

CS491/591-002: Introduction to Machine Learning

Instructor: Terran Lane
Spring 2006
Tue/Thu 4:00-5:15
ORTG 121

1 Goals and Subjects (What We're Doing Here)

This class is about machine learning. I'll say more about what that means in the first lecture, but the short short version is: systems and algorithms that get smarter over time.

This class is mostly about the basics — formulating the problem, mathematical foundations, definitions, results, basic algorithms, etc. Unfortunately, the field is just *way, way* too big to do it all in one class. My goal is to hit the important foundations so that, by the end of the class, you will all know what is out there, know how to apply it, and be able to get started in ML research if you want to.

This class is mostly focused on the statistical approach to ML. It's the predominant current approach, and has proven to be pretty effective. There are some other formulations, though — we may talk about some of them as we have time/interest.

Speaking of interest. . . This is your class as much as mine. I have too many topics that I would like to cover, and not enough time for all of them. And there's stuff that some of you are interested in that I may not even have thought about. I'm happy to take suggestions on what to cover. If you're deeply passionate about something, let me know and I'll see if we can go there.

2 Textbook (What to Read When the Lecture Makes No Sense)

The “official” (primary recommended) text for the class is:

Duda, R. O., Hart, P. E., and Stork, D. G., *Pattern Classification*, 2nd. John Wiley & Sons, 2001.

It should be available from the UNM bookstore or your favorite online purveyor of fine technical textbooks.

Some other very useful texts include:

Mitchell, T. M., *Machine Learning*, McGraw-Hill, 1997. (Another canonical text. Very accessible, but not as unified as the DHS text.)

Witten, I. H and Frank, E., *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann, 2000. (Much less technical, but gives some good, practical, real-world examples. And code.)

Hastie, T., Tibshirani, R., and Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, 2001. (Very, very technical. Excellent and thorough, but can be difficult to follow.)

Sutton, R. and Barto, A., *Reinforcement Learning*, MIT Press, 1998. (Available in full text online, if you look around a bit.)

Hand, D., Mannila, H., and Smyth, P., *Principles of Data Mining*, MIT Press, 2001. (A good, relatively in-depth text, focusing on the theory and practice of data mining specifically.)

3 Class Resources (Me, the Web, etc.)

I can be reached most easily by email to terran@cs.unm.edu

I will (usually) hold office hours 9:00-11:00 AM on Wednesdays. I will also be available for up to 30 minutes after class for questions, brainstorming, philosophy, etc. If none of those times work for you, I'm also happy to make an appointment to meet with you — send me email.

Further information on the class (including the content of readings, specifics on homework assignments, etc.) will be made available via the web:

<http://www.cs.unm.edu/~terran/classes/cs591-s06/>

There is a mailing list for this class: ml-class@cs.unm.edu. I *strongly* suggest that you subscribe to this list, as it will be used for class administrative updates as well as for some discussion of class topics. (Go to <http://www.cs.unm.edu/cgi-bin/mailman/listinfo/ml-class> to subscribe.)

Hint: *Please, please, please*, if you're having difficulties in the class, for whatever reason, contact me *early*. I really want all of you to enjoy and succeed in this class, but I can't help you do so if you don't talk to me. Preferably early enough to do something about it — two days before the final is probably too late make any difference! I'm happy to talk to you in office hours, or, if you're more comfortable with written communication, by email.

4 Assignments, Grading, and Handins (What You'll Do)

There are four parts to the final grade:

(15%) Participation in discussions — especially on the assigned readings.

(20%) Homework assignments.

(30%) Exams (2).

(35%) A final project.

4.1 Readings

The goal of doing research is to be at the cutting edge. The problem is that the cutting edge keeps moving.¹ So we have to read the peer reviewed literature in the field. A lot. That keeps us up on what's going on and what the big trends are. And, even more importantly, it keeps us from reinventing the wheel.

As it turns out, learning to read research literature is itself a skill (and even a slightly different one for each discipline!). While much of this class will be based on the text, we'll also look at some papers from the current/recent literature.

I will announce the readings in class and on the web at least a week in advance. Near the start of the term, we will divide into reading groups, each of which will be responsible for reading and discussing the designated papers *in advance* and producing a short written critique of the readings (due at the *beginning* of the corresponding class). Each group should also produce a list of questions and/or observations on the readings for the class discussion. The written critiques and questions go toward the "participation" part of your grade.

Last note on readings: I have some target topics and papers in mind, but I'll welcome suggestions. If there's a topic that you're curious about (and especially if you have a suggested paper or two), I'll be happy to consider it.

4.2 Homework Assignments

There will be a small number of homework assignments, which will include a mixture of mathematical exercises, informal analysis, and short programming assignments. Homework is due at the *beginning* of the indicated class session. I encourage you to *discuss* homework problems with your classmates, but each individual is responsible for *doing* and *writing up* her or his own solution to the homework.

4.3 Exams

There will be a midterm and a final exam. These will be in-class, individual tests.

4.4 The Final Project

The biggest single part of your grade will be based on a final project. These are to be individual projects, though in exceptional circumstances I may consider a request to do a group project. The goal is for you to explore some topic(s) in greater depth than we have time for in the lecture/readings. More on this later. For now, keep an eye out for a topic of interest.

Hint: The final project is the chance to "play to your strengths". You have a chance to choose something you like and that you're good at. If parts of the rest of the class have left you cold (or been overwhelming), the final project is a great chance to show off your coolness.

¹Blast those darned productive researchers!

5 Schedule (When Things Happen)

This is a rough and optimistic schedule. And it's subject to revision, given your interests. Give me feedback!

Weeks 1–2 Introduction; basic concepts and definitions; examples; empirical methodology.

Weeks 3–5 Introductory classification and regression; decision trees; classification by linear machines; support vector machines; linear regression.

Weeks 6–8 Introduction to generative classification; Bayes' rule; principles of Maximum Likelihood and MAP; graphical models.

Decision point At this point, we have a couple of topics that we can do for the rest of the semester. I'll take votes from the class for one of the following:

Reinforcement learning Time series processes; Markov chains; decision theory; the Markov decision process formalism; the reinforcement learning problem; model-based and model-free RL; POMDPs (if time permits).

Unsupervised learning Clustering (agglomerative and distance based); mixture models; expectation-maximization; Bayesian networks; dimensionality reduction methods (if time permits).

Week 15 Final project presentations.

Final Exam Week The final exam will be Tues, May 9 2006, 5:30-7:30 PM, in Ortega 121 (the regular lecture room).

6 CS491/591 FAQ

6.1 What do I have to know to take this class?

We'll cover this in the first lecture. But for the record, here are some things that would be very useful to know when you walk in:

- Be able to program in at least one language
- Linear algebra
- Probability and statistics
- Data structures and algorithms

If you have had all of these at the undergraduate level, you should be ok in this course. (Though you might have to do some catching up.)

6.2 When is stuff due?

All assignments are due at the *beginning* of class on the due date. Not the middle or the end. (I.e., you have incentive to actually be in class on time. ;-)

6.3 Can I hand stuff in on paper?

Sure. I welcome that. Do please *either* write clearly and legibly, or type (word process) it. If I can't read it, I can't grade it.

6.4 Can I turn stuff in electronically?

Absolutely. Save trees! Of course, it may be harder for me to mark up your answers in informative ways, but I'll give it my best.

6.5 Shiny! I'll ship you my AmigaEdit 1.3 file right away!

NOOOOOOOO!!!!

Please format your submission reasonably nicely and send it in either PDF, PostScript, HTML, or plain ASCII. Please do *not* send non-portable document formats.

I DO NOT ACCEPT MS-WORD DOCUMENTS.

You may send me MIME attachments, Zip-compressed archives, or tar/zip archives if you wish. If you send me an archive (.zip or .tar.gz), *please* set it up to create a subdirectory for its files, named with your last name and the assignment (e.g., "lane_hw3"). This prevents your files from clobbering someone else's files.

6.6 Ok, so when is *electronic* stuff due?

Your submission must arrive in my mailbox (as measured by *my* computer's clock) by the beginning of class on the specified day.

6.7 What programming languages do I need to work in?

I don't care. No, really. I want to see that you can turn high level math and algorithms into functioning code in *some* language. I don't really care which one.² My motto is: "Use the right tool for the job."

6.8 So you can, like, read every programming language in existence?

Absolutely! How else do you think you get to be a CS professor?

Well, ok, not so much.

The truth is — I'm not going to read your code in depth at all. This is a graduate (or nearly so) advanced pre-research course. It's not a software engineering course. If you want me to grade your code, take CS351. I only ask you to turn in your code so that (a) I can tell that you wrote it and (b) I can tell that nobody else wrote it for you. If your results look alien, I might look in more depth to see if I can figure out what went wrong, but really, I mostly care about the results themselves.

6.9 So what do I turn in for programming projects?

When an assignment (including the final project) includes programming, you should turn in a full copy of your code with the rest of the assignment. You can email code separately from a paper hand-in of results if you like.

6.10 What if I *really, really* have to turn something in late?

Ok, life happens. We can't all be on time all of the time. So there's slack built into the system.³

Here's the deal: Say an assignment is due at the beginning of class (4:00 PM) on Thurs, Mar 9, 2006.⁴ By 4:05 or so, it's late. From then 'til 4:00 PM on the 10th, it's 1 day late. From 4:00 PM on the 10th 'til 4:00 PM on the 11th, it's 2 days late. And so on.

Every day late knocks off 33% of the grade value. So if, in that example, you turned your homework in at 11:53 AM on Fri, Mar 10, it would be 1 day late. Suppose the "face grade" of the assignment is 83/100. Then the "final grade" for the assignment would be $0.67 * 83 = 55.61/100$.

6.11 33% per day? You call that slack?

No, here's the slack bit. Everybody gets 3 free "slack days" for the whole semester. One slack day wipes out one late day. So, basically, you get three free late days over the course of the semester.

²But don't abuse this. If you find some language where "build a support vector classifier" is a primitive, atomic operation, that kind-of defeats the point. Be reasonable. If you have a question, ask.

³Thanks to Dave Ackley for the slack mechanic.

⁴All times in this example are Mountain time zone.

6.12 When do I tell you where I want to assign the slack days?

You don't. At the end of the semester, I'll go back and put in slack days in such a way as to maximize your grade.

6.13 Are leftover slack days worth anything?

Yes, leftover slack days can be redeemed at the end of the semester for an unspecified amount of beneficence points.

6.14 What if my iguana dies or my car is struck by lightning in Montana over spring break or I'm kidnapped by aliens or something else that takes more than three days?

Well, we'll all hope for no emergencies that take more than 3 days. Slack days are *intended* to cover things like colds, broken leg while skiing, etc.⁵ But life does happen. If you have a real, serious emergency, please come talk to me. I'll be happy to work with you if you have something big happen in life.

6.15 Can I work with other people in the class on homework and stuff?

Absolutely! You're smarter as a group than you are as individuals. And you're all adults now. I *encourage* you to work with other people in the class.

6.16 Define: "work with"

You are allowed/encouraged to:

- Talk to other people about homework problems.
- Ask math questions.
- Brainstorm about how to solve problems.
- Post questions or suggestions on approaches to the class mail list.
- Help *debug* each other's code.
- Help each other read/interpret papers.
- Brainstorm on project ideas.
- Proofread/proofwatch each other's final project write-ups/presentations.

⁵Corollary: don't try to spend slack days early or unwisely. You may need them for the Martian Death Flu in March.

6.17 Wow. That's pretty broad. What's left for me to do?

Ultimately, you have to prove to me that *you* understand what's going on. So:

- *You* must write up your own solutions to all problems yourself, in your own words. This can be based on discussions with other students, but the final solution must be yours.
- All code must be your own.
- You must run all experiments yourself, using your own code.
- The final project must be your own work, your own writeup, and your own presentation.
- Exams are completely individual. Duh.

6.18 It's all so confusing!

Be at peace, grasshopper. If you have any questions about anything — what's allowed, what you need to do for some assignment, whatever — please ask. I'm happy to help fill things in. And I'm *much* happier to clarify policy *beforehand* than to have to take “corrective action” afterward.

So just remember — if you're confused: ask first.

6.19 Is there anything else I should know?

- Staples are your friend.
- The spell checker is also a friend, albeit a frequently deceptive one. Rely on it with caution.
- Gauss was a real person, so things derived from his name should be capitalized. Gaussian distribution, Gauss-Jordan elimination, Gauss integral, etc. Similarly, Markov, Laplace, Dirichlet, etc. should all be properly capitalized.
- Plots should always be fully labeled — both axes, and a title indicating content. If the plot contains more than one data set, please distinguish the different data sets by symbol or line type (e.g., '+' vs 'o' or ':' vs '-') and include a legend that names each. Finally, be sure to distinguish *discrete* from *continuous* data in your plots. E.g., if you have a sample of 10 measured points, do *not* simply draw a curve through them. At the very least, indicate the location of each point with a discrete symbol, such as a cross or square, superimposed on the curve. Better to omit the curve altogether as it implies the presence of unmeasured points. (Unless, of course, you have good reason to draw such a curve, and make it clear that it's interpolated rather than measured.)
- I expect proper grammar, spelling, punctuation, etc., on all assignments. And it wouldn't hurt to brush up on it in email too. ; -) Being a computer hacker does not excuse you from natural language skills!

The Dark Side of the FAQ

6.20 I know of this great web site with *all* the answers posted! And there's some k3w1 warz3 sites with all the code!

Ok, I shouldn't have to say this to people at your level. But this comes up occasionally, so it's worth being very clear up front:

Dishonest behaviors, including but not limited to plagiarism, copying of another student's work (or providing your own to another), group consultation on individual projects or work, copying solutions from the web, etc., will not be tolerated.

My general feeling is that being caught cheating should be more painful than not having done the assignment at all. Therefore, I will generally *at least* assign a negative penalty equal to the full value of the assignment if I discover someone cheating on an assignment. I.e., if an assignment is worth 10% of the final grade, the individual would receive not zero credit for the assignment, but -10%.

Check that out again: *It is better to not do the assignment than to cheat on it.*

6.21 Yeah, but I'm really smart. I can get away with it!

Yeah, whatever. If you're smart enough not to get caught, then you're smart enough to not to have to cheat in the first place.