some parts based on Lin, Madgon-Ismail, and Abu-Mostafa. Teaching machine learning to a diverse audience: the foundation-based approach. Teaching Machine Learning Workshop @ ICML ’12.
About Me

Hsuan-Tien Lin

- Chief Data Scientist, Appier
- Professor, Dept. of CSIE, National Taiwan University
- Co-author of textbook “Learning from Data: A Short Course”
- Instructor of the NTU-Coursera Mandarin-teaching ML Massive Open Online Courses
  - “Machine Learning Foundations”: www.coursera.org/course/ntumltone
  - “Machine Learning Techniques”: www.coursera.org/course/ntumltwo
Diversity in ML classes

NTU ML 2011 Fall (77 students)

- **background diversity**
- **“maturity” diversity**
  - junior: 8
  - senior: 20
  - master: 44
  - phd: 5
- similarly diverse in RPI and in Caltech (online course)

**challenge:**
serving CS students while accommodating the needs of diverse non-CS audience

mindset of the audience?

1http://work.caltech.edu/telecourse
Observed Mindsets of the Diverse Audience

- highly **motivated** to learn—not satisfied with only shallow comic-book stories
- often with **minimum but non-empty** math/programming background—capable of downloading and trying the latest packages

words of a student from industry (Caltech online course 2012)

While it’s easy to pick up a couple of algorithms from the many text-books and online materials out there, it is the solid foundation, both mathematical and practical, as well as this better intuition that I would have missed studying alone without this class. Also, the cadence of the lectures and the demand: **solid foundation** (+ better intuition)!
Our Proposed Teaching Approach

- foundation-based, and foundation-first
- then, compensate foundation with a couple of useful algorithms/techniques

Comparison to techniques-based

- techniques-based: hops through the forest of many latest and greatest techniques
- foundation-based: illustrate the map (core) first to prevent getting lost in the forest

foundation-based: prepare students for easy learning of untaught/future techniques
Our Proposed Teaching Approach [Cont.]

- foundation-based, and foundation-first
- then, compensate foundation with a couple of useful algorithms/techniques

comparison to foundation-later

- foundation-later:
  - first, techniques to raise interests
  - then, foundations to consolidate understanding
- foundation-first: build the basis (core) first to perceive the techniques from the right angle

foundation-first: let students know when and how to use the powerful tools before getting addicted on the power
Our Proposed Foundation: Three Concepts

understand learnability, approximation and generalization
- when can we learn and what are the tradeoffs?
- conducting machine learning properly

use simple models first
- the linear model coupled with some nonlinear transforms is typically enough for most applications
- conducting machine learning safely

deal with noise and overfitting carefully
- how to tackle the “dark side” of learning?
- conducting machine learning professionally

our experience: worth starting with those foundations, even for a diverse audience
learnability, approximation & generalization — conducting machine learning *properly*

- good learning (test performance) = good approximation (training performance) + good generalization (complexity penalty)

- a must-teach key message
- can be illustrated in different forms (e.g. VC bound, bias-variance, even human-learning philosophy)
- make learning non-trivial and fascinating to students
learnability, approximation & generalization — conducting machine learning *properly* [Cont.]

**Wrong use of learning (beginner’s mistakes)***
- Ensure **good approximation**, pray for **good generalization**
  — praying for something out-of-control

**Right use of learning***
- Ensure **good generalization**, try best for **good approximation**
  — trying something possibly in-control

*We cannot guarantee learning. We can “guarantee” no disasters.* That is, after we learn we will either declare success or failure, and in both cases we will be right.
linear models
—conducting machine learning safely

linear models
= good generalization
with established optimization tools for good approximation

- after knowing approximation/generalization: a good stage for learning safe techniques
- sufficiently useful for many practical problems (Yuan et al., 2012)
- building block in sophisticated techniques through feature transforms
- make learning concrete to students
linear models — conducting machine learning safely [Cont.]

Wrong use of learning (beginner’s mistakes)

Start with the “greatest” techniques first — a point of no return

Right use of learning

Start with the simplest techniques first — and yes, it can work well

A rich and representative family of linear techniques

- Classification: approx. combinatorial optimization (perceptron-like)
- Regression: analytic optimization (pseudo-inverse)
- Logistic regression: iterative optimization (SGD)

Students coming from diverse backgrounds not only get the big picture, but also the finer details in a concrete setting.
deal with noise and overfitting
—conducting machine learning **professionally**

- overfit = difficult to ensure good generalization/learning with **stochastic** or **deterministic noise** on finite data
- regularization = tools for further guaranteeing **good generalization**
- validation = tools for certifying **good learning**

- turn amateur students to **professionals**
- make learning **artistic** to students
deal with noise and overfitting —conducting machine learning professionally [Cont.]

wrong use of learning (beginner’s mistakes)
apply all possible techniques and choose by best approximation result —high risk of overfitting

right use of learning
apply a reasonable number of well-regularized techniques and choose by best validation result —relatively immune to noise and overfitting

Complex situations call for simpler models.
Support Vector Machine

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>generalization</td>
<td>large-margin bound</td>
</tr>
<tr>
<td>approximation</td>
<td>quadratic programming</td>
</tr>
<tr>
<td>linear model</td>
<td>basic formulation</td>
</tr>
<tr>
<td>feature transform</td>
<td>through kernel</td>
</tr>
<tr>
<td>regularization</td>
<td>large-margin</td>
</tr>
<tr>
<td>validation</td>
<td>#-SV bound</td>
</tr>
</tbody>
</table>

Neural Network

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>#-neuron bound</td>
<td></td>
</tr>
<tr>
<td>gradient decent et al.</td>
<td></td>
</tr>
<tr>
<td>neurons</td>
<td></td>
</tr>
<tr>
<td>through cascading</td>
<td>weight-decay or early-stopping</td>
</tr>
<tr>
<td>for choices in regularization</td>
<td></td>
</tr>
</tbody>
</table>

[libsvm-2.9]$ ./svm-train -t 2 -g 0.05 -c 100 heart_scale
optimization finished, #iter = 1966

Total nSV = 113

- good approximation (by choosing kernel and optimization)
- good generalization (by regularization)
- good learning (by using #SV as validation indicator)
Caltech 2012: (mixed) 7 weeks of foundations, 0.5 week of NNet, 0.5 week of RBF Net, 1 week of SVM

NTU ML (with MOOCs): (sequential) 8 weeks of foundations, 3 weeks of SVM, 3 weeks of aggregation, 2 weeks of deep learning —with an in-class data mining competition where students exploited taught/not-taught techniques with ease

often incremental efforts to teach/learn a new technique after solid foundations
Mini Summary

foundation-based, foundation-first
—works well in our experience

- Learnability: *philosophical* understanding, make learning *non-trivial*, conduct learning *properly*
- Linear models: *algorithmic* modeling, make learning *concrete*, conduct learning *safely*
- Overfitting: *practical* tuning, make learning *artistic*, conduct learning *professionally*
Excitement of Competition

史丹佛這樣教創新


「第六、鼓勵學生競賽。從來沒有一件事像「競爭」這樣，能讓人廢寢忘食、24小時工作絲毫不倦。我們鼓勵學生參加各式各樣的國際競賽，我們的學生蓋了一間太陽能屋，做電動車、機器人，參加DARPA(國防高等研究計劃署)挑戰賽，也參加企業營運書的競賽。」
Machine Learning Competition: Mini-KDD Cup

### Background

- an annual competition on KDD (knowledge discovery and data mining)
- organized by ACM SIGKDD, starting from 1997, now **the most prestigious data mining competition**
- usually lasts 3-4 months
- participants include famous research labs (IBM, AT&T) and top universities (Stanford, Berkeley)
My Design: Time Line

key dates:

- report due (i.e. overall competition end): as late as possible —often **4 days before I need to submit the scores to NTU**
- award ceremony (i.e. early competition end): usually **last class**
- announcement: best timing to be **right after midterm** —but may highly depend on TAs’ schedule
- start designing: **two or more weeks before** announcement
an interesting story makes the competition exciting!

- ML2014:

In this final project, you are going to be part of an exciting machine learning competition. Consider a startup company that features a coming product on the mobile phone. The core of the product is a robust character recognition system...... To win the prize, you need to fight for the leading positions on the score board. Then, you need to submit a comprehensive report that describes not only the recommended approaches, but also the reasoning behind your recommendations. Well, let’s get started!

- more interesting ones:
  - ML2014, ML2013: **optical character recognition**
  - ML2012: **ad click prediction** (derived from KDDCup 2012)

—often okay to **reuse with modifications**
My Design: Team Size

- most ideal team size IMHO is 3:
  - collaborative, dispute resolution, fewer free riders, etc.
  - but can also allow 4 if class size too big for the TAs to grade
- usually allow $\leq 3$:
  - so students do not have the burden to find exactly 3
  - students can flexibly break teams if needed
  - but evaluate with workloads of 3 for fairness
- still sometimes hard for some students to find team members:
  - motto: provide matching mechanism, but not force anyone to any team
- prevent free riders: need workload distribution in report
My Design: Scoreboard

- core place that makes the game **exciting**
- thanks to my TAs in all those years for creating and maintaining the service
- basically, a simple **submit-judge-scoreboard** system
- usually provide the students an additional **description** field to interact—though few use it for serious purposes
My Design: Award Ceremony

- purpose: to **add more fun**
- **light presents** (postcards, paper notebooks, etc.)
- some students list their **good-performing awards in resume**
- may serve some **educational purposes**
- in addition to good-performing awards, can also give **interesting awards**
ML2012: How Much Overfitting Can We Get?

9472 submissions from 52 teams within 1.5 months......
<table>
<thead>
<tr>
<th>team</th>
<th>scoreboard</th>
<th>hidden</th>
<th>algorithm</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimaximizer</td>
<td>0.7632</td>
<td>0.7407</td>
<td>rwa</td>
<td>2013/01/01 00:00:08</td>
</tr>
<tr>
<td>team</td>
<td>submission count</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>1097</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>anything</td>
<td>1149</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
My Design: Grade

- generally based on report, not competition, but correlated
  - too much emphasis on competition ⇒ utilitarianism
  - too little emphasis on competition ⇒ less interesting game

- ask TAs to act as “bosses”: The grading TAs would grade qualitatively with letters: A++[210], A+[196], A[186], B+[176], B[166], C+[156], C[146], D+[136], D[126], F+[116], F[76], F-[36], Z[0]

- list basic requirements corresponding to B
  - to get B, students only need to work ≈ usual homeworks
  - to get more, need more to convince the TAs

- generally “loose” about basic requirements
  —most students perform way beyond the basic requirements anyway

- generally team grade, but adjust individual grade if workload unbalanced
My Design: Loading

- ideal: a bit **harder than homework**
- estimate: 60 to 90 man-hours to finish basic requirements (**30 man-hour per member**)
- sometimes need to **adjust loading of other homeworks** —not an easy task, though
My Design: TAs

- good TAs’ help **essential**—I cannot thank them enough!
- **design, system setup, discuss with students**
My Design: TAs

always note: TAs are **busy**!!
My Design: Instructor

my main job: **heat up the competition**
my two other jobs:

- participate **seriously in the design**
- maintain **fairness** of competition
Some Summary Thoughts

**Positive Side**
- **fun** for most students, TAs and instructor
- students, TAs and instructor **learn a lot**

**Negative Side**
- **exhausting** for most students, TAs and instructor
- **can be disappointing** for some students

Questions and Discussions?