Inference and Representation: An Odyssey

Rahul G. Krishnan

New York University

Lab 14, Dec 13, 2016

Outline

Autoencoders

- *Hypothesis:* Reconstruct data by passing it through an "information bottleneck" to learn about data generating process
- Two parts: Encoder and decoder
- Many variants: Denoising autoencoder, variational autoencoder
- Extended to sequence level data by seq2seq models (used extensively for Machine Translation)

Variational Autoencoders

- Proper probabilistic model: $p(z)p(x|z)$
- Instead of EM, we do stochastic gradient ascent in the ELBO with respect to:
  - The generative model: With CPDs typically parameterized by conditional probability distributions
  - The inference network: Use to approximate the optimal variational parameters during learning

Generative Adversarial Networks

- Directed generative model
- Sample from random noise, pass through generator to transform to domain of interest
- Learning uses a discriminator to distinguish between real and generated samples
- Within the class of likelihood free models

Tradeoffs

- **Objective Function:** VAEs optimize a valid bound on the marginal likelihood of the data. GANs are hard to evaluate quantitatively
- **Sample Quality:** On images, VAEs tend to produce blurry images vs GANs that tend to produce more realistic samples
- **Inference:** VAEs are learned with an inference network posterior inference. GANs typically do not though methods have been proposed to alleviate this (BiGAN, ALI)
- **Likelihood:** VAEs specify a distribution for which the input is drawn under. GANs optimize what is colloquially called a "perceptual" loss.

Review

- Lets see a map of everything we have studied.

Some tips for debugging

- As you've gone through your project, you've probably developed ways to debug your code and algorithm
- We'll talk about a few methods that are commonly used
- **Baselines:** No way to know if you're doing something useful without it
- **Modularity:** Reason about your errors by decomposing them

Approximation Error

- How much does limiting yourself to decision trees, random forests, deep neural networks affect you?
- **Symptoms:** Overfitting (your model won't generalize) or underfitting (you're not doing as well as you could)

Estimation Error

- How much does having finite data affect you? A function of the choice of hypothesis class.
- A more complex hypothesis class will require, in general, more data to achieve a low estimation error
- **Suggestions:** Data augmentation, pre-train parameters on another related task

Optimization Error

- What if you had used a larger learning rate? How about a more complex optimization scheme (such as Newton's method)?
- How sensitive is the final test accuracy to how well you solve the optimization problem of interest?

Some strategies

- Taken from “Debugging machine learning”

1. Trivially overfit: add the label as a feature. If this fails, it means your optimizer likely has a bug or needs an upgrade.
2. Remove a subset of training data. If your validation accuracy takes a severe hit, you need more data.
3. Typically when you have “enough” data, your test accuracy plateaus by reducing the training data.

References

- Do you have any more add? :)
- The Tradeoffs of Large Scale Learning
- Debugging machine learning

Application Specific Techniques

- The algorithms you learned here are general. For specific applications, some models might work better than others
- NLP: Sentiment analysis, text summarization, machine translation
- Computer Vision: Predicting objects, bounding boxes, action recognition etc.
- Multimodal Learning: Caption generation, audio transcription
- Healthcare, speech, politics and many others

Semi-supervised learning

- If we have a million training examples, we can train good classifier.
- What if we only had a hundred data points?
- One potential approach: Leverage generative models to learn better classifiers Kingma et al

Domain Adaptation (Covariate Shift)

- We've always assumed that the train and test data are drawn from the same distribution
- What if this assumption was violated? Can we still reliably train a classifier?

Probabilistic Logic

- A natural way to combine probabilistic logic with graphical models
- Markov Logic Networks

Optimization in ML

- Combinatorial Optimization:
  - MAP estimation in MRFs
- Convex Optimization:
  - Marginal Inference in MRFs
- Non-convex Optimization:
  - Stochastic gradient descent in deep learning: can only find local optima but often good enough
- For example see : Tutorial on Frank-Wolfe and Greedy Optimization for Learning with Big Data

More on Bayesian Inference

- Model Criticism: Allows us to have checks and balances on model fit.
- Bayesian Nonparametrics: Infinite dimensional priors for data, use more parameters as you get more data
- Tutorials on Bayesian Nonparametrics

MAP Estimation

- In MRFs, may be framed as a combinatorial optimization problem
- Can use black-box solvers : Gurobi, CPLEX or leverage problem specific solvers

Supervised Learning

- Logistic Regression:
  - With the right features you can do very well with logistic regression
  - Used universally.
- Deep Learning:
  - Learnable hierarchical representations
  - Encode domain knowledge during learning
  - Can build powerful density estimators for modalities like images, text and sound
  - MLP, ConvNets, ResNets, StochasticNets, DenseNets ...the list goes on.

So where does that leave us?

- Congratulations! You have all now begun your journey into the world of machine learning
- There is still *a lot* more to learn

Keep studying machine learning?

- Consider applying to a PhD program in machine learning!
- Do research with professors here at NYU
- Advice for Graduate Students : Kevin Murphy
- Career Advice by Terence Tao