Latent-Variable Models

A Brief Overview

M. Rule

A teaching talk for lecturer in machine learning at the University of Edinburgh School of Informatics
20th June, 2022

Latent variable models are statistical models where some random quantities are not directly measured.

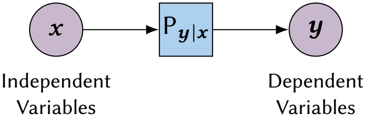
Learning goals:

1. What are some uses of latent-variable models?
2. General ideas used to design inference algorithms:
   - What is expectation maximization?
   - What is variational Bayes?
3. What is a variational autoencoder?
   - What is amortized inference?
   - How can we train them via backpropagation?

Statistical models with latent variables

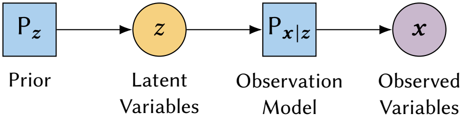
Fully-observed model

- Observe $(\boldsymbol{x}, \boldsymbol{y})$ and learn $\boldsymbol{x} \to \boldsymbol{y}$



Latent-variable model

- Explain variation in observed $\boldsymbol{x}$ using latent causes $\boldsymbol{z}$



Use a latent variable model:

Measurements are indirect observations (of an underlying simpler structure)

- Capture complicated interactions using a few latent variables and explore high-dimensional data

Generative: $\approx$ probability of $\boldsymbol{x}$ in training data $\mathrm{P}_{\boldsymbol{x}}$

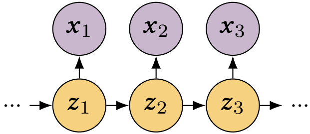
- Draw new samples similar to the training data
- Get probability of a scenario given the model and detect outliers or surprising inputs

Latent variable models are statistical models where some random quantities are not directly measured.

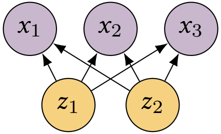
They can be used, for example, to infer missing data, explain complicated relationships with simpler ones, learn the distribution of the training data.

Some types of latent-variable models

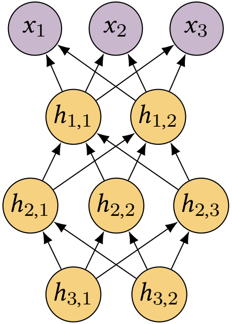
Hidden Markov model



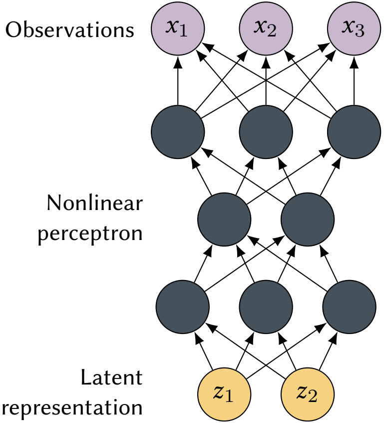
Factor Analysis



Deep Belief Network



Deep generative model



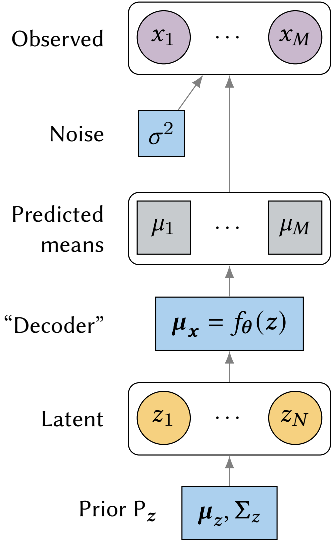
Part of Generative Adversarial Networks and *Variational Autoencoders* (and others)

Latent variable models are statistical models where some random quantities are not directly measured.

They can be used, for example, to infer missing data, explain complicated relationships with simpler ones, learn the distribution of the training data.

They encompass a wide variety of problems: filtering, state inference, generative models, unsupervised learning, to name a few examples.

Concrete Example: A generative model with a Gaussian latent space



Gaussian prior on latents $\mathrm{P}_{\boldsymbol{z}} = \mathcal{N}(\boldsymbol{\mu}_z, \Sigma_z)$.

- Keep it simple: $\boldsymbol{\mu}_z = 0, \Sigma_z = \boldsymbol{I}_N$

"Decoder" generates mean $\mu_i = [f_\theta(z)]_i$ for each x_i

Model x as conditionally Gaussian given z:

- $\mathrm{P}^{\boldsymbol{\theta}}_{\boldsymbol{x}|\boldsymbol{z}} = \mathcal{N}[f_{\boldsymbol{\theta}}(\boldsymbol{z}), \sigma^2 \boldsymbol{I}_M]$

Recall

- Marginal probability: $\mathrm{P}_{\boldsymbol{x}} = \int_{dz} \mathrm{P}_{\boldsymbol{x},\boldsymbol{z}}$
- Conditional probability chain rule: $\mathrm{P}_{\boldsymbol{x},\boldsymbol{z}} = \mathrm{P}_{\boldsymbol{x}|\boldsymbol{z}} \mathrm{P}_{\boldsymbol{z}}$

$$\mathrm{P}^{\boldsymbol{\theta}}_{\boldsymbol{x}} = \int_{dz} \mathrm{P}^{\boldsymbol{\theta}}_{\boldsymbol{x},\boldsymbol{z}} = \int_{dz} \mathrm{P}^{\boldsymbol{\theta}}_{\boldsymbol{x}|\boldsymbol{z}} \mathrm{P}_{\boldsymbol{z}}$$

Excellent: We now have a generative model $\mathrm{P}_{\boldsymbol{x}} \approx \mathrm{P}^{\theta}_{\boldsymbol{x}}$

How do I train a latent variable model?

Given a set of (independent) observations $\mathscr{X} = \{\boldsymbol{x}_1, .., \boldsymbol{x}_L\}$, maximize average log-likelihood

$$\hat{\boldsymbol{\theta}} \leftarrow \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \langle \ln \mathsf{P}_{\boldsymbol{x}}^{\boldsymbol{\theta}} \rangle_{\boldsymbol{x} \in \mathscr{X}}$$

Easier said than done: We need to integrate $\mathsf{P}_{\boldsymbol{x}}^{\boldsymbol{\theta}} = \int_{d\boldsymbol{z}} \mathsf{P}_{\boldsymbol{x}|\boldsymbol{z}}^{\boldsymbol{\theta}} \mathsf{P}_{\boldsymbol{z}}$

- *This may lack a closed-form solution and be impractical to sample*

Many methods for estimating latent states and parameters

- Sample from the latent states (and/or unknown parameters)
- Message-passing: update parts of a larger model based on dependencies between variables
- More: Bayesian and assumed-density filtering, contrastive learning, Viterbi and dynamic programming

Today, briefly: **Expectation maximization**; **Variational Bayes**;

Expectation Maximization

Know $\boldsymbol{z}$ and $\boldsymbol{x}$? Get $\boldsymbol{\theta}$ via supervised learning

Know $\boldsymbol{\theta}$, $\boldsymbol{x}$, $\mathrm{P}_{\boldsymbol{z}}$? Get $\mathrm{P}^{\boldsymbol{\theta}}_{\boldsymbol{z}|\boldsymbol{x}}$ via Bayes' rule

Choose initial θ and iterate:

- **Expectation:**
  - Given $\boldsymbol{\theta}$, get $\mathrm{P}^{\boldsymbol{\theta}}_{\boldsymbol{z}|\boldsymbol{x}}$
  - Loss $\mathcal{L}_{\boldsymbol{\theta}}(\hat{\boldsymbol{\theta}})$: average $\ln \mathrm{P}^{\hat{\boldsymbol{\theta}}}_{\boldsymbol{x}|\boldsymbol{z}}$ over $\mathrm{P}^{\boldsymbol{\theta}}_{\boldsymbol{z}|\boldsymbol{x}}$
- **Maximization:**
  - Adjust $\hat{\boldsymbol{\theta}}$ to maximize $\mathcal{L}_{\boldsymbol{\theta}}(\hat{\boldsymbol{\theta}})$

Converges to *local* optimum

Variational Bayes

Calculating $\mathrm{P}^{\boldsymbol{\theta}}_{\boldsymbol{z}|\boldsymbol{x}}$ *may be intractable*

Approximate $\mathrm{P}^{\boldsymbol{\theta}}_{\boldsymbol{z}|\boldsymbol{x}}$ with simpler $\mathrm{Q}^{\boldsymbol{\phi}}_{\boldsymbol{z}|\boldsymbol{x}}$ (parameters $\boldsymbol{\phi}$)

Adjust $\boldsymbol{\phi}$ to maximize Evidence Lower Bound (ELBO)

$$\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi}) := \langle \ln \mathrm{P}^{\boldsymbol{\theta}}_{\boldsymbol{x}|\boldsymbol{z}} \rangle_{\mathrm{Q}^{\boldsymbol{\phi}}_{\boldsymbol{z}|\boldsymbol{x}}} + \langle \ln \mathrm{P}_{\boldsymbol{z}} \rangle_{\mathrm{Q}^{\boldsymbol{\phi}}_{\boldsymbol{z}|\boldsymbol{x}}} + \mathrm{H}[\mathrm{Q}^{\boldsymbol{\phi}}_{\boldsymbol{z}|\boldsymbol{x}}]$$

- $\langle \ln \mathbf{P}^{\boldsymbol{\theta}}_{\boldsymbol{x}|\boldsymbol{z}} \rangle_{\mathbf{Q}}$: Choose $\boldsymbol{z}$ consistent with $\boldsymbol{x}$
- $\langle \ln \mathbf{P}_{\boldsymbol{z}} \rangle_{\mathbf{Q}}$: Keep $\mathrm{Q}^{\boldsymbol{\phi}}_{\boldsymbol{z}|\boldsymbol{x}}$ close to prior
- $\mathbf{H}[\mathbf{Q}^{\boldsymbol{\phi}}_{\boldsymbol{z}|\boldsymbol{x}}]$: Maximize entropy

In the Homework:

- Show: $\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi}) \leq \ln \mathrm{P}^{\boldsymbol{\theta}}_{\boldsymbol{x}}$
- Show: maximizing $\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi})$ in $\boldsymbol{\phi} \longrightarrow$ tighter bound

Latent variable models are statistical models where some random quantities are not directly measured.

They can be used, for example, to infer missing data, explain complicated relationships with simpler ones, learn the distribution of the training data.

They encompass a wide variety of problems: filtering, state inference, generative models, unsupervised learning, to name a few examples.

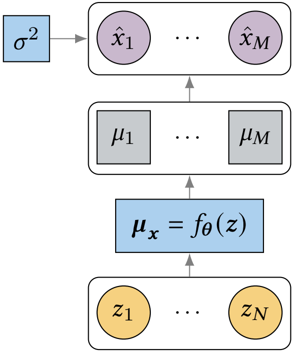
It can be hard to directly optimize the likelihood of a latent variable model. There are many inference approaches and approximations.

Expectation maximization and variational Bayes try to approximate the maximum-likelihood parameters while learning distributions over latent states.

The “variational” in variational autoencoders comes from the fact that we use variational Bayes to train them. Where does the “autoencoder” come from?

Variational Autoencoders: Amortized inference

Optimizing $Q^{\phi}_{z|\boldsymbol{x}}$ for each $\boldsymbol{x}$ is slow

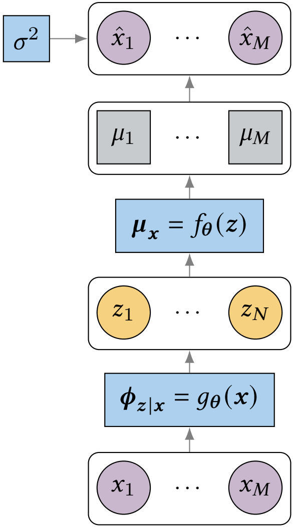


Variational Autoencoders: Amortized inference

Optimizing $Q^{\phi}_{z|\boldsymbol{x}}$ for each $\boldsymbol{x}$ is slow

Amortized inference:

- Predict $\boldsymbol{\phi}_{z|\boldsymbol{x}} = \{\boldsymbol{\mu}_{z|\boldsymbol{x}}, \boldsymbol{\sigma}_{z|\boldsymbol{x}}\}$ from $\boldsymbol{x}$
  - Train neural network to map $g : \boldsymbol{x} \longrightarrow \boldsymbol{\phi}$
- Evaluate $\ln P^{\theta}_{\boldsymbol{x}|z}$
  - Sample $z \leftarrow \xi \boldsymbol{\sigma}_{z|\boldsymbol{x}} + \boldsymbol{\mu}_{z|\boldsymbol{x}}$ where $\xi \sim \mathcal{N}[0, \boldsymbol{I}_N]$
  - Compute $\ln P^{\theta}_{\boldsymbol{x}|z}$
- How can I train this?! It is stochastic?



Variational Autoencoders: Amortized inference

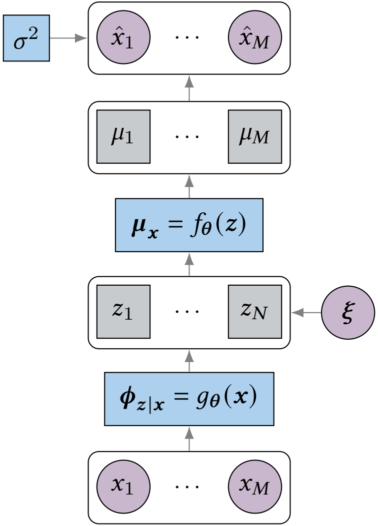
Optimizing $Q^{\phi}_{z|x}$ for each x is slow

Amortized inference:

- Predict $\boldsymbol{\phi}_{z|x} = \{\boldsymbol{\mu}_{z|x}, \boldsymbol{\sigma}_{z|x}\}$ from $\boldsymbol{x}$
  - Train neural network to map $g : \boldsymbol{x} \longrightarrow \boldsymbol{\phi}$
- Evaluate $\ln P^{\theta}_{x|z}$
  - Sample $z \leftarrow \xi \boldsymbol{\sigma}_{z|x} + \boldsymbol{\mu}_{z|x}$ where $\xi \sim \mathcal{N}[0, \boldsymbol{I}_N]$
  - Compute $\ln P^{\theta}_{x|z}$
- How can I train this?! It is stochastic?

Reparameterization trick:

- Freeze ξ: no more random quantities
- Backpropagation:
  - Tune f_θ to increase log-likelihood
  - Tune g_θ to improve $\boldsymbol{\phi}_{z|x}$



Variational Autoencoders: Amortized inference

Optimizing $Q^{\boldsymbol{\phi}}_{z|\boldsymbol{x}}$ for each $\boldsymbol{x}$ is slow

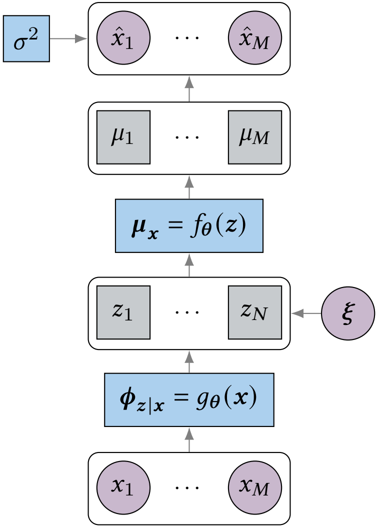
Amortized inference:

- Predict $\boldsymbol{\phi}_{z|\boldsymbol{x}} = \{\boldsymbol{\mu}_{z|\boldsymbol{x}}, \boldsymbol{\sigma}_{z|\boldsymbol{x}}\}$ from $\boldsymbol{x}$
  - Train neural network to map $g : \boldsymbol{x} \longrightarrow \boldsymbol{\phi}$
- Evaluate $\ln P^{\boldsymbol{\theta}}_{\boldsymbol{x}|z}$
  - Sample $z \leftarrow \xi \boldsymbol{\sigma}_{z|\boldsymbol{x}} + \boldsymbol{\mu}_{z|\boldsymbol{x}}$ where $\xi \sim \mathcal{N}[0, \boldsymbol{I}_N]$
  - Compute $\ln P^{\boldsymbol{\theta}}_{\boldsymbol{x}|z}$
- How can I train this?! It is stochastic?

Reparameterization trick:

- Freeze ξ: no more random quantities
- Backpropagation:
  - Tune f_θ to increase log-likelihood
  - Tune g_θ to improve $\boldsymbol{\phi}_{z|\boldsymbol{x}}$

Use stochastic gradient descent (and many tricks)



Latent variable models are statistical models where some random quantities are not directly measured.

They can be used, for example, to infer missing data, explain complicated relationships with simpler ones, learn the distribution of the training data.

They encompass a wide variety of problems: filtering, state inference, generative models, unsupervised learning, to name a few examples.

It can be hard to directly optimize the likelihood of a latent variable model. There are many inference approaches and approximations.

Expectation maximization and variational Bayes learn and approximate maximum-likelihood parameters while learning distributions over latent states

Variational autoencoders are a type of deep latent-variable model that maps latent states to observations using a decoding network, and computes approximate distributions over latent states using an encoding network

Appendix

"Zero-Shot Text-to-Image Generation"
Ramesh et al. (2021)
`openai.com/blog/dall-e`

Example: Gaussian mixture model

- **E:** Get $\Pr$ that $\boldsymbol{x}$ comes from each cluster
- ***M*:** Use as weights to update μ, Σ for all clusters