Teaching "Prediction: Machine Learning and Statistics"

Cynthia Rudin RUDIN@MIT.EDU
MIT Sloan School of Management
Massachusetts Institute of Technology
100 Main Street Cambridge MA 02142

Abstract

The course *Prediction: Machine Learning and Statistics* is taught currently at MIT to mathematically oriented non-experts. The course focuses generally on predictive modeling from data, and contains topics within data mining, machine learning, and statistics, often going back and forth between machine learning and statistical views of various algorithms and concepts. The course is structured in three parts: an overview of most of the "top 10" algorithms in data mining based on the ICDM survey (Wu et al., 2008), statistical learning theory and kernels, and Bayesian analysis. We present insights from this course.

1. Course Motivation

Here is the beginning of the course description for *Prediction: Machine Learning and Statistics*:

> "Prediction is at the heart of almost every scientific discipline, and the study of generalization (that is, prediction) from data is the central topic of machine learning and statistics, and more generally, data mining. Machine learning and statistical methods are used throughout the scientific world for their use in handling the information overload that characterizes our current digital age. Machine learning developed from the artificial intelligence community, mainly within the last 30 years, at the same time that statistics has made major advances due to the availability of modern computing. However, parts of these two fields aim at the same goal, that is, of prediction from data. This course provides a selection of the most important topics from both of these subjects."

The course *Prediction: Machine Learning and Statistics* is an introductory course that serves primarily PhD and MS students from the MIT Operations Research Center, though its audience includes other PhD students (e.g., from economics) and several undergraduates. The students have a broad mathematical background, and know probability and sometimes optimization, but very few of them previously know machine learning or statistics. So far, none of the students have come into the course intending to work in machine learning or statistics after they finish the course, and most do not expect to take another class on the topic afterwards. There are numerous challenges in teaching for this audience: first, how to provide students with a truly practical knowledge of learning algorithms and why or when they work well, how to give them a broader perspective of predictive modeling in terms of its role in knowledge discovery in data mining, to give them an understanding of the relationships between the history and type of work done in machine learning and statistics in order to understand where these fields are moving, and to help them appreciate the beauty of the mathematical foundations underlying statistical learning.

The course's title was inspired by a blog entry of Brendan O'Connor called "Statistics vs. Machine Learning, Fight!" which aims to put into perspective whether there truly is a difference between machine learning and statistics, quoting several prominent researchers who lie at the intersection. The answer he finds is that clearly these two subjects are not very different, but the most interesting differences are "institutional," including things like teaching style (using Ng's lecture notes as an example of good teaching style, Ng, 2009), and the marketing of ideas and vibrancy (machine learning seems somehow much more vibrant). Much of this has to do with where the fields started, and how

Appearing in *Proceedings of the 29th International Conference on Machine Learning*, Edinburgh, Scotland, UK, 2012.

they evolved over the last 30 years. Statistics started with things of interest to the state – like money, land, and population – modern statistics beginning perhaps with John Graunt studying the plague in England. Machine learning emerged instead from within artificial intelligence. The bottom line is that it is worthwhile to be able to combine the best of both worlds – in other words, to combine tried-and-true methodologies and understanding of statistics with the goals and excitement of machine learning – and that is what the course is about.

In what follows, we will first overview the organizational structure of the course. In the process, we discuss specific choices that make the course unusual amongst machine learning courses, and other choices that help adapt the course to today's students.

2. Overview

There are three components to the course: the "top 10" algorithms in data mining, a theoretical component, and Bayesian analysis.

It is important that a major algorithmic part of the course comes *first* in that students gain a practical understanding and appreciation of what it means to predict from data. This allows the students the possibility of approaching topics like Reproducing Kernel Hilbert Spaces, VC dimension, and covering numbers, even though many of them have not previously had background in functional analysis. By studying the "top 10" algorithms, students feel like they understand why the material is important, and gain the excitement of having an overview worthy of approval by the world's current data mining experts. The students come away with a toolbox of these algorithms that they can use on their own datasets after they leave the course, through functionality built into R (or Matlab).

In between some of the top 10 algorithms, we cover *processes of knowledge discovery*, which seems to be a fairly unusual topic to cover in a machine learning course, and is not, for instance covered in many courses (e.g., that of Ng, 2009), nor is it covered in most major machine learning textbooks (e.g., it is not covered in Hastie et al., 2009; Russell & Norvig, 2009; Mitchell, 1997; Bishop, 2006; Barber, 2012). This topic includes CRISP-DM (Chapman et al., 2000) and the KDD process (Fayyad et al., 1996), giving a history of CRISP-DM, having been developed by several companies that aimed to make the process of discovering knowledge more formal and scientific. Students enjoy recognizing that predictive modeling is only *one* step in the formal process of discovering knowledge from data.

There are several themes that are woven throughout the course. In particular, the theme of the regularized risk:

$$\sum_i \ell(f(x_i), y_i) + CR^{\text{reg}}(f). \tag{1}$$

The functional (1) appears in many different contexts, starting from the slack variables of SVM, to the maximum likelihood derivation of logistic regression and least squares regression, to the maximum a posteriori derivation of ridge regression, to the fortuitous equivalence of AdaBoost to coordinate descent on the exponential loss, to the cost complexity pruning within CART. Each of these algorithms is provided with their usual derivation, and then the connection is made to (1). It is a surprise whenever a special case of (1) appears, having been derived in a new way each time.

The regularized risk is explained at the beginning of the second lecture (after the Apriori algorithm in the first lecture). It is tied to the concepts of overfitting/underfitting the bias/variance tradeoff, and the idea that generalization is "data plus knowledge." The algorithms are introduced in a specific order, where the first four are the simplest to understand. The last several algorithms are ordered so that a concept from each algorithm carries over into the next algorithm.

- Apriori: this is often described as a data miners' first tool, but is often omitted from machine learning curricula (including again major textbooks Bishop, 2006; Barber, 2012).

- K-Means, K-NN, and Naive Bayes: these are the simplest to understand, which is why they are discussed early.

- Decision Trees, with separate derivations for C4.5 and CART. It is useful to study two decision tree algorithms to appreciate that there are a variety of ways to construct a decision tree.

- Logistic Regression, which coincidentally is not one of the "top 10" algorithms in data mining. We give a derivation and history of the concepts of logistic regression, starting with Verhulst's study of the growth of populations (1804-1849), through to the controversy of whether the logistic function could replace the cdf of the normal distribution in the mid-20th Century. Logistic regression, along with the next two algorithms, creates a linear model, and has an interpretation in terms of (1).

- AdaBoost, which in many ways has a very strong relationship to logistic regression. In particular,

they both yield estimates of $P(y = 1|x)$ from extremely similar formulas. Though we provide the intuition behind the original ideas of a weak learning algorithm being made into a strong learning algorithm, we derive AdaBoost using the statistical approach to boosting (Friedman et al., 2000). However, because of this choice, we could not use the original convergence rate calculations for AdaBoost, so we adapted them to the "statistical approach" perspective. We end the topic by discussing the concept of the margin, and that AdaBoost approximately maximizes the ℓ_1 margin, but does not always do so. This leads right into the motivation for support vector machines, which optimizes the ℓ_2 margin.

- Support Vector Machines (SVM) - We provide a tutorial lecture for students on convex optimization beforehand in case they need it.

Each of these algorithms has an implementation in R that students use for their homework assignments, using a folder of processed datasets from the UCI repository that we provide for them (Frank & Asuncion, 2010). Occasionally we ask them not to use the pre-implemented versions of the algorithms, in order to deconstruct an algorithm to understand particular concepts. For instance, one of our homework problems involved implementing K-Means to determine whether the cluster assignment step or the centroid move step generally reduces the cost more than the other one.

Another of our homework assignments asked students to derive a new algorithm, by considering the usual logistic regression model

$$\begin{aligned} \ln\left(\frac{P(Y=1|\mathbf{x},\boldsymbol{\lambda})}{P(Y=0|\mathbf{x},\boldsymbol{\lambda})}\right) &= \boldsymbol{\lambda}^T\mathbf{x} \text{ that is,} \\ \frac{P(Y=1|\mathbf{x},\boldsymbol{\lambda})}{P(Y=0|\mathbf{x},\boldsymbol{\lambda})} &= e^{\boldsymbol{\lambda}^T\mathbf{x}}. \end{aligned}$$

Why does the right side need to be $e^{\boldsymbol{\lambda}^T\mathbf{x}}$? Perhaps it could be replaced by another nonnegative function, like $(\boldsymbol{\lambda}^T\mathbf{x})^2$. Then the students need to calculate the -log likelihood and derive a boosting-style coordinate descent algorithm to optimize it. Then they can figure out whether this is even a good idea.

The second part of the course focuses on kernels and statistical learning theory, and we discuss kernels first. Kernels are motivated using support vector machines, and the fact that some of the most powerful predictive algorithms gain their power through the choice of kernel. However, in order to use a kernel, we need to ensure that the kernel mapping and the higher dimensional feature space really do exist. The notes start from standard linear kernels, and gradually build up to polynomial kernels of arbitrary degree d through a series of examples (in some sense following examples given by Shawe-Taylor & Cristianini, 2000; Schölkopf & Smola, 2001). With the intuition from these examples, we do a calculation in a finite dimensional state space, to show why that if a function k has any chance of being an inner product in a feature space, then every Gram matrix needs to be positive semi-definite. After this, we officially define a kernel, and the first representation of its corresponding Reproducing Kernel Hilbert Space (RKHS), which is the one from the Moore-Aronszajn Theorem. This representation is useful for motivation because one can draw 1-dimensional and 2-dimensional illustrations of the elements of the feature space, simply as real-valued functions (like gaussians). Then we go to the other representation of the RKHS, which is a generalized version of the finite state example we did. This comes from Mercer's Theorem. We then use that representation of the RKHS to prove the Representer Theorem of Kimeldorf and Wahba. At that point it ties back to the general expression (1), because the theorem gives the form of the solution for *any* loss function, with the RKHS regularizer. The rest of the lectures on kernels discuss how to construct kernels that are well-known in practice (e.g, the gaussian).

We motivate statistical learning theory as a way of formalizing that generalization is "data plus knowledge." By this point, the class understands that simpler models that describe the data well are probably the ones that work well for prediction. How to formulate this abstract notion, in a mathematically elegant and precise way, is the goal of statistical learning theory. Before giving any notation, we give intuition through the train/test error vs. complexity tradeoff curves. Those curves illustrate what we are trying to prove, namely the closeness of the training error curve and the test error curve, which depends on the simplicity of the function class. The simplicity of a function class can be measured in many different ways, and we list some of them with a small description of each (VC dimension, covering number, Rademacher averages). Only after this intuition do we introduce notation and start to formalize the concepts. The learning theory notes follow mainly the outline of Bousquet et al. (2003). Illustrations are provided for as many of the concepts as possible, for instance for the regression function and Bayes Classifier, as well as for the idea of a uniform bound.

The course finishes with Bayesian analysis, following Gelman et al. (2003). We start by explaining that most of the other machine learning tools that we discussed

(SVM, boosting, decision trees, etc.) do not make any underlying assumption about how the data were generated; whereas for the remainder of the course, we will assume that the underlying distribution is one of a set. Given our data, our goal is then to determine *which* of these probability distributions generated the data. The lecture follows the "coin flip" example through several concepts: maximum likelihood, MAP, conjugate priors and exponential families. While we are studying MAP, we give the example of linear regression, and show that a multivariate normal prior leads all the way back to the beginning of the course, back to (1) with a least square loss and ℓ_2 regularization.

3. An Updated Teaching Style for Today's Students

Media: It is no longer the case that most graduate courses at MIT are taught on the blackboard. The blackboard, in our experience, is useful in that it allows the lecturer to regulate the speed of the lecture, and allows students to concentrate on the meaning of each symbol as it is written. The lectures of Ng (2009) and lectures at many other universities are still taught on the blackboard; however, we find that generally, MIT students no longer prefer this style. This is a dramatic change, for instance, from 5-8 years ago, and this is not yet the case at many other universities. Some of our students never developed the skill needed to copy and follow the lecture at the same time.

The issue of copying can be somewhat assisted by scribe notes, where one student from the class is assigned to type up the lectures. Scribe notes, in our experience, are very useful, but often have errors, and do not convey intuition of the same quality as is provided by the lecturer. Timing of scribe notes is also an issue: if the lecturer has never taught the topic before, the scribe notes might not be available until days after the lecture is over. Further, if the lecturer changes the lecture from year to year, the scribe notes from the previous year do not completely reflect the lecture from the current year, and cannot be used to follow the current lecture. We do not use scribe notes in our course.

To address this new generation of students, we use pre-prepared LaTeX lecture notes *in large font*, that are designed to be self-contained, but with *specific denoted gaps missing* for blackboard examples and questions for the class. Each blackboard example is designed to show a specific point (for instance, how to get from one stage of Apriori to the next). The questions for the class are placed in a box, so students can see a question coming, and think about how to answer before we get to it. Using a large font format is important, as the notes appear directly on the screen, and it forces us to use short intuitive explanations of each topic, and lots of pictures, in order to avoid large paragraphs on the screen. Having these notes pre-prepared eliminates problems with copying off the blackboard, and focuses the students on the material directly. Further it eliminates the need for scribe notes and the errors and other issues associated with them. We find that as long as the speed of the lecture is regulated very carefully, the students seem to really enjoy this format.

The lecture notes are written in an unusual style, where the English is very *informal*, but with precise mathematical derivations. This allows us in the descriptions to focus on providing useful intuition, while at the same time, being clear technically. The English explanations are written the same as we would provide intuition verbally.

Powerpoint and internet resources are also used for showing practical examples from recent research, and to illustrate the convergence of different algorithms. For instance, we use videos to demonstrate the convergence of K-Means, we use Yoav Freund's applet demonstrating AdaBoost[1], and Yann LeCun's LeNet[2] webpage and an example of our own work (on smart grid maintenance, Rudin et al., 2010) to show machine learning in practice.

Schedule: We also usually take a break an hour into the lecture (which is 1 hour and 20 minutes), and encourage students to come up and ask additional questions. This allows us to gauge the understanding of the students in the earlier part of the lecture, without having to embarrass anyone by cold-calling in front of a crowd to see whether they understood something only a few minutes after they had heard it for the first time. Very often we get excellent questions during the break, and formally clarify the answers to the class before moving on.

Diversity: The audience for this course has a diverse mathematical background, as is true for courses taught in an interdisciplinary department. This course is taught at a management school, which is extremely interdisciplinary. Since it is a mix of students from almost all academic levels at the university (advanced PhD, early PhD, masters, and advanced undergraduates), it needed to be very self-contained, include introductory material, and provide a lot of intuition. At the same time, the lectures needed to cover a range of material that would be interesting to advanced stu-

[1] http://cseweb.ucsd.edu/~yfreund/adaboost/index.html
[2] `http://yann.lecun.com/exdb/lenet/`

dents in other fields.

4. Summary

We presented the course "Prediction: Machine Learning and Statistics" that aims to bridge the best of both ML and statistics, and to unify ideas from both fields. The course is designed for mathematically-oriented non-experts, with a very wide range of expertise and personal interest in the topic. We presented ideas about i) merging themes from ML and statistics, ii) deriving most of the top 10 algorithms in the beginning of the course, iii) relating as many algorithms as possible to the risk functional (1), iv) considering topics from the broader data mining perspective (Apriori, processes for knowledge discovery) and v) using the format of large font lecture notes and blackboard examples, with an informal style.

References

Barber, D. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012.

Bishop, Christopher M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 2006.

Bousquet, Olivier, Boucheron, Stéphane, and Lugosi, Gábor. Introduction to statistical learning theory. In *Advanced Lectures on Machine Learning*, volume 3176 of *Lecture Notes in Computer Science*, pp. 169–207. Springer, 2003.

Chapman, Pete, Clinton, Julian, Kerber, Randy, Khabaza, Thomas, Reinartz, Thomas, Shearer, Colin, and Wirth, Rüdiger. CRISP-DM 1.0. Technical report, SPSS, 2000.

Fayyad, Usama, Piatetsky-Shapiro, Gregory, and Smyth, Padhraic. From data mining to knowledge discovery in databases. *AI Magazine*, 17:37–54, 1996.

Frank, A. and Asuncion, A. UCI machine learning repository, 2010. URL http://archive.ics.uci.edu/ml.

Friedman, Jerome, Hastie, Trevor, and Tibshirani, Robert. Additive logistic regression: A statistical view of boosting. *Annals of Statistics*, 38(2):337–374, April 2000.

Gelman, Andrew, Carlin, John B., Stern, Hal S., and Rubin, Donald B. *Bayesian Data Analysis, Second Edition*. CRC Press, 2003.

Hastie, Trevor, Tibshirani, Robert, and Friedman, Jerome. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Springer, 2009.

Mitchell, Tom. *Machine Learning*. McGraw Hill, 1997.

Ng, Andrew. Course notes for cs229, 2009.

Rudin, Cynthia, Passonneau, Rebecca, Radeva, Axinia, Dutta, Haimonti, Ierome, Steve, and Isaac, Delfina. A process for predicting manhole events in Manhattan. *Machine Learning*, 80:1–31, 2010.

Russell, Stuart and Norvig, Peter. *Artificial Intelligence: A Modern Approach, 3rd Edition*. Prentice Hall, 2009.

Schölkopf, Bernhard and Smola, Alexander J. *Learning With Kernels*. MIT Press, 2001.

Shawe-Taylor, John and Cristianini, Nello. *Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000.

Wu, Xindong, Kumar, Vipin, Quinlan, J. Ross, Ghosh, Joydeep, Yang, Qiang, Motoda, Hiroshi, McLachlan, Geoffrey J., Ng, Angus, Liu, Bing, Yu, Philip S., Zhou, Zhi-Hua, Steinbach, Michael, Hand, David J., and Steinberg, Dan. Top 10 algorithms in data mining. *Knowl Inf Syst*, 14:1–37, 2008.