

# CSE 6740 Lecture 23

## *How Do I Apply All This in the Real World? (Data Analysis)*

Alexander Gray

agray@cc.gatech.edu

Georgia Institute of Technology

# Quiz

1. For large datasets, dual-tree is faster than single-tree. T.
2. Multipole methods are a special case of the generalized  $N$ -body algorithm. T.
3. Some GNPs can be computed exactly with tree-based methods. T.

# Today

1. Solving a Real Problem with ML
2. Some Big Open Problems in ML
3. Your ML Experience/Interests

# Solving a Real Problem with ML

Some steps to consider.

# Understand the Problem/Client

- Understand the real aims of the client
- Why is ML going to help here?
- Understand the prior knowledge/assumptions that can help

# Understand the Problem/Client

- Understand the general constraints of the client
  - accuracy
  - speed
  - confidence estimate
  - interpretability
  - costs of different errors
- Identify the ML task – possibly try several

# Understand the Data

- Get (good) data from the client (often most time-consuming step)
- Get a feel for the structure of the data (*exploratory data analysis*)
  - plots of various kinds
  - unsupervised learning (clusters, associations, manifolds)
  - information visualization techniques

# Design for the Data: Modeling

- Non-negativity (images, text): e.g. model with multinomials
- Multi-modality (clusters): e.g. model with mixture
- Temporal structure: e.g. model with Markov model
- Spatial structure: e.g. model with spatial kernel
- Invariant structure: e.g. model with tangent distance

# Design for the Data: Sensor Fusion

- Features of various scales: e.g. renormalize by standard deviation
- Mixed discrete and continuous: e.g. use mixed probabilistic model
- Highly-varying scale within a feature: e.g. use log transformation

# Design for the Data: Non-Ideal Data

- Outliers: e.g. use robust loss
- Systematic errors or effects: e.g. model them directly
- Missing values: e.g. use EM fill in values (*imputation*)
- Imbalanced classes: e.g. use balanced resampling
- Values have varying (known) quality (*measurement errors*): e.g. model with convolutions

# Design for the Data: Non-Ideal Data

- Systematic omission/truncation of data (*censoring*): model the censoring mechanism
- Data not drawn from the full distribution (*selection bias*): e.g. estimate selection mechanism and correct
- Test distribution differs from train distribution (*covariate shift, concept drift*): e.g. reweight data in cross-validation

# Design for the Deployment

- Few labeled data, many unlabeled: e.g. use *semi-supervised* learning
- Data are expensive but you can get more: e.g. learn which is most useful to obtain (*active learning*)
- Receive one datum at a time (*online, incremental*): e.g. stochastic gradient descent
- Dataset is too large for one machine: e.g. distributed learning
- Learning to be done on a power-limited device: e.g. method with very low computation

# Evaluate the Results

- Run it and evaluate
  - Evaluate confidence bands: e.g. look at *coverage*
  - Interpret predictive model: e.g. use decision tree rules
  - Look at error tradeoffs: e.g. look at *ROC curves*, *cost curves*
  - Compare various predictors: e.g. use *hypothesis testing*
- Iterate with client

# Some Big Open Problems in ML

Want a thesis topic?

# Some ML Problems

- Non-vector data: e.g. relational
- Good theory of active learning
- Effective reinforcement learning

# ML Foundations

- Better cross-validation: e.g. natural gradient
- Optimal construction of features, distances, kernels
- Automatic robustness for complex models

# ML on Large Data

- Large-scale graphical model inference: e.g. approximate
- High-dimensional generalized  $N$ -body problems
- High-dimensional integration
- Accurate confidence bands which are also cheap
- Efficient constrained optimization with many parameters

# Real-World ML

- Complete ML software: e.g. which includes all the capabilities in this lecture, stable, fast
- Highly interpretable ML
- Complete/easy data visualization software

# Your ML Experiences/Interests

Now I want to hear from you...

# Your ML Experiences/Interests

- What did you implement? Was it hard/easy/useful?
- Which class topic do you want to know more about?
- Any guest speaker you'd like to see?