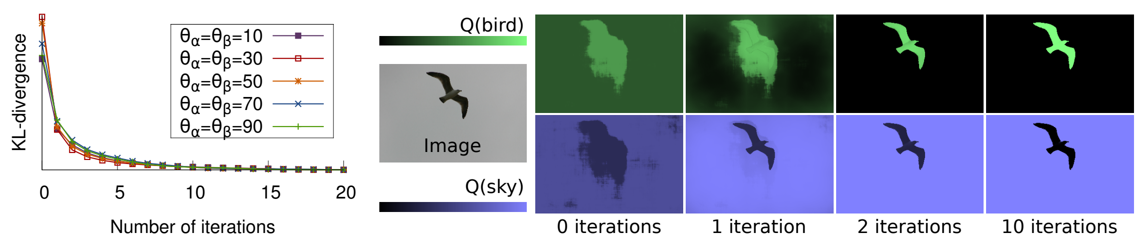
- Can do MCMC, but slow
- Instead use Variational Inference
- Then filter some variables for speed up

Results



(a) Image (b) Unary classifiers (c) Robust P^n CRF (d) Fully connected CRF, MCMC inference, 36 hrs (e) Fully connected CRF, our approach, 0.2 seconds

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.



(a) KL-divergence (b) Distributions $Q(X_i = \text{“bird”})$ (top) and $Q(X_i = \text{“sky”})$ (bottom)

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.

Figures from Krähenbühl & Koltun (2011)

Joint Parsing and Alignment with Weakly Synchronized Grammars

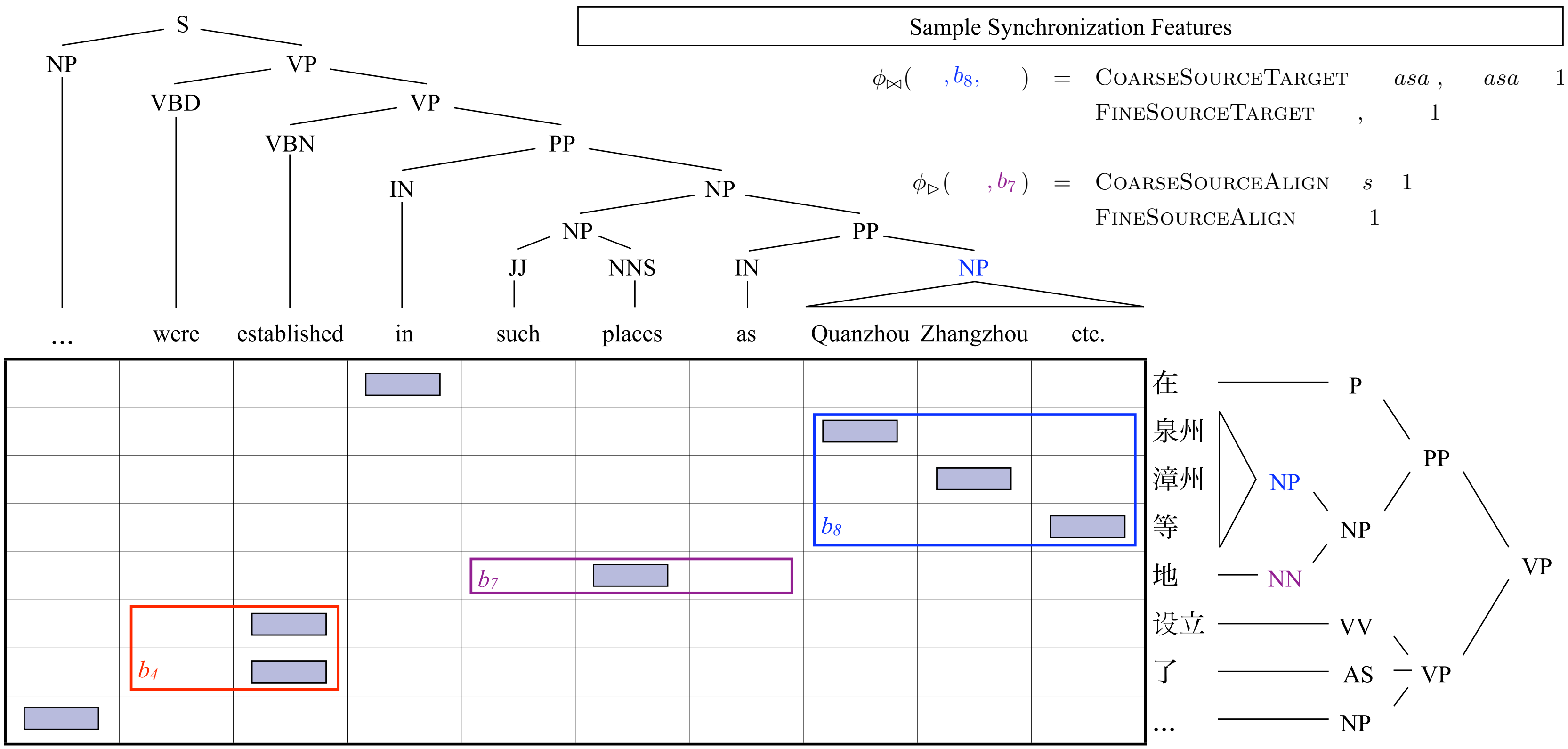
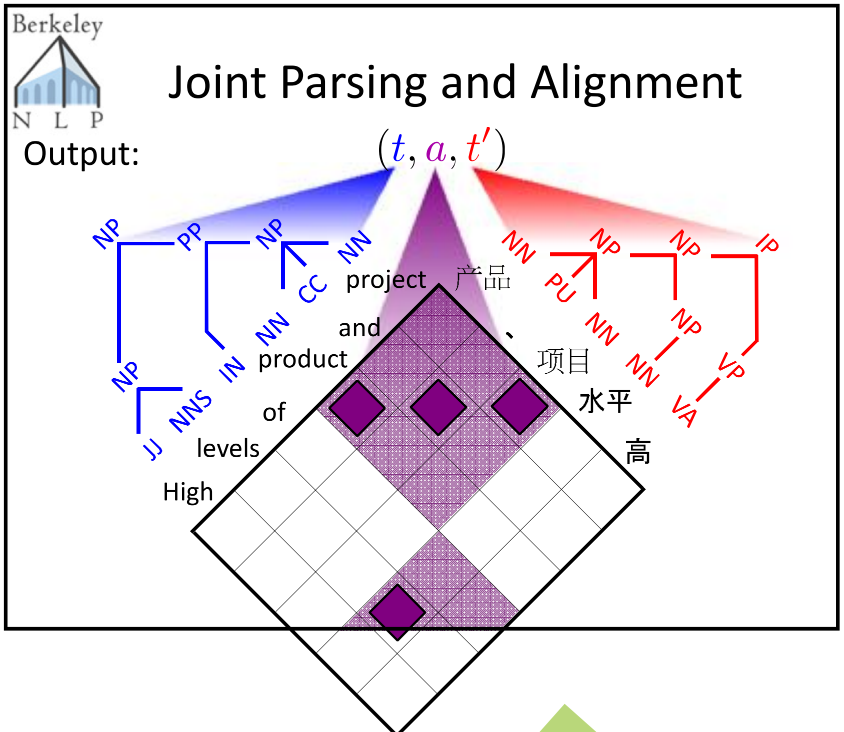


Figure 2: An example of a Chinese-English sentence pair with parses, word alignments, and a subset of the full optimal ITG derivation, including one totally unsynchronized bispan (b_4), one partially synchronized bispan (b_7), and and fully synchronized bispan (b_8). The inset provides some examples 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)



This uses
Structured
Mean Field V.I.

	Test Results		
	Ch F_1	Eng F_1	Tot F_1
Monolingual	83.6	81.2	82.5
Reranker	**86.0**	83.8	84.9
Joint	85.7	**84.5**	**85.1**

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

	Test Results			
	Precision	Recall	AER	F_1
HMM	86.0	58.4	30.0	69.5
ITG	**86.8**	73.4	20.2	79.5
Joint	85.5	**84.6**	**14.9**	**85.0**

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

Figures from Burkett et al. (2010)

HIDDEN STATE CRFS

Case Study: Object Recognition

Data consists of images $\boldsymbol{x}$ and labels y.

pigeon

rhinoceros

leopard

llama

Case Study: Object Recognition

Data consists of images x and labels y.

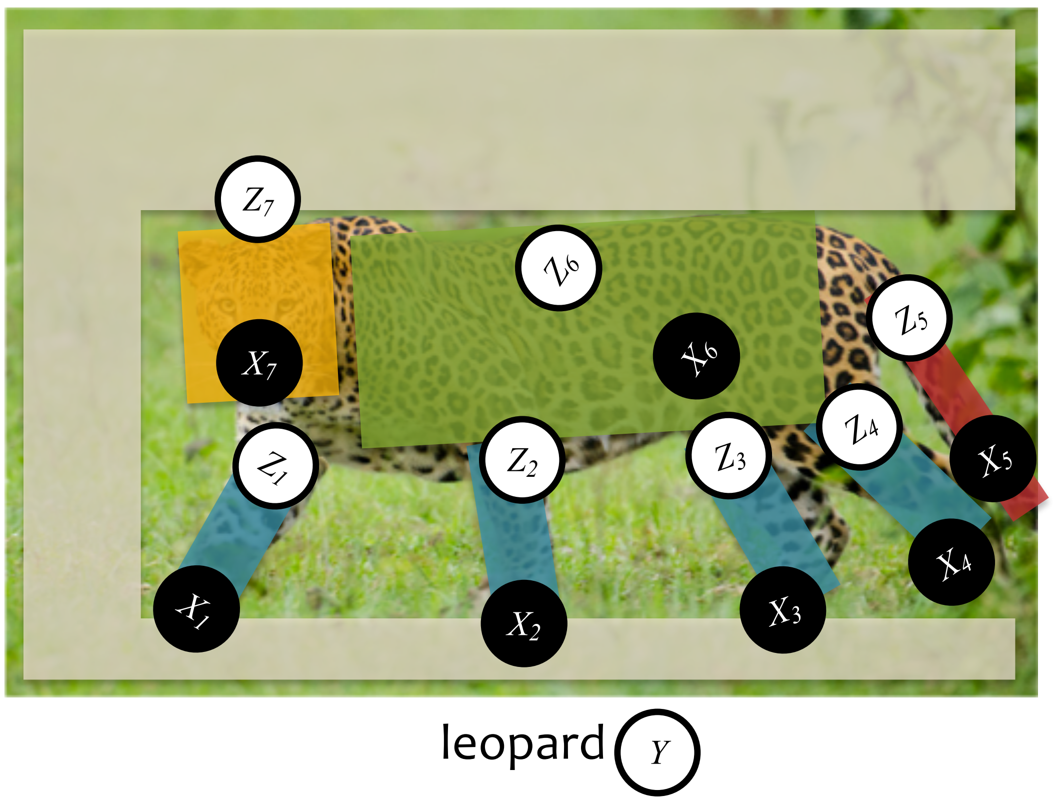
- Preprocess data into "patches"
- Posit a latent labeling z describing the object's parts (e.g. head, leg, tail, torso, grass)
- Define graphical model with these latent variables in mind
- z is not observed at train or test time

leopard

Case Study: Object Recognition

Data consists of images $\boldsymbol{x}$ and labels y.

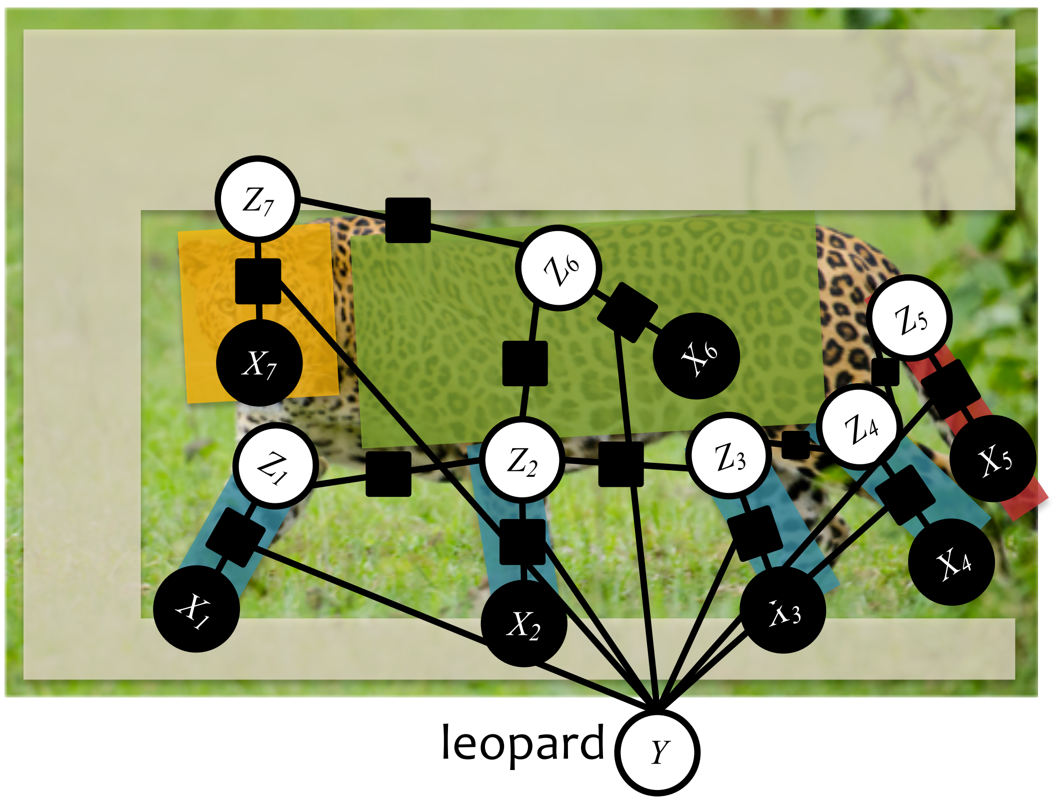
- Preprocess data into "patches"
- Posit a latent labeling z describing the object's parts (e.g. head, leg, tail, torso, grass)
- Define graphical model with these latent variables in mind
- z is not observed at train or test time



Case Study: Object Recognition

Data consists of images $\boldsymbol{x}$ 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

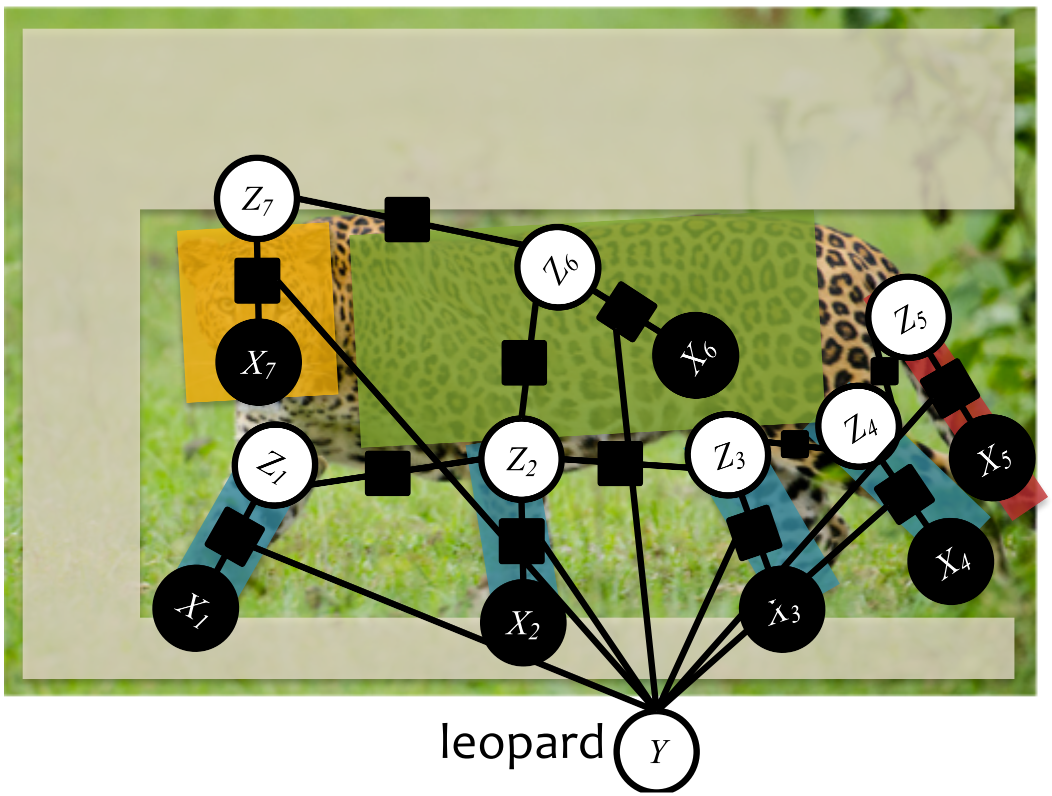


Hidden-state CRFs

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

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

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



Hidden-state CRFs

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

Joint model: $p_{\boldsymbol{\theta}}(\boldsymbol{y}, \boldsymbol{z} \mid \boldsymbol{x}) = \dfrac{1}{Z(\boldsymbol{x}, \boldsymbol{\theta})} \prod_{\alpha} \psi_\alpha(\boldsymbol{y}_\alpha, \boldsymbol{z}_\alpha, \boldsymbol{x})$

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

We can train using gradient based methods:

(the values $\boldsymbol{x}$ are omitted below for clarity)

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

$$= \sum_{n=1}^{N} \sum_{\alpha} \left(\sum_{\boldsymbol{z}_{\boldsymbol{\alpha}}} \underbrace{p_{\boldsymbol{\theta}}(\boldsymbol{z}_{\boldsymbol{\alpha}} \mid \boldsymbol{y}^{(n)})}_{\text{Inference on } \textbf{clamped} \text{ factor graph}} f_{\alpha,j}(\boldsymbol{y}_\alpha^{(n)}, \boldsymbol{z}_\alpha) - \sum_{\boldsymbol{y}_{\boldsymbol{\alpha}}, \boldsymbol{z}_{\boldsymbol{\alpha}}} \underbrace{p_{\boldsymbol{\theta}}(\boldsymbol{y}_{\boldsymbol{\alpha}}, \boldsymbol{z}_{\boldsymbol{\alpha}})}_{\text{Inference on } \textbf{full} \text{ factor graph}} f_{\alpha,j}(\boldsymbol{y}_{\boldsymbol{\alpha}}, \boldsymbol{z}_{\boldsymbol{\alpha}}) \right)$$