A Data Mining Course for Computer Science: Primary Sources and Implementations

David R. Musicant
Carleton College
Department of Mathematics and Computer Science
One North College Street
Northfield, MN 55057
dmusican@carleton.edu

ABSTRACT

An undergraduate elective course in data mining provides a strong opportunity for students to learn research skills, practice data structures, and enhance their understanding of algorithms. I have developed a data mining course built around the idea of using research-level papers as the primary reading material for the course, and implementing data mining algorithms for the assignments. Such a course is accessible to students with no prerequisites beyond the traditional data structures course, and allows students to experience both applied and theoretical work in a discipline that straddles multiple areas of computer science. This paper provides detailed descriptions of the readings and assignments that one could use to build a similar course.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning—*concept learning, induction*; I.5.2 [**Pattern Recognition**]: Design Methodology—*classifier design and evaluation*; I.5.3 [**Pattern Recognition**]: Clustering—*algorithms, similarity measures*; K.3.2 [**Computers and Education**]: Computer and Information Science Education—*computer science education*.

General Terms

Algorithms, measurement, design, experimentation.

Keywords

Data mining, machine learning, course design.

1. INTRODUCTION

Data mining is an exciting and relatively new area of computer science that lies at the intersection of artificial intelligence and database systems. Defined as the "non trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data" [11], data mining concerns itself with how to automatically find, simplify, and summarize patterns within large sets of data. Machine learning, said to be "concerned with the question of how to construct computer programs that automatically improve with experience"[16], overlaps heavily with data mining in that many of its algorithms learn from data. A course in machine learning and data mining (hereafter simplified to just "data mining") is a wonderful elective class to offer to undergraduates.

A data mining elective has been offered twice at Carleton College. This course has turned out to be a marvelous opportunity for students to use theoretical computer science ideas to solve practical "real-world" problems. Data mining requires a variety of ideas from data structures and algorithms, which gives students the opportunity to see these concepts in practice. It should therefore be pointed out that this paper actually serves a dual role: readers of this paper might find that some of the concepts or assignments contained herein would be useful examples in an advanced data structures class. There are also significant issues with privacy and ethics in data mining, and this provides an opportunity to link computer science with wider affairs. Because students can choose their own datasets to analyze, they get a personal sense of ownership in the work that they do because they can choose data from some application area that interests them. Data mining is a new field, and so most of the seminal work has been written within the last ten years. This adds to the motivational aspects of the course, since the students are actually learning something new to everyone. Finally, I should admit my biases up front: my research is in data mining, and thus I wished to offer my liberal arts students a chance to see how engaging these ideas are.

Why should the fields of machine learning and data mining be taught together in one course? The areas of machine learning and data mining have a very large intersection, which could perhaps be described very simply as "learning from data." There are areas of machine learning that do not interact much with data mining (such as reinforcement learning), and there are areas of data mining that do not seem to capture the flavor of machine learning (such as how to make data analysis algorithms scale gracefully), but the central idea of learning from data is common to both fields. Material found in machine learning books and in data mining books is quite similar. The first time that I offered my course at Carleton, I actually just called it

SIGCSE'06, March 1–5, 2006, Houston, Texas, USA.

"Data Mining." Students indicated in post-course surveys that the name "Machine Learning" was considerably more attractive to them, and students would be more likely to take the course if both names were in the title.

Another question that might be proposed is "Why offer such a course at all? If some of this material is worthwhile, why not merely split up the material between artificial intelligence and database courses?" Bits of these ideas do end up in some of our other courses. I do cover a healthy dose of machine learning in my AI course, and data mining at least gets half a class of discussion in my database course. But the coherent area of data mining is worthy of study in and of itself, and easily can span a semester. Artificial intelligence courses tend to survey AI, and so the amount of time that one can spend on machine learning is constrained. Database courses need to spend significant amounts of time on the functioning of database systems themselves, and thus algorithms for learning from data are hard to fit in.

The course that I have constructed and taught is designed to appeal to computer science students and to reinforce computer science ideas that they have seen elsewhere. Because we are a small program and our courses do not run all that often, it helps to boost enrollments if prerequisites are minimal. Therefore, the only prerequisite that I require is our data structures course. One of the challenges in teaching data mining to undergraduate computer scientists is its high overlap with statistics, which can require significant background by students. Therefore, the course that I have put together is based on two pivotal elements: reading research papers as primary source material, and implementing data mining algorithms via programming. Textbooks on data mining for a course such as this are quite limited. Most data mining textbooks are either not technical enough or require too much mathematical background. A few books out there do seem to fit the audience [10, 23], and I do use Margaret Dunham's text [10] as a side reference. Nonetheless, the collection of research papers that I have assembled seems to provide a considerably richer experience for the students. I have been careful to make sure that these papers are readable by the students with minimal prerequisites. While I do sometimes need to present some material in class to supplement the papers (some linear algebra, some statistics, etc.), with a bit of support students are capable of learning a substantial amount from them.

2. RELATED WORK

I should make sure to point out the differences between my course and other data mining courses that have been presented to the educational community. Dian Lopez and Luke Ludwig at University of Minnesota, Morris have successfully taught a data mining course [15] based on using the Weka data mining environment and its associated textbook [24]. Their course provided a wonderful opportunity for students to use data mining software and to learn about how the algorithms underneath worked, but had a very different emphasis than the course discussed here.

At ITiCSE '05, Richard Roiger presented an excellent tutorial on data mining [20] that introduced educators to the field. He covered many of the ideas in data mining quite clearly so that those new to the field could think about teaching a course in the area. That tutorial is quite different from this paper. Here, I describe in detail how to actually run a data mining course by using specific research papers that are accessible to undergraduates and by creating assignments that allow the students to implement data mining algorithms themselves. Specifically, my course focuses on the "computer science" aspects of data mining. By encouraging students to read papers and actually implement some of the algorithms, the course allows the students to identify more closely with data mining researchers and challenges the students to understand in detail the intricacies of data mining algorithms.

3. THE ASSIGNMENTS

Data mining is a fairly wide field, and in one of our ten week terms the amount of material that can be covered is limited. Further complicating the matter is that I cover some supervised machine learning in my Artificial Intelligence course, but I don't require AI as a prerequisite for Data Mining. Therefore, the material that I cover on supervised machine learning in Data Mining is designed to be complementary to the content that appears in my AI course.

In order to encourage students to complete readings before class, I require that they electronically post to a forum that I have created for the course on Caucus [1], a system that Carleton uses for managing online discussion groups. (Any course management system, blog, or email list would serve the same purpose.) Specifically, I require that the students post one thing that they did not understand about the paper, or alternatively post a particular point that they found interesting. I also require that they post a potential exam question. This gets them thinking about what the important details are, but also makes my life easier when I have to write exams. After each reading has been assigned, we discuss it in class. I use the students' postings to help determine what portions of the papers we need to discuss more carefully in class.

Following is a list of the various topics I cover in the class, along with the readings that I use and the assignments that I give. The readings and assignments that I describe form the major contributions of this paper, as they form the backbone for my course.

3.1 What is Data Mining?

The paper **"Data Mining and Statistics: What's the Connection?"** [12] is an entertaining and controversial look at the distinctions between the fields of data mining and statistics, and what the two fields do correctly and incorrectly in trying to solve similar problems in their own separate ways. I particularly like this paper because it helps students to understand the different perspectives that computer scientists and statisticians offer (for better or for worse!). This reading significantly helps to inspire a specific subset of students who might otherwise mistakenly be concerned that they had been taken by a "bait and switch scheme" to get them into a statistics course.

For an assignment, I direct students to the well-known "census-income" dataset [14] and ask them to provide some simple summary statistics on the data. For example, I ask them to provide the following information:

- how many records and features there are
- how many features are continuous, and how many are nominal

- for the continuous features, the average, median, minimum, maximum, and standard deviation
- 2-dimensional scatter plots of two features at a time

I then encourage them to be creative and to find any interesting patterns that they can.

3.2 Classification and Regression

Classification and regression are forms of supervised learning, where the computer is given labeled data and asked to learn the relationship between the data and the labels. In class, we discuss first the general concepts of classification and regression, and look at various error metrics that one can use such as accuracy, precision, recall, ROC curves, and others. Most of this material is covered in the Dunham text, and so the students are asked to read the sections on these topics.

Traditional algorithms such as decision trees, neural networks, and support vector machines are covered in my AI course, so they get only peripheral mention here. Of course, one could cover these topics in more detail if one wished. Instead, I focus on the k-Nearest Neighbor algorithm from a data mining perspective: how can it be scaled to run on large datasets? The paper **"Nearest Neighbor Queries"** [21] does a fine job at explaining a particular technique for scaling such queries to large datasets. This paper does require an understanding of R-trees, which I cover in class. Otherwise, though, this paper is a wonderful example of how an algorithm can be scaled through the use of data structures and heuristics.

Students are asked to program a straightforward version of the k-nearest neighbor algorithm, and apply it to the census-income dataset. They are instructed to experiment with a variety of values of k, and to try different distance metrics such as Euclidean, Manhattan, and cosine distances. They then produce plots showing training and test set accuracies for these variations, and interpret them in a report that they submit.

3.3 Clustering

Clustering is the act of finding prototypical examples that concisely summarize an entire dataset. This a crucial topic in data mining that I use in my own research, so it receives a significant amount of time in my course. To get started, the students read material in Dunham's text that addresses the basics of clustering, and we discuss it in class. They then implement a basic clustering algorithm (k-means). Now that they have had some experience on a dataset that I chose (census-income), I direct the students to go to the UCI Machine Learning Repository [22] or the UCI KDD Archive [14] and obtain a dataset that interests them. They use this dataset for this assignment and most of the remaining ones. For this assignment, the students are instructed to try a variety of values for k, as well as two different techniques for initializing the algorithm. They turn in a report summarizing the distinctions that they find, as well as interpretations of the clusters that they find. If the dataset that they pick already has a class label associated with each data point, they are encouraged to discard it to make the exercise more interesting.

The students then read two papers on clustering. **"Scaling Clustering Algorithms to Large Databases"** [7] describes the "Scalable K-Means" algorithm, which can cluster data with only one pass through the entire dataset. This paper is particularly interesting for its "data mining desiderata," which is a list of characteristics that every scalable data mining algorithm should have. The list makes for interesting classroom discussion. The second clustering paper that they read is **"CURE: An Efficient Clustering Algorithm for Large Databases"** [13]. CURE is an agglomerative algorithm, and so this illustrates an entirely different approach to clustering. The students then implement agglomerative clustering in an assignment with a stripped-down version of CURE. CURE requires the use of a heap (at least, it does in the way I pose the assignment), so this is a great opportunity to reinforce that idea.

3.4 Association Rules

Association rules are sometimes used for supermarket basket analysis: "what items do people typically purchase at the same time?" The legendary and somewhat mythical example that beer and diapers are often purchased together makes for lively class discussion. The paper **"Fast Algorithms for Mining Association Rules"** [5] presents the classic Apriori algorithm. This is something of a long paper, and so I focus the students by telling them to concentrate on Apriori and its variants and only to skim the material on AIS and SETM. This section of the course usually takes more time than I think it will. This paper is quite rich, and proving that the algorithm itself and its optimizations work correctly takes some doing. The experience is well worth it, as students are again exposed to detailed algorithms that require the user of data structures. For an assignment, they implement the basic form of the Apriori algorithm and implement it on their own dataset.

3.5 Web Mining

Performing data mining on the Web, particularly regarding mining the link structure of the Web, is the most interesting topic in the course for many of the students. It should be noted that mining graphs (such as the Web) is a particularly important current topic: two of the plenary talks given at KDD 2005 were on related material [18, 6]. I first assign the classic paper **"The PageRank Citation Ranking: Bringing Order to the Web"** [17], which described how the basic version of the Google PageRank feature works. One of the most fascinating aspects of the problem is that determining which web pages are more important than others becomes an eigenvalue problem, and thus one needs to focus on how to solve a problem in a scalable fashion. This exposes the students to a small amount of numerical analysis and linear algebra on a critical applied problem. The students also read **"Mining the Link Structure of the World Wide Web"** [9] which presents the HITS algorithm for determining web page importance. This reading is worthwhile as it lets the students see that PageRank is not the only way to solve this problem, and HITS uses a considerably different set of heuristics than PageRank.

For an assignment, the students actually implement the PageRank algorithm and use it on an archive of our department's website. I have learned the hard way, multiple times, that asking students to write a webcrawler for a class assignment can be a bad idea. Having a class of students crawl campus web servers in parallel is a recipe for disaster: this can ultimately result in a well intentioned denial of service attack. For those who wish to remain on good

terms with their IT support, I do not recommend such an assignment! I therefore create an archive of our department's website for the students to use. Rather than requiring the students to parse HTML, I create the archive using some scripts wrapped around `lynx` [2] so that all webpages are already converted to text with links appearing on the bottom of each page. This makes the scanning aspect of the assignment reasonably straightforward. I point out to the students that they must store the data in some kind of sparse format or they will likely run out of memory, and thus they need to use hashing or some other form of map to make their code efficient. The students submit a report indicating what they find, and how they choose to interpret the heuristics in the PageRank paper (some technical details are not fleshed out in the paper).

3.6 Collaborative Filtering

Collaborative filtering systems, also known as recommender systems, make recommendations to users as to items they may wish to buy or what movies they wish to watch based on past preferences. The paper **"Empirical Analysis of Predictive Algorithms for Collaborative Filtering"** [8] surveys a number of collaborative filtering algorithms, and spends time discussing both the algorithms and techniques for measuring how well they work. Students experience collaborative filtering regularly, particularly if they visit websites such as Amazon.com. They are therefore inspired by this material, especially in the context of finding new books or music that interests them. I show them Yahoo's Launchcast [3] as a particularly compelling example.

At this point in the term, students are already working on their final project (see section 3.8), so I do not give any assignments to go with this topic.

3.7 Ethics

At KDD 2004, a panel discussion titled **"Data Mining: Good, Bad, or Just a Tool?"** [19] did a wonderful job of discussing some of the ethical concerns posed by data mining, particularly with regards to preservation of privacy. This panel was videotaped, as were all talks at KDD 2004, and is available from Old Tacoma Communications [19]. I asked my students to post to Caucus before watching the video and to provide an example of how data mining could be exploited for evil purposes. They were also to describe what could be done to prevent this, or to indicate if nothing could be done. I also asked them to post followup commentary after watching the video. I originally conjured up this video as something to keep my students busy while I was out of town, but they seemed to actually find it quite interesting. The students pointed out that more problems than solutions were raised, and in fact many of the students were frustrated by the fact that good solutions to many of these problems were not clear. Many of them also pointed out that they had never before considered the issue of accountability, which is a major theme of the discussion. If someone's privacy is violated, who is responsible (or as one of the speakers ironically points out, "who is sued?"). There is not a clear cut answer to this question either, and the class discussion that followed helped to raise students' awareness of these issues.

3.8 Final Project

For the final project for this course, I give my students freedom to pursue almost anything within the realm of data mining so long as I approve it. Some students choose to find a paper with an algorithm that interests them, and to implement it. Other students want to do a full-fledged data mining study on data which interests them, and so they use a combination of their own code and Weka to analyze some data and produce a report describing what they find. I have also experimented with encouraging students to attack KDD Cup problems. The most recent KDD Cup problem [4] captured the imagination of many of my students, but the task was a little too big and vague for them. It also required a significant amount of data manipulation which was highly educational, but also very time consuming. I do think that with the right scaffolding, however, problems of this type can work quite well.

4. COMMENTARY AND CONCLUSION

The most memorable part of the course for most of the students seems to be the papers. For many of them, this is the only course where they have the opportunity to learn a significant amount of material from sources other than textbooks or their professors. Some of these papers are tricky to read, and some are written better than others. Students seem to gain a better understanding of how to write a paper from reading these. The most difficult part in using these papers is the time that I need to put in to preparing to use them, particularly regarding filling in background holes that the students do not have. This has the added advantage, though, that in-class discussion of a particular paper does not need to consist of me parroting material that is in it. Instead, I present ideas that that the paper does not cover well or at all, and tie it back to the paper as relevant.

Using discussion forums to encourage students to read works well, at least in a more advanced elective such as this one. Students have said that they found this to be a pain to do, but were thankful for it after the fact because it encouraged them to be prepared for class. I do need to make sure that I set the deadline for posting to be a number of hours before class so that I have time to review their comments beforehand.

The programming assignments also seem to mostly work well. I give the students the option of working on these alone or in pairs, and I allow them to program in the language of their choice so long as I can compile and run the code myself. The most challenging part of these assignments is in grading them. Many of the algorithms that I ask them to implement leave some details unspecified, and therefore students obtain differing results from each other even when the data input is the same. I could completely specify the assignments, but the students seem to get something out of the experience of working through how to handle these judgment calls. They also learn from experience that not all primary sources document their work as well as they should. This aspect of the course, combined with the fact that I allow students to use datasets of their own choice, makes grading somewhat difficult. I find that I end up grading assignments on the quality of their post-analysis and on a general sense of "likelihood of being correct" rather than on a precise scrutiny of code and results.

Finally, I should mention that I also assign two take-home midterm exams throughout the term. These are open book written exams that attempt to determine how well the students understand the ideas that we discuss and implement.

The course that I have described above has been quite successful. Students have indicated in evaluations that they really enjoy the opportunity to combine "real-world" problems with computer science theory and algorithms. It has also been immensely useful in introducing students to computer science research; I have found that this course helps considerably in preparing students to work on research with faculty.

All of my course materials are available on my website at `http://www.mathcs.carleton.edu/faculty/dmusican/cs377s05/`.

5. REFERENCES

[1] Caucus. `http://caucus.com`.

[2] Lynx. `http://lynx.isc.org/`.

[3] Yahoo! Launchcast Radio. `http://launch.yahoo.com`.

[4] KDD-Cup Knowledge Discovery and Data Mining competition, 2005. `http://kdd05.lac.uic.edu/kddcup.html`.

[5] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In J. B. Bocca, M. Jarke, and C. Zaniolo, editors, *VLDB*, pages 487–499. Morgan Kaufmann, 1994.

[6] A.-L. Barabsi. The architecture of complexity: The structure and the dynamics of networks, from the web to the cell. Invited talk at ACM-KDD, August 2005.

[7] P. S. Bradley, U. M. Fayyad, and C. Reina. Scaling clustering algorithms to large databases. In *Proceedings of Knowledge Discovery in Data Conference*, pages 9–15, 1998.

[8] J. S. Breese, D. Heckerman, and C. M. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In G. F. Cooper and S. Moral, editors, *UAI*, pages 43–52. Morgan Kaufmann, 1998.

[9] S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, S. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Mining the link structure of the world wide web. *IEEE Computer*, 32(8):60–67, August 1999.

[10] M. H. Dunham. *Data Mining: Introductory and Advanced Topics*. Prentice Hall, August 2002.

[11] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. *Advances in Knowledge Discovery and Data Mining*, chapter From Data Mining to Knowledge Discovery: An Overview, pages 1–30. AAAI Press, Menlo Park, CA, 1996.

[12] J. Friedman. Data mining and statistics: What's the connection? In *Proceedings of the 29th Symposium on the Interface Between Computer Science and Statistics*, 1997. `www-stat.stanford.edu/~jhf/ftp/dm-stat.ps`.

[13] S. Guha, R. Rastogi, and K. Shim. Cure: An efficient clustering algorithm for large databases. In L. M. Haas and A. Tiwary, editors, *SIGMOD Conference*, pages 73–84. ACM Press, 1998.

[14] S. Hettich and S. Bay. The UCI KDD archive, 1999. `http://kdd.ics.uci.edu`.

[15] D. Lopez and L. Ludwig. Data mining at the undergraduate level. In *Proceedings of the Midwest Instruction and Computing Symposium*, Cedar Falls, IA, April 2001.

[16] T. M. Mitchell. *Machine Learning*. McGraw-Hill, Boston, 1997.

[17] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford University, November 1999. `http://dbpubs.stanford.edu/pub/1999-66`.

[18] P. Raghavan. Incentive networks. Invited talk at ACM-KDD, August 2005.

[19] R. Ramakrishnan, D. Mulligan, D. Jensen, M. J. Pazzani, and R. Agrawal. Data mining: Good, bad, or just a tool? Available from Old Tacoma Communications, `http://www.oldtacoma.com/`, August 2004.

[20] R. J. Roiger. Teaching an introductory course in data mining. In *ITiCSE '05: Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education*, pages 415–415, New York, NY, USA, 2005. ACM Press.

[21] N. Roussopoulos, S. Kelley, and F. Vincent. Nearest neighbor queries. In M. J. Carey and D. A. Schneider, editors, *SIGMOD Conference*, pages 71–79. ACM Press, 1995.

[22] C. B. S. Hettich and C. Merz. UCI repository of machine learning databases, 1998.

[23] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison-Wesley, May 2005.

[24] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2nd edition, 2005.