CSC412/2506
Probabilistic Learning and Reasoning

Introduction
Today

• Course information

• Overview of ML with examples

• Ungraded, anonymous background quiz

• **Thursday**: Basics of ML vocabulary (cross-validation, objective functions, overfitting, regularization) and basics of probability manipulation
Course Website

- www.cs.toronto.edu/~duvenaud/courses/csc412

- Contains all course information, slides, etc.
Evaluation

• Assignment 1: due Feb 10th worth 15%
• Assignment 2: due March 3rd worth 15%
• Assignment 3: due March 24th worth 20%
• 1-hour Midterm: Feb 23rd worth 20%
• Project: due April 10th worth 30%
• 15% per day of lateness, up to 4 days
Related Courses

• CSC411: List of methods, (K-NN, Decision trees), more focus on computation

• STA302: Linear regression and classical stats

• ECE521: Similar material, more focus on computation

• STA414: Mostly same material, slightly more introductory, more emphasis on theory than coding, exam instead of project

• CSC321: Neural networks - about 30% overlap
Textbooks + Resources

• No required textbook


• Kevin Murphy (2012), *Machine Learning: A Probabilistic Perspective*.

• Trevor Hastie, Robert Tibshirani, Jerome Friedman (2009) *The Elements of Statistical Learning*


• Deep Learning (2016) Goodfellow, Bengio, Courville
Stats vs Machine Learning

- Statistician: Look at the data, consider the problem, and design a model we can understand
  - Analyze methods to give guarantees
  - Want to make few assumptions
- ML: We only care about making good predictions!
  - Let’s make a general procedure that works for lots of datasets
  - No way around making assumptions, let’s just make the model large enough to hopefully include something close to the truth
  - Can’t use bounds in practice, so evaluate empirically to choose model details
  - Sometimes end up with interpretable models anyways
Types of Learning

- **Supervised Learning**: Given input-output pairs \((x,y)\) the goal is to predict correct output given a new input.

- **Unsupervised Learning**: Given unlabeled data instances \(x_1, x_2, x_3\ldots\) build a statistical model of \(x\), which can be used for making predictions, decisions.

- **Semi-supervised Learning**: We are given only a limited amount of \((x,y)\) pairs, but lots of unlabeled \(x\)’s.

- All just special cases of estimating distributions from data: \(p(y|x)\), \(p(x)\), \(p(x, y)\).

- **Active learning and RL**: Also get to choose actions that influence future information + reward. Can just use basic decision theory.
Finding Structure in Data

\[ P(x) = \frac{1}{Z} \sum_h \exp \left[ x^\top W_h \right] \]

Vector of word counts on a webpage

Latent variables: hidden topics

804,414 newswire stories
Matrix Factorization

Hierarchical Bayesian Model

Rating value of user $i$ for item $j$

Latent user feature (preference) vector

Latent item feature vector

\[ r_{ij} | u_i, v_j, \sigma \sim \mathcal{N}(u_i^T v_j, \sigma^2), \]

\[ u_i | \sigma_u \sim \mathcal{N}(0, \sigma_u^2 I), \quad i = 1, \ldots, N. \]

\[ v_j | \sigma_v \sim \mathcal{N}(0, \sigma_v^2 I), \quad j = 1, \ldots, M. \]

Latent variables that we infer from observed ratings.

Prediction: predict a rating $r^*_{ij}$ for user $i$ and query movie $j$.

\[ P(r^*_{ij} | R) = \int \int P(r^*_{ij} | u_i, v_j)P(u_i, v_j | R)du_idv_j \]

Posterior over Latent Variables

Infer latent variables and make predictions using Bayesian inference (MCMC or SVI).
Finding Structure in Data

Collaborative Filtering/
Matrix Factorization/
Product Recommendation

Netflix dataset:
480,189 users
17,770 movies
Over 100 million ratings.

Learned `"genre"

Fahrenheit 9/11
Bowling for Columbine
The People vs. Larry Flynt Canadian
Bacon
La Dolce Vita

Independence Day
The Day After Tomorrow
Con Air
Men in Black II
Men in Black

Friday the 13th
The Texas Chainsaw Massacre
Children of the Corn
Child's Play
The Return of Michael Myers

• Part of the winning solution in the Netflix contest (1 million dollar prize).
Impact of Deep Learning

- Speech Recognition
- Computer Vision
- Recommender Systems
- Language Understanding
- Drug Discovery and Medical Image Analysis
Multimodal Data

mosque, tower, building, cathedral, dome, castle

kitchen, stove, oven, refrigerator, microwave

ski, skiing, skiers, snowmobile

bowl, cup, soup, cups, coffee

beach

snow
Caption Generation

- A car is parked in the middle of nowhere.
- A wooden table and chairs arranged in a room.
- There is a cat sitting on a shelf.
- A ferry boat on a marina with a group of people.
- A little boy with a bunch of friends on the street.
Density estimation using Real NVP. Ding et al, 2016
Pixel Recurrent Neural Networks
Aaron van den Oord, Nal Kalchbrenner, Koray Kavukcuoglu
Density estimation using Real NVP. Ding et al, 2016
Course Themes

• Start with a simple model and add to it
  • Linear regression or PCA is a special case of almost everything
• A few ‘lego bricks’ are enough to build most models
  • Gaussians, Categorical variables, Linear transforms, Neural networks
  • The exact form of each distribution/function shouldn’t matter much
  • Your model should have a million parameters in it somewhere (the real world is messy!)
• Model checking is hard and important
  • Learning algorithms are especially hard to debug
Computation

• Later assignments will involve a bit of programming. Can use whatever language you want, but Python + Numpy is recommended.

• For fitting and inference in high-dimensional models, gradient-based methods are basically the only game in town

• Lots of methods conflate model and fitting algorithm, we will try to separate these
ML as a bag of tricks

Fast special cases:

• K-means
• Kernel Density Estimation
• SVMs
• Boosting
• Random Forests
• K-Nearest Neighbors

Extensible family:

• Mixture of Gaussians
• Latent variable models
• Gaussian processes
• Deep neural nets
• Bayesian neural nets
• ??
Regularization as a bag of tricks

Fast special cases:

- Early stopping
- Ensembling
- L2 Regularization
- Gradient noise
- Dropout
- Expectation-Maximization

Extensible family:

- Stochastic variational inference
A language of models

• Hidden Markov Models, Mixture of Gaussians, Logistic Regression

• These are simply “sentences” - examples from a language of models.

• We will try to show larger family, and point out common special cases.

Courtesy of Matthew Johnson
AI as a bag of tricks

Russel and Norvig’s parts of AI:
- Machine learning
- Natural language processing
- Knowledge representation
- Automated reasoning
- Computer vision
- Robotics

Extensible family:
- Deep probabilistic latent-variable models + decision theory
Advantages of probabilistic latent-variable models

• **Data-efficient learning** - automatic regularization, can take advantage of more information

• **Compose-able models** - e.g. incorporate data corruption model. Different from composing feedforward computations

• **Handle missing + corrupted data** (without the standard hack of just guessing the missing values using averages).

• **Predictive uncertainty** - necessary for decision-making

• **conditional predictions** (e.g. if brexit happens, the value of the pound will fall)

• **Active learning** - what data would be expected to increase our confidence about a prediction

• Cons:
  
  • intractable integral over latent variables

• Examples: medical diagnosis, image modeling
Probabilistic graphical models

+ structured representations
+ priors and uncertainty
+ data and computational efficiency
- rigid assumptions may not fit
- feature engineering
- top-down inference

Deep learning

- neural net “goo”
- difficult parameterization
- can require lots of data
+ flexible
+ feature learning
+ recognition networks
Gentlemen, our learner overgeneralizes because the VC-Dimension of our Kernel is too high. Get some experts and minimize the structural risk in a new one. Rework our loss function, make the next kernel stable, unbiased and consider using a soft margin.
The unreasonable easiness of deep learning

- Recipe: define an objective function (i.e. probability of data given params)
- Optimize params to maximize objective
- Gradients are computed automatically, you just define model by some computation
- Show demo here
Differentiable models

- Model distributions implicitly by a variable pushed through a deep net:
  \[ y = f_\theta(x) \]

- Approximate intractable distribution by a tractable distribution parameterized by a deep net:
  \[ p(y|x) = \mathcal{N}(y|\mu = f_\theta(x), \Sigma = g_\theta(x)) \]

- Optimize all parameters using stochastic gradient descent
Modeling idea: graphical models on latent variables, neural network models for observations

unsupervised learning

supervised learning

Courtesy of Matthew Johnson
Learning outcomes

• Know standard algorithms (bag of tricks), when to use them, and their limitations. For basic applications and baselines.

• Know main elements of language of deep probabilistic models (distributions, expectations, latent variables, neural networks) and how to combine them. For custom applications + research

• Know standard computational tools (Monte Carlo, Stochastic optimization, regularization, automatic differentiation). For fitting models
Tentative list of topics

- Linear methods for regression + classification, Bayesian linear regression
- Probabilistic Generative and Discriminative models, Regularization methods
- Stochastic Optimization (practically important)
- Neural Networks
- Model Comparison and marginal likelihood (conceptually important)
- Stochastic Variational Inference
- Time series and recurrent models
- Mixture Models, Graphical Models and Bayesian Networks
- Kernel Methods, Gaussian processes, Support Vector Machines
Quiz
Machine-learning-centric History of Probabilistic Models

- **1940s - 1960s** Motivating probability and Bayesian inference
- **1980s - 2000s** Bayesian machine learning with MCMC
- **1990s - 2000s** Graphical models with exact inference
- **1990s - present** Bayesian Nonparametrics with MCMC (Indian Buffet process, Chinese restaurant process)
- **1990s - 2000s** Bayesian ML with mean-field variational inference
- **2000s - present** Probabilistic Programming
- **2000s - 2013** Deep undirected graphical models (RBMs, pretraining)
- **2010s - present** Stan - Bayesian Data Analysis with HMC
- **2000s - 2013** Autoencoders, denoising autoencoders
- **2000s - present** Invertible density estimation
- **2013 - present** Stochastic variational inference, variational autoencoders
- **2014 - present** Generative adversarial nets, Real NVP, Pixelnet
- **2016 - present** Lego-style deep generative models (attend, infer, repeat)