CSC2541: Differentiable Inference and Generative Models

Lecture 2: Variational autoencoders

Admin:

- TAs:
  - Tony Wu (ywu@cs.toronto.edu)
  - Kamal Rai (kamal.rai@mail.utoronto.ca)
- Extra seminar: Model-based Reinforcement learning
- Seminar sign-up

Seminars

- 7 weeks of seminars, about 8-9 people each
- Each day will have one or two major themes, 3-6 papers covered
- Divided into 2-3 presentations of about 30-40 mins each
- Explain main idea, relate to previous work and future directions

Computational Tools

- Automatic differentiation
- Neural networks
- Stochastic optimization
- Simple Monte Carlo

Computational Tools

- Can specify arbitrarily-flexible functions with a deep net:

$$y = f_\theta(x)$$

- Can specify arbitrarily complex conditional distributions with a deep net:
  - Density networks: $p(y|x) = \mathcal{N}(y|\mu = f_\theta(x), \Sigma = g_\theta(x))$

$$p(y = c|x) = \frac{1}{Z_\theta} \exp([f_\theta(x)]_c)$$

  - Bayesian neural network: $p(y|x) = \int f_\theta(x) p(\theta) d\theta$

Computational Tools

- Can optimize continuous parameters wrt any objective given unbiased estimates of its gradient.
- given $\mathbb{E}_{p(x)}\left[grad(J)(\theta, x)\right] = \nabla_\theta J(\theta)$
- can use: $\hat{\theta} = \mathrm{SGD}(\theta_{\mathrm{init}}, \hat{\mathrm{g}}\mathrm{rad}(\mathrm{J})) \approx \mathrm{argmin}_\theta(J)$

Computational Tools

- Can differentiate any deterministic, continuous function using reverse-mode automatic differentiation (backprop)
- Cost of evaluating gradient about same as evaluating function

Computational Tools

- Simple Monte Carlo gives unbiased estimates of integrals given samples

Benefits of Bayesianism

- Examples: Diagnosing disease, doing regression
- Captures uncertainty
  - Necessary for decision-making
  - Why pretend we’re certain?
- Automatic regularization from ensembling
- Latent variables can be meaningful
- Can combine datasets/models (semi-supervised learning)
- Marginal likelihood automatically chooses model capacity
- Inference is deterministic given model, automatic answer for hyperparameters

What is inference?

- Estimate posterior: $$p(z|x,\theta) = \frac{p(x|z,\theta)p(z)}{\int p(x|z',\theta)p(z')dz'}$$
- Compute expectations: $$\mathbb{E}_{p(z|x,\theta)}\left[f(z|x,\theta)\right]$$
- Make predictions: $$p(x_2|x_1,\theta) = \int p(x_2|z)p(z|x_1,\theta)dz$$
- Marginal likelihood: $$p(x|\theta) = \int p(z)p(z|x,\theta)dz$$
- Can all be estimated using samples from the posterior and Simple Monte Carlo!

From IS to Variational Inference

[from Shakir Mohamed]

Integral problem

$$\log p(y) = \log \int p(y|z)p(z)dz$$

Proposal

$$\log p(y) = \log \int p(y|z)p(z)\frac{q(z)}{q(z)}dz$$

Importance Weight

$$\log p(y) = \log \int p(y|z)\frac{p(z)}{q(z)}q(z)dz$$

Jensen's inequality

$$\log \int p(x)g(x)dx \geq \int p(x)\log g(x)dx$$

$$\log p(y) \geq \int q(z)\log\left(p(y|z)\frac{p(z)}{q(z)}\right)dz$$

$$= \int q(z)\log p(y|z) - \int q(z)\log\frac{q(z)}{p(z)}$$

Variational lower bound

$$\boxed{= \mathbb{E}_{q(z)}[\log p(y|z)] - KL[q(z)\|p(z)]}$$

Interpretations

- Bound maximized when $q(z|x) = p(z|x)$
- Reconstruction + difference from prior
- MAP + Entropy

Show demos

- Toy example
- Mixture example
- Bayesian neural network

When we have lots of data, and global model parameters:

$$p(x|\theta) = \prod_{i=1}^{N} (x_i|z_i, \theta) p(z_i) d\theta$$

- Can alternate optimizing variational parameters, model parameters
- A generalization of Expectation-Maximization
- Slow because of alternating optimization - need to update theta, then each $q(z_i|x_i, \theta)$
- Slow and memory-intensive when we have many datapoints

Variational autoencoders

- Model: Latent-variable model p(x|z, theta) usually specified by a neural network

- Inference: Recognition network for q(z|x, theta) usually specified by a neural network

- Training objective: Simple Monte Carlo for unbiased estimate of Variational lower bound

- Optimization method: Stochastic gradient ascent, with automatic differentiation for gradients

Show VAE demo

- Maximizing ELBO, or minimizing KL from true posterior

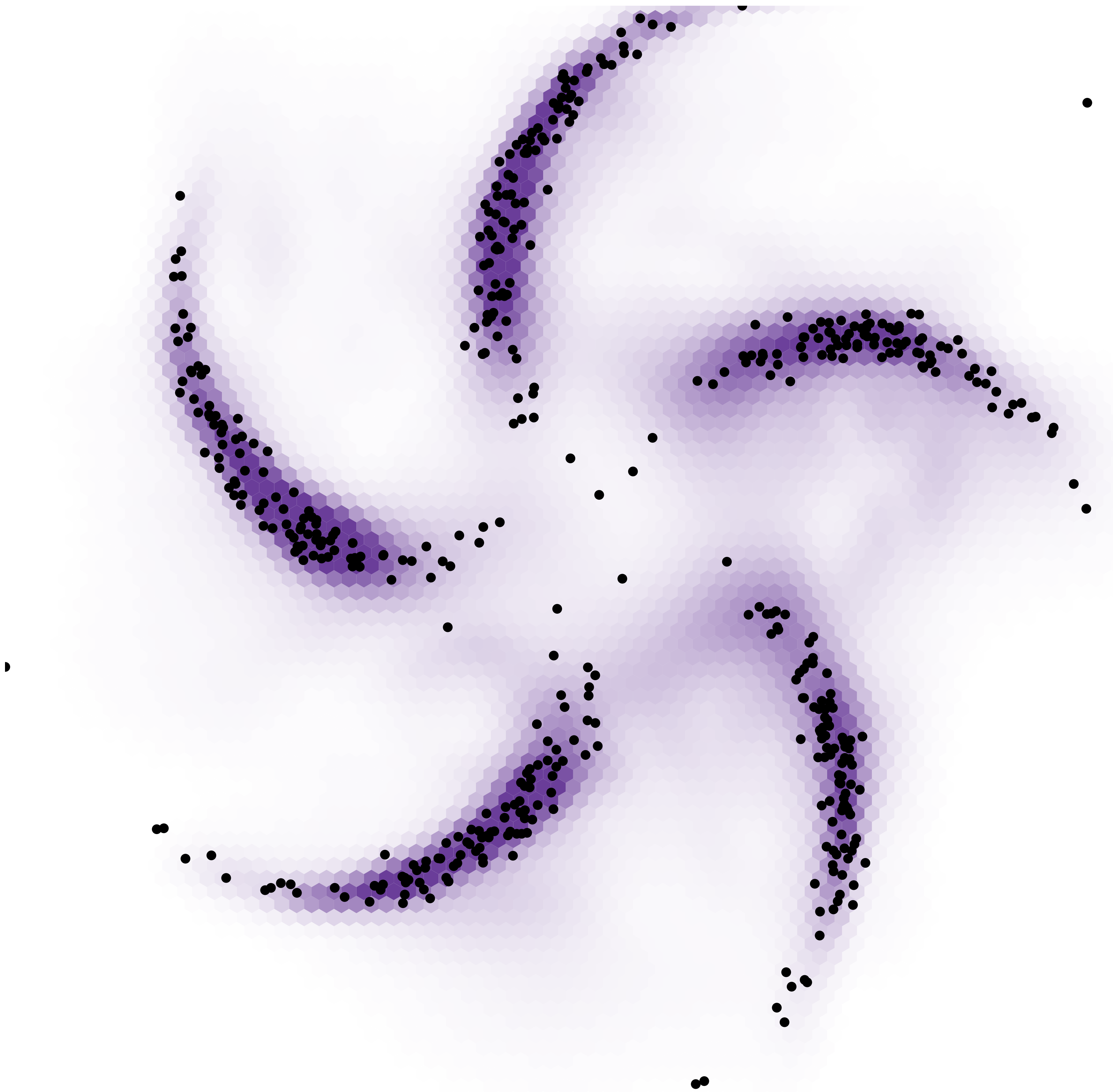
- Relation to denoting autoencoders: Training ‘encoder’ and ‘decoder’ together

- Decoder specifies model, encoder specifies inference

Pros and Cons

- Flexible generative model
- End-to-end gradient training
- Measurable objective (and lower bound - model is at least this good)
- Fast test-time inference

- Cons:
    - sub-optimal variational factors
    - limited approximation to true posterior (will revisit)
    - Can have high-variance gradients



Questions

Class Projects

- **Develop a generative model for a new medium**
- **Extend existing models, inference, or training**
- **Apply an existing approach in a new way**
- **Review / comparison / tutorials**

Other ideas

- Backprop through BEAM search
- Backprop through dynamic programming for DNA alignment
- Conditional GANs for mesh upsampling
- Apply VAE SLDS to human speech
- Generate images from captions
- Learn to predict time-reversed physical dynamics
- Investigate minimax optimization methods for GANS
- Model-based RL (show demo)