Introduction to Foundations of Graphical Models

David M. Blei
Columbia University

September 2, 2015

Probabilistic modeling is a mainstay of modern machine learning and statistics research, providing essential tools for analyzing the vast amount of data that have become available in science, government, industry, and everyday life. This course will cover the mathematical and algorithmic foundations of this field, as well as methods underlying the current state of the art.

[What kinds of problems with data do you care about?]

Over the last century, many problems that have been solved (at least partially) with probabilistic models. Here are some examples:

- Group genes into clusters
- Filter email that is likely to be spam
- Transcribe speech from a recorded signal
- Identify recurring patterns in gene sequences
- Uncover hidden topics in collections of texts
- Predict what someone will purchase based on his or her purchase history
- Track an object's position via radar measurements
- Determine the structure of the evolutionary tree of a set of species
- Identify the ancestral populations embedded in the human population.
- Diagnose a disease from its symptoms
- Decode an original message from a noisy transmission
- Understand the phase transitions in a physical system of electrons
- Find the communities embedded in a massive social network
- Locate politicians on the political spectrum based on their voting records

For each of these applications of probabilistic modeling, someone determined a statistical model, fit that model to observed data, and used the fitted model to solve the task at hand. As one might expect from the diversity of applications listed above, each model was developed and studied within a different intellectual community.

Over the past two decades, scholars working in the field of machine learning have sought to unify such data analysis activities. Their focus has been on developing tools for devising, analyzing, and implementing probabilistic models in generality. These efforts have lead to the body of work on *probabilistic graphical models*, a marriage of graph theory and probability theory. Graphical models provide a language for expressing assumptions about data, and a suite of efficient algorithms for reasoning and computing with those assumptions.

As a consequence, graphical models research has forged connections between signal processing, coding theory, computational biology, natural language processing, computer vision, and many other fields. Knowledge of graphical models is essential to academics working in machine learning and statistics, and is of increased importance to those in the other scientific and engineering fields to which these methods have been applied.

Example: Latent Dirichlet allocation

To give you an idea of what applied probabilistic modeling is, I will quickly descrbe latent Dirichlet allocation (LDA) (Blei et al., 2003), which is a kind of probabilistic topic model. (If you have seen me speak, you have probably heard about LDA.)

Basically, LDA is a model of large document collections that can be used to automatically extract the hidden topics that pervade them and how each document expresses those topics. It has become a widely-used method for modeling digital content, and is an example of a successfully deployed probabilistic model. (I developed LDA with Andrew Ng and Michel Jordan in the late nineties. Note it was my final project in a class like this. Andrew Ng was the TA; Michael Jordan was the professor.)

[Slides about LDA and projects from my research group.]

Box's Loop

[This text was taken, largely unchanged, from Blei (2014).]

Our perspective is that building and using probabilistic models is part of an iterative process for solving data analysis problems. First, formulate a simple model based on the kinds of hidden structure that you believe exists in the data. Then, given a data set, use an inference algorithm to approximate the posterior—the conditional distribution of the hidden variables given the data—which points to the particular hidden pattens that your data exhibits. Finally, use the posterior to test the model against the data, identifying the important ways that it succeeds and fails. If satisfied, use the model to solve the problem; if not satisfied, revise the model according to the results of the criticism and repeat the cycle. Figure 1 illustrates this process.

We call this process "Box's loop". It is an adaptation—an attempt at revival, really—of the ideas of George Box and collaborators in their papers from the 1960s and 1970s (Box and Hunter, 1962, 1965; Box and Hill, 1967; Box, 1976, 1980). Box focused on the scientific method, understanding nature by iterative experimental design, data collection, model formulation, and model criticism. But his general approach just as easily applies to other applications of probabilistic modeling. It applies to engineering, where the goal is to use a model to build a system that performs a task, such as information retrieval or item recommendation. And it applies to exploratory data analysis, where the goal is to

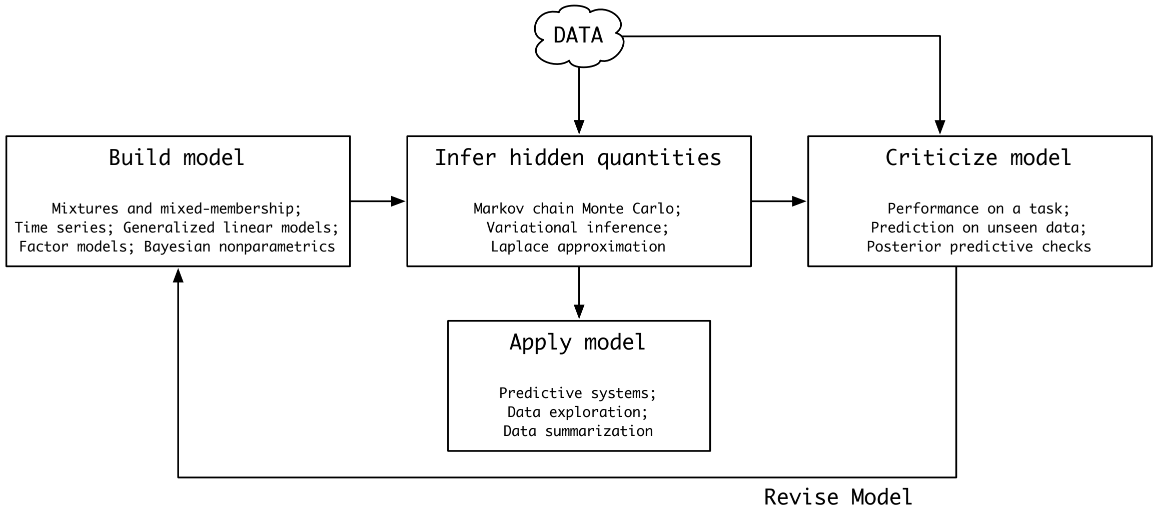
DATA
Build model
Mixtures and mixed-membership; Time series; Generalized linear models; Factor models; Bayesian nonparametrics
Infer hidden quantities
Markov chain Monte Carlo; Variational inference; Laplace approximation
Criticize model
Performance on a task; Prediction on unseen data; Posterior predictive checks
Apply model
Predictive systems; Data exploration; Data summarization
Revise Model

Figure 1: Building and computing with models is part of an iterative process for solving data analysis problems. This is Box's loop, an adaptation of the perspective of Box (1976).

summarize, visualize, and hypothesize about observational data, i.e., data that we observe but that are not part of a designed experiment.

Why revive this perspective now? The future of data analysis lies in close collaborations between domain experts and modelers. Box's loop cleanly separates the tasks of articulating domain assumptions into a probability model, conditioning on data and computing with that model, evaluating it in realistic settings, and using the evaluation to revise it. It is a powerful methodology for guiding collaborative efforts in solving data analysis problems.

As machine learning researchers and statisticians, our research goal is to make Box's loop easy to implement, and modern research has radically changed each component in the half-century since Box's inception. We have developed intuitive grammars for building models, scalable algorithms for computing with a wide variety of models, and general methods for understanding the performance of a model to guide its revision. This course gives a curated view of the state-of-the-art research for implementing Box's loop.

In the first step of the loop, we build (or revise) a probability model. *Probabilistic graphical models* (Pearl, 1988; Dawid and Lauritzen, 1993; Jordan, 2004) is a field of research that connects graph theory to probability theory, and provides an elegant language for building models. With graphical models, we can clearly articulate what kinds of hidden structures are governing the data and construct complex models from simpler components—like clusters, sequences, hierarchies, and others—to tailor our models to the data at hand. This language gives us a palette with which to posit and revise our models.

The observed data enters the picture in the second step of Box's loop. Here we compute the posterior distribution, the conditional distribution of the hidden patterns given the obervations, to understand how the hidden structures we assumed are manifested in the data.[1] Most useful models are difficult to compute with, however, and researchers have developed powerful *approximate posterior inference* algorithms for approximating these conditionals. Techniques like Markov chain Monte Carlo (MCMC) (Metropolis et al., 1953; Hastings, 1970; Geman and Geman, 1984) and variational inference (Jordan et al., 1999; Wainwright and Jordan, 2008) make it possible for us to examine large data sets with sophisticated statistical models. Moreover, these algorithms are modular—recurring components in a graphical model lead to recurring subroutines in their corresponding inference algorithms. This has led to more recent work in efficient generic algorithms, which can be easily applied to a wide class of models (Gelfand and Smith, 1990; Bishop et al., 2003).

Finally we close the loop, studying how our models succeed and fail to guide the process of revision. Here again is an opportunity for a revival. With new methods for quickly building and computing with sophisticated models, we can make better use of techniques like predictive sample reuse (Geisser, 1975) and posterior predictive checks (Box, 1980; Rubin, 1984; Gelman et al., 1996). These are general techniques for assessing model fitness, contrasting the predictions that a model makes against the observed data. Understanding a model's performance in the ways the matter to the task at hand—an activity called *model criticism*—is essential to solving modern data analysis problems.

Course Topics

See the syllabus for a week-by-week description what we will cover. Note the readings will often go beyond what we can cover in lecture.

Additional Discussion

Programming languages. There will be a handful of programming assignments. However, we expect that you already know a good programming language (or two).

For prototyping and developing algorithms, I like the programming language R and embellishments like RStudio. This is not the only choice—I know that many like to use Python, Julia, and probably others I do not know about. (Matlab seems to have fallen out of favor.)

On the backend, to make things fast, I use C. But it seems that my collaborators mostly use C++.

[1]In a way, we take a Bayesian perspective because we treat all hidden quantities as random variables and investigate them through their conditional distribution given observations. However, we prefer the more general langauge of latent variables, which can be either parameters to the whole data set or local hidden structure to individual data points (or something in between). Further, in performing model criticism we will step out of the Bayesian framework to ask whether the model we assumed has good properties in the sampling sense.

Stan is a probabilistic programming language that is actively developed here at Columbia by Andrew Gelman, Bob Carpenter, and colleagues. It lets you specify a probabilistic model programmatically and then "compile" it down to an inference algorithm, an executable that takes data as input and returns estimates of the posterior distribution. I encourage you to try it out at some point during the semester.

Solving real problems involves many hours of "data wrangling", working with online APIs and otherwise cleaning and manipulating data so that it is easy to analyze. For this important activity, you will need to be fluent in a scripting language. I recommend Python.

Applications. In lecture we focus on methods. We will mention applications, especially as motivating concrete examples, but there will not be readings about specific applications.

That said, most of you will be doing a project connected to an application, and we expect you to come up to speed on the state of the art of that application. A student doing a project about recommendation systems should read about probabilistic recommendation systems; a student doing a project about population genetics should read about the probabilistic perspective on their field.

Building and using models. This course is about how to build and compute with probabilistic models that are tailored to the problem at hand. (Note that it is not a course that gives a "cookbook" of methods and when to use them.) Returning to the figure about Box's loop, we are going to focus on the model building piece and the inference piece. What components are in my toolbox with which to build models? How do I compose them together? What algorithms are available to compute with the resulting model and what are their properties? How do I derive an algorithm for the model I want to work with?

Two of the other pieces of the picture—getting the right data and using the results of inference—are equally important, but are specific to the problems that you will be individually working on. The final piece—revising models (and building them in the first place)—is a fuzzy and difficult problem. We will discuss it toward the end of the semester. Building and diagnosing models is more of a craft at this point, one learned through experience.

References

Bishop, C., Spiegelhalter, D., and Winn, J. (2003). VIBES: A variational inference engine for Bayesian networks. In *Neural Information Processing Systems*. Cambridge, MA.

Blei, D. (2014). Build, compute, critique, repeat: Data analysis with latent variable models. *Annual Review of Statistics and Its Application*, 1:203–232.

Blei, D., Ng, A., and Jordan, M. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Box, G. (1976). Science and statistics. *Journal of the American Statistical Association*, 71(356):791–799.

Box, G. (1980). Sampling and Bayes' inference in scientific modeling and robustness. *Journal of the Royal Statistical Society, Series A*, 143(4):383–430.

Box, G. and Hill, W. (1967). Discrimination among mechanistic models. *Technometrics*, 9(1):pp. 57–71.

Box, G. and Hunter, W. (1962). A useful method for model-building. *Technometrics*, 4(3):pp. 301–318.

Box, G. and Hunter, W. (1965). The experimental study of physical mechanisms. *Technometrics*, 7(1):pp. 23–42.

Dawid, A. and Lauritzen, S. (1993). Hyper Markov laws in the statistical analysis of decomposable graphical models. *The Annals of Statistics*, 21(3):1272–1317.

Geisser, S. (1975). The predictive sample reuse method with applications. *Journal of the American Statistical Association*, 70:320–328.

Gelfand, A. and Smith, A. (1990). Sampling based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85:398–409.

Gelman, A., Meng, X., and Stern, H. (1996). Posterior predictive assessment of model fitness via realized discrepancies. *Statistica Sinica*, 6:733–807.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741.

Hastings, W. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109.

Jordan, M. (2004). Graphical models. *Statistical Science*, 19(1):140–155.

Jordan, M., Ghahramani, Z., Jaakkola, T., and Saul, L. (1999). Introduction to variational methods for graphical models. *Machine Learning*, 37:183–233.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, M., and Teller, E. (1953). Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann.

Rubin, D. (1984). Bayesianly justifiable and relevant frequency calculations for the applied statistician. *The Annals of Statistics*, 12(4):1151–1172.

Wainwright, M. and Jordan, M. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305.