Learning **for** Big Data

Hsuan-Tien Lin (林軒田)
htlin@csie.ntu.edu.tw

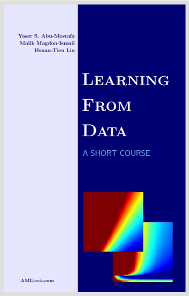
Department of Computer Science
& Information Engineering

National Taiwan University

slightly modified from my keynote talk
in IEEE BigData 2015 Taipei Satellite Session

About the Title

- “Learning **for** Big Data”
  —my wife: you have made a **typo**
- do you mean “Learning **from** ~~Big~~ Data”?
  —no, not a **shameless sales campaign**
  for my co-authored **best-selling** book ☺
  (`http://amlbook.com`)



as machine learning researcher

machine learning **for** big data
—easy?! ☺

as machine learning educator

human learning **for** big data
—**hard!!**

will focus on **human** learning **for** big data

Human Learning for Big Data

Todo

- some FAQs that I have encountered as
  - **educator** (NTU and NTU@Coursera)
  - **team mentor** (KDDCups, TSMC Big Data competition, etc.)
  - **researcher** (CLLab@NTU)
  - **data scientist** (Appier), a AI-based startup
- my imperfect yet **honest** answers that hint **what shall be learned**

First Honest Claims

- must-learn for **big data** $\approx$ must-learn for **small data** in ML, but the former with **bigger seriousness**
- system design/architecture **very important**, but somewhat beyond my pay grade

I wish I had an answer to that
because I'm tired of answering that question.
—Yogi Berra (Athlete) ☺

Big Data FAQs (1/4)

how to ask good questions from (my precious big) data?

My Polite Answer

good start already ☺, any more thoughts that you have in mind?

My Honest Answer

I don't know.

or a slightly longer answer:
if you don't know, I don't know.

A Similar Scenario

how to ask good questions from
(my precious big) data?

how to find a research topic for my thesis?

My Polite Answer

good start already ☺, any more thoughts that you have in mind?

My Honest Answer

I don't know.

or a slightly longer answer:
I don't know, but perhaps you can **start** by thinking about **motivation** and **feasibility**.

Finding (Big) Data Questions $\approx$ Finding Research Topics

- **motivation**: what are you interested in?
- **feasibility**: what can or cannot be done?

motivation

- something publishable? oh, possibly **just for people in academia** ☺
- something that **improves xyz performance**
- something that inspires deeper study

—helps **generate** questions

feasibility

- **modeling**
- **computational**
- budget
- timeline
- . . .

—helps **filter** questions

brainstorm from **motivation**;
rationalize from **feasibility**

Finding **Big** Data Questions

generate questions from motivation

- variety: **dream more** in big data age
- velocity: evolving data, **evolving questions**

filter questions from feasibility

- volume: **computational** bottleneck
- veracity: **modeling** with **non-textbook** data

almost never find right question in your **first try**
—good questions come **interactively**

Interactive Question-Asking from Big Data: Our KDDCup 2011 Experience (1/2)

Recommender System

- **data**: how users have rated movies
- **goal**: predict how a user would rate an unrated movie

A Hot Problem

- competition held by Netflix in 2006
  - 100,480,507 ratings that 480,189 users gave to 17,770 movies
  - 10% improvement = **1 million dollar prize**
- similar competition (movies $\rightarrow$ songs) held by Yahoo! in KDDCup 2011, the most **prestigious data mining competition**
  - 252,800,275 ratings that 1,000,990 users gave to 624,961 songs

National Taiwan University got two **world champions** in KDDCup 2011—with Profs. Chih-Jen Lin, Shou-De Lin, and many students.

Interactive Question-Asking from Big Data: Our KDDCup 2011 Experience (2/2)

Q1 (pre-defined): can we improve rating prediction of (user, song)?

Q1.1 after **data analysis**:
two types of users, **lazy** 7% (same rating always) & **normal**
—if a user gives 60, 60, . . . during training, how'd she rate next item?

same (80%)	different (20%)

Q1.1.1: can we **distinguish 80%** using other features?

. . .

—**failed** (something you normally wouldn't see in paper ☺)

Q1.2 after **considering domain knowledge**: test data are **newer logs**
—shall we emphasize newer logs in training data?

Q1.2.1: can we just give each log different **weight**? (but how?)

Q1.2.2: can we **tune optimization** to effectively emphasize newer logs? (**yes this worked** ☺)

our KDDCup experience: **interactive** (**good or bad**) **question-asking** kept us going!

Learning to Ask Questions from Big Data

Must-learn Items

- true interest for **motivation**
  —big data don't generate questions, **big interests do**
- capability of machines (when to use ML?) for **feasibility**

Taught in ML Foundations on NTU@Coursera

1. exists underlying **pattern** to be learned
2. **no easy/programmable definition** of pattern
3. having data **related to** pattern

  —ML **isn't cure-all**

- research cycle for **systematic steps**
  —a **Ph.D.** or serious research during M.S./undergraduate study

Computers are useless. They can only give you answers.—Pablo Picasso (Artist)

Big Data FAQs (2/4)

what is the best machine learning model for (my precious big) data?

My Polite Answer

the best model is **data-dependent**, let's **chat about your data first**

My Honest Answer

I don't know.

or a slightly longer answer:
I don't know about **best**, but perhaps you can **start** by thinking about **simple models**.

Sophisticated Model for Big Data

what is the best machine learning model for (my precious big) data?

what is the **most sophisticated** machine learning model for (my precious big) data?

- myth: my **big data** work best with **most sophisticated** model
- partially true: deep learning for image recognition @ Google —**10 million images** on **1 billion internal weights**

  (Le et al., Building High-level Features Using Large Scale Unsupervised Learning, ICML 2012)

Science must begin with myths,
*and with the **criticism of myths**.*
—Karl Popper (Philosopher)

Criticism of Sophisticated Model

myth: my **big data** work best
with **most sophisticated** model

Sophisticated Model

- time-consuming to **train** and **predict**
  —often **mismatch** to big data
- difficult to **tune** or **modify**
  —often **exhausting** to use
- point of **no return**
  —often **cannot “simplify” nor “analyze”**

sophisticated model shouldn’t be
first-choice for big data

Linear First (1/2)

what is the *first* machine learning model for (my precious big) data?

Taught in ML Foundations on NTU@Coursera

linear model (or simpler) first:

- efficient to **train** and **predict**, e.g. *(Lin et al., Large-scale logistic regression and linear support vector machines using Spark. IEEE BigData 2014)*

  —my favorite in Appier
- easy to **tune** or **modify**

  —key of our **KDDCup winning solutions** in 2010 (educational data mining) and 2012 (online ads)

Linear First (2/2)

what is the first machine learning model for (my precious big) data?

Taught in ML Foundations on NTU@Coursera

linear model (or simpler) first:

- somewhat **"analyzable"**
  —my students' winning choice in TSMC Big Data Competition (just old-fashioned **linear regression**! ☺)
- little **risk**
  - if linear good enough, **live happily thereafter** ☺
  - otherwise, try something more complicated, with **theoretically nothing lost** except "wasted" computation

My KISS Principle:
Keep It Simple, ~~Stupid~~ ***Safe***

Learning to Start Modeling for Big Data

Must-learn Items

- **linear** models, especially
  - how to **tune** them
  - how to **interpret** their outcomes
- decision tree (or perhaps even better, **Random Forest**) as a KISS **non-linear** model

*An explanation of the data should be made **as simple as possible**, but no simpler.—[?] Albert Einstein (Scientist)*

Big Data FAQs (3/4)

how should I improve ML performance with (my precious big) data?

My Polite Answer

do we have **domain knowledge** about your problem?

My Honest Answer

I don't know.

or a slightly longer answer:
I don't know for sure, but perhaps you can **start** by encoding your **human intelligence/knowledge**.

A Similar Scenario

how should I improve ML performance with (my precious big) data?

how should I improve the performance of my classroom students?

instructor teaching $\equiv$ student learning

- teach more **concretely** $\longrightarrow$ better performance
- teach more **professionally** $\longrightarrow$ better performance
- teach more **key** points/aspects $\longrightarrow$ better performance

to improve learning performance,
you should perhaps **teach better**

Teaching Your Machine Better with Big Data

- **concrete**:
  good research questions, as discussed ☺
- **professional**:
  embed **domain knowledge** during data **construction**
- **key**:
  facilitate your learner using proper data **pruning/cleaning/hinting**

IMHO, data **construction** is more important for big data than machine learning is

Your Big Data Need Further **Construction**

Big Data Characteristics

many fields, and many abstract ones

Our KDDCup 2010 Experience

educational data mining

(Yu et al., Feature Engineering and Classifier Ensemble for KDD Cup 2010)

- *"Because all feature meanings are available, we are able to manually identify some useful pairs of features ..."*:
  - domain knowledge: "**student** s **does step** i **of problem** j **in unit** k"
  - hierarchical encoding: **[has student** s **tried unit** k**]** more meaningful than [has student s tried step i]
- *"Correct First-Attempt Rate"* c_j of each problem j:
  - domain knowledge: $c_j \approx$ hardness
  - condensed encoding: c_j **physically more meaningful** than j

feature engineering: make your (feature) data **concrete** by embedding **domain knowledge**

Learning to Construct Features for Big Data

Must-learn Items

- **domain knowledge**
  - if available, great!
  - if not, start by **analyzing** data first, not by **learning from data** —correlations, co-occurrences, informative parts, frequent items, etc.
- common **feature construction** techniques
  - encoding
  - combination
  - importance estimation: linear models and Random Forests especially useful (**simple models, remember?** ☺)

one secret in winning KDDCups:
ask **interactive questions** (remember?)
that allows encoding **human intelligence**
into **feature construction**

Big Data FAQs (4/4)

how should I escape from the unsatisfactory test performance on (my precious big) data?

My Step by Step Diagnosis

if (training performance okay) **[$>$ 90% of the time]**

- combat **overfitting**
- correct training/testing **mismatch**
- check for **misuse**

else

- **construct better features** by asking more questions, remember? ☺
- now you can try more **sophisticated models**

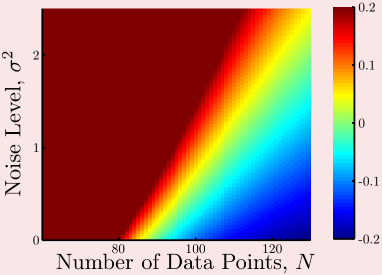
will focus on the **first part**

Combat Overfitting (1/2)

myth: my **big data** is so big that overfitting is impossible

- no, big data usually **high-dimensional**
- no, big data usually **heterogeneous**
- no, big data usually **redundant**
- no, big data usually **noisy**

Overfitting Hazard



(Learning from Data book)

data-size-to-noise ratio is what matters!

big data still require
careful treatment of overfitting

Combat Overfitting (2/2)

Driving Analogy of Overfitting

learning	driving
overfit	commit a car accident
sophisticated model	“drive too fast”
noise	bumpy road
limited data size	limited observations about road condition

—**big data** only cross out **last line**

Regularization

regularization	put brake

—important to know **where the brake is**

Validation

validation	monitor dashboard

—important to ensure **correctness**

Overfitting is real, and here to stay.—Learning from Data (Book)

Correct Training/Testing Mismatch

A True Personal Story

- Netflix competition for movie recommender system:
  10% **improvement = 1M US dollars**
- on my own validation data, first shot, showed 13% improvement
- **why am I still here?** ☺
  validation: random examples within data;
  test: "last" user records "after" data

Technical Solutions

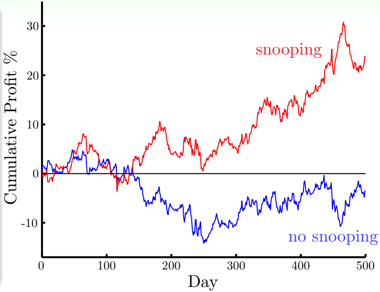
practical rule of thumb: **match test scenario as much as possible**

- training: emphasize later examples (KDDCup 2011)
- validation: use "late" user record

If the data is sampled in a biased way, learning will produce a similarly biased outcome.—Learning from Data (Book)

Biggest Misuse of Machine Learning: Data Snooping

- 8 years of currency trading data
- first 6 years for **training**, last two 2 years for **testing**
- feature = previous 20 days, label = 21th day
- snooping versus no snooping: superior profit possible



- snooping: shift-scale all values by training + testing
- no snooping: shift-scale all values by training only

Data Snooping by Data Reusing

Data Snooping by Data Reusing: Research Scenario

with my precious data

- paper 1: propose algorithm 1 that works well on data
- paper 2: find room for improvement, propose algorithm 2 —and **publish only if better** than algorithm 2 on data
- paper 3: find room for improvement, propose algorithm 3 —and **publish only if better** than algorithm 2 on data
- . . .

- if all papers from the same author in **one big paper**: *as if* using a super-sophisticated model that includes algorithms 1, 2, 3, . . .
- step-wise: later author **snooped** data by reading earlier papers, bad generalization worsen by **publish only if better**

If you torture the data long enough, it will confess.—Folklore in ML/DM ☺

Avoid Big Data Snooping

data snooping $\Longrightarrow$ human overfitting

Honesty Matters

- **very hard to avoid** data snooping, unless being extremely honest
- extremely honest: **lock your test data in safe**
- less honest: **reserve validation and use cautiously**

Guidelines

- be blind: avoid **making modeling decision by data**
- be suspicious: interpret findings (including your own) by proper **feeling of contamination**—keep your data **fresh** if possible

one last secret to winning KDDCups:
"art" to carefully balance between
data-driven modeling (snooping) &
validation (no-snooping)

Learning to Escape Traps for Big Data

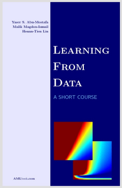
Must-learn Items

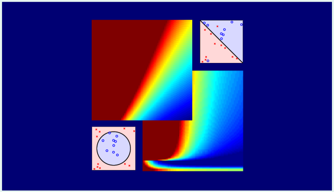
- **combat overfitting**: regularization and validation
- **correct training/testing mismatch**: philosophy and perhaps some heuristics
- **avoid data snooping**: philosophy and research cycle (remember? ☺)

happy **big data learning**! ☺

Summary

- **human must-learn** ML topics for big data:
  - procedure: research cycle
  - tools: simple model, feature construction, overfitting elimination
  - sense: philosophy behind machine learning
- **foundations** even more important in big data age
  —now a **shameless sales campaign** for my co-authored book and online course ☺





—special thanks to Prof. Yuh-Jye Lee and Mr. Yi-Hung Huang for suggesting materials

Thank you!

Appendix: ML Foundations on NTU@Coursera

When can machines learn?

- L1: the learning problem (☺)
- L2: learning to answer yes/no (☺)
- L3: types of learning (☺)
- L4: feasibility of learning

Why can machines learn?

- L5: training versus testing
- L6: theory of generalization
- L7: the VC dimension (☺)
- L8: noise and error

How can machines learn?

- L9: linear regression (☺)
- L10: logistic regression (☺)
- L11: linear models for classification (☺)
- L12: nonlinear transformation (☺)

How can machines learn better?

- L13: hazard of overfitting (☺)
- L14: regularization (☺)
- L15: validation (☺)
- L16: three learning principles (☺)

☺ $\approx$ must-learn