ORIE 4741: Learning with Big Messy Data

Review for Final Exam

Professor Udell
Operations Research and Information Engineering
Cornell

November 30, 2017
Outline

Learning Big Messy Data

How to study for the exam

What did we not talk about?
Review

- learning
- big
- messy
- data
Review: data

- first of all: look at it!
- are there missing values?
- decide what you want to learn or predict
- input space $\mathcal{X}$, output space $\mathcal{Y}$
  - real, boolean, nominal, ordinal, text, …
Review: messy

- probabilistic model: $(x, y) \sim P(x, y)$
- deterministic model: $y = f(x)$
- additive noisy model: $y = f(x) + \varepsilon$
  - additive noise model makes no sense for non-real data types (boolean, ordinal, nominal)
Review: messy features

- feature engineering
  - can convert other data to real valued features
  - enables easy fitting of complex nonlinear models
- unsupervised learning
  - can use to fill in missing data
  - can use to reduce dimensionality of feature space
- regularization
  - can reduce sensitivity of estimate to corrupted or novel feature values
Review: messy labels

- robust loss functions
  - huber, $\ell_1$, quantile
- loss functions for classification
  - hinge, logistic
- loss functions for ordinals and nominals
  - by learning a vector
  - by learning probabilities
Review: learning

- view data as samples from $P(x, y)$
- goal is to learn $f: \mathcal{X} \rightarrow \mathcal{Y}$
- complex models fit both data and noise better
- diagnose underfitting vs overfitting
- generalization: how do we know if we’re overfitting?
  - bootstrap: how big are the error bars?
  - (cross)validate: how big are the out-of-sample errors?
  - compute error on test set + use Hoeffding bound
  - posit a probabilistic model + use bias variance tradeoff
- correcting overfitting: regularize or add data
- correcting underfitting: add new (engineered or measured) features
Review: learning a mapping

how to learn $f : \mathcal{X} \rightarrow \mathcal{Y}$?

- using a greedy iterative procedure, like the perceptron method or decision trees
- by regularized empirical risk minimization
  - minimizing some loss function to fit the data
  - + some regularizer to generalize well
Review: learning by optimizing

how to solve an optimization problem?

▶ is it smooth?
  ▶ set the gradient to 0; solve resulting system of equations
▶ does it have a differentiable part and a part with an easy prox function?
  ▶ use proximal gradient
▶ does it have a subdifferentiable part and a part with an easy prox function?
  ▶ use proximal gradient with decreasing step sizes
Review: big

- algorithms for big data should be **linear** in the number of samples $n$
- three big data algorithms for least squares:
  - gradient descent ($O(nd)$ per iteration)
  - QR ($O(nd^2)$)
  - SVD ($O(nd^2)$) (mostly used as analysis tool)
- proximal gradient
  - $O(nd)$ per iteration
- parallelize!
  - gradient calculation
  - Gram matrix calculation
Review: unsupervised

- use unsupervised learning to
  - reduce dimensionality
  - reduce noise in data
  - fill in missing entries
  - plot or cluster messy high-dimensional features
  - find vector representation for nominal features

- Generalized low rank models
  - includes PCA, \( k \)-means, non-negative matrix factorization, matrix completion, \ldots
  - can fit using alternating proximal gradient method
Outline

Learning Big Messy Data

How to study for the exam

What did we not talk about?
How to study for the exam

- reread your notes **and** the lecture slides
How to study for the exam

- reread your notes and the lecture slides
- make up exam questions
  - what do you want to remember from this class?
  - what will you need to remember to learn from big messy data effectively?
  - what will you need to remember to learn from big messy data without learning things that are false?
  - then ask how to do these and why these methods work.
  - do these out loud, with a friend.
How to study for the exam

▶ reread your notes and the lecture slides
▶ make up exam questions
  ▶ what do you want to remember from this class?
  ▶ what will you need to remember to learn from big messy data effectively?
  ▶ what will you need to remember to learn from big messy data without learning things that are false?
▶ ask when. e.g.,
  ▶ when to use a decision tree vs linear model?
  ▶ when should we use different losses and regularizers?
  ▶ when should we cross validate?
  ▶ when to bootstrap?
  ▶ when to use NNMF? When PCA?
  ▶ when would you want sparsity?
How to study for the exam

- reread your notes and the lecture slides
- make up exam questions
  - what do you want to remember from this class?
  - what will you need to remember to learn from big messy data effectively?
  - what will you need to remember to learn from big messy data without learning things that are false?
- ask when. e.g.,
  - when to use a decision tree vs linear model?
  - when should we use different losses and regularizers?
  - when should we cross validate?
  - when to bootstrap?
  - when to use NNMF? When PCA?
  - when would you want sparsity?
- then ask how to do these and why these methods work.
How to study for the exam

- reread your notes and the lecture slides
- make up exam questions
  - what do you want to remember from this class?
  - what will you need to remember to learn from big messy data effectively?
  - what will you need to remember to learn from big messy data without learning things that are false?
- ask when. e.g.,
  - when to use a decision tree vs linear model?
  - when should we use different losses and regularizers?
  - when should we cross validate?
  - when to bootstrap?
  - when to use NNMF? When PCA?
  - when would you want sparsity?
- then ask how to do these and why these methods work.

do these out loud, with a friend
Outline

Learning Big Messy Data

How to study for the exam

What did we not talk about?
What did we not talk about? I

more learning techniques

- nearest neighbors and exemplar methods
- fancier decision trees: random forests (bagging), gradient boosted trees (boosting)
- fancier feature engineering: neural nets, deep learning

resources:

- CS machine learning class next semester (Prof. Weinberger)
- Hastie, Tibshirani and Friedman, “Statistical Learning”
- ScikitLearn
- distributed ML frameworks: H2O, TensorFlow, MLLib, . . .
- ML conferences: NIPS, ICML, AAAI, . . .
What did we not talk about? II

more optimization techniques

- more stable, reliable methods, e.g., interior point
- more parallel distributed methods, e.g., ADMM
- more iterative methods for huge data, e.g., SGD, ADAM, AdaGrad

resources:

- Convex Analysis next semester (Prof. Lewis)
- CS machine learning class next semester (Prof. Weinberger)
What did we not talk about? III

more domain specific techniques

- text (e.g., hashing)
- images, video
- time series
- signal processing
- speech
What did we not talk about? IV

- causal inference
- bias and fairness
- automatic machine learning
- model interpretability vs accuracy
To do for remainder of semester

- project due 11:59pm 12-4
- final exam 9am 12-6
- project peer reviews due 11:59pm 12-10
- fill out course evaluation