CS519 — Research Methods in Computer Science — Fall 1996

Personnel

Instructor:

Tom Dietterich, Dearborn 306, 737-5559, tgd@cs.orst.edu
Office Hours: MW 1:00-2:00pm

Meeting Times

Lecture: Tues, Thurs 12:30-13:50 ECE 101

Texts

Copies of papers and notes will be distributed in class.

Grading

Approximate percentages of grade:

Paper critiques (6)	30%	30	
Homework assignments (2)	10%	11	(1)
Reports (2)	15%	15	
Grant proposal critiques (2)	10%	12	(3)
15-minute Conference Talk	10%	7	
Grant Proposal	20%	20	
Grant proposal peer review	5%	6	

Course Schedule (tentative)

		PART 1: Choosing Research Projects
Oct	1	Introduction. Research life cycle. Goals and methods.
	3	How to critique a paper. Exploratory Research.
	8	Development of units. Evaluation and criticism of units in isolation.
	10	Evaluation of units in comparison with others.
	15	Study of space of possible units.
	17	Technology transfer. Brainstorming.
	22	Assessment Reports.
		PART 2: Methods and Supporting Skills
	28	3:30pm Dearborn 401: Choosing evaluation criteria. Statistical analysis of data.
	29	Statistical analysis of data
	31	Experiment design
Nov	5	Techniques from experimental psychology.
		PART 3: Professional Career Issues
	7	What is in a proposal?
	8	1:00pm Dearborn 401: Critiques of proposals
	14	How to give a talk. Example talk and critique.
	19	Ethical issues in proposals and presentations. ** Proposal first draft due **
	21	Financial issues in funding; funding agencies
	26	Mini-conference
	28	NO CLASS (Thanksgiving)
Dec	3	NO CLASS (NIPS conference)
	5	NO CLASS (NIPS conference) ** Final Proposal Due **
	13	** Proposal reviews due 9:30am **

Written Work Due Dates (tentative)

		PART 1: Choosing Research Projects
Oct	8	Exploratory critique
	10	Development critique
	10	Isolated Methods critique
	15	Comparison critique
	17	Space of possible units critique
	24	Area Assessment report
		PART 2: Methods and Supporting Skills
	29	Logical Next Step report
	31	Data analysis assignment due
		PART 3: Professional Career Issues
Nov	14	Proposal critiques (3)
	19	Proposal first draft due
	26	15-minute conference talk
Dec	1	Final proposal due
Dec	13	Proposal reviews due.

Lifecycle Readings

- **Concept Learning**

1. Bruner, J.S., Goodnow, J.J., Austin, G. A. (1957) *A Study of Thinking.* New York: John Wiley & Sons, Inc.
2. Hunt, E. B., Marin, J., Stone, P. J. (1966). Experiments in Induction. New York: Academic Press.
3. Quinlan, J. R. (1983). Learning efficient classification procedures and their application to chess endgames, in Michalski, R. S., Carbonell, J., and Mitchell, T. M., (eds.), *Machine learning: An artificial intelligence approach, Vol. I,* Palo Alto: Tioga Press. 463–482.
4. Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning internal representations by error propagation. In Rumelhart, D. E., and McClelland, J. L., (eds.) *Parallel Distributed Processing*, Vol 1. 318–362.
5. Dietterich, T. G., Hild, H., Bakiri, G. (1990) A comparative study of ID3 and backpropagation for English text-to-speech mapping. *Proceedings of the Seventh International Conference on Machine Learning* (pp. 24–31). Austin, TX: Morgan Kaufmann.
6. Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM, 27,* 1134–1142.
7. Blumer, A., Ehrenfeucht, A., Haussler, D., Warmuth, M. (1986). Classifying learnable geometric concepts with the Vapnik-Chervonenkis dimension. 18th ACM Symposium on the Theory of Computing, Berkeley, CA, 1986. 273–282.
8. Ehrenfeucht, A., Kearns, M., Haussler, D., Valiant, L. (1988). A general lower bound on the number of examples needed for learning. *COLT 88*, 139–154.

- **Reduced Instruction Set Computers**

1. Foster, C. C., Gonter, R. H., and Riseman, E. M. (1971). Measures of op-code utilization. *IEEE Transactions on Computers.* 582–584.
2. Lunde, A. (1977). Empirical evaluation of some features of instruction set processor architectures. *Communications of the ACM, 20* (3) 143–153.

3. Patterson, D. A., Ditzel, D. R. (1980). The case for the reduced instruction set computer. *Computer Architecture News, 8* (6) 25–33.
4. Clark, D. W., Strecker, W. D. (1980). Comments on "The case for the reduced instruction set computer" by Patterson and Ditzel. *Computer Architecture News, 8* (6) 34–38.
5. Radin, G. *IBM Journal of Research and Development, 27*(3) 237–246.
6. Patterson, D. A., Sequin, C. H. (1981). RISC I: A reduced instruction set VLSI computer. *Proceedings of the 8th Annual International Symposium on Computer Architecture.* 443–457.
7. Hennessy, J. L. (1984). VLSI processor architecture. *IEEE Transactions on Computers, C-33* (12) 1221-1246.
8. McFarling, S., Hennessy, J. L. (1986). Reducing the cost of branches.
9. Hennessy, J. L., Jouppi, N. P. (1991). Computer technology and architecture: an evolving interaction. *IEEE Computer*, September. 18–29.
10. Bhandarkar, D., Clark, D. W., (1991). Performance from architecture: Comparing a RISC and a CISC with similar hardware organization.

- **Constraint Logic Programming**

1. Borning, A. (1981). The programming language aspects of THINGLab, a constraint-oriented simulation laboratory. *ACM Transactions on Programming Languages and Systems*, 3(4), 252–387.
2. Steele, G., Sussman, G. J. (1980). CONSTRAINTS—A language for expressing almost hierarchical descriptions. *Artificial Intelligence, 14* (1), 1–39.
3. Colmerauer, A. (1984). Equations and inequations on finite and infinite trees. *Proceedingss of the 2nd Conference on Fifth Generation Computer Systems.* ICOT. 85–99.
4. Jaffar, J., Lassez, J-L. (1987). Constraint logic programming. *Proceedings of the 14th ACM Symposium on Principles of Programming Languages.* Munich. 111–119.
5. Lassez, C., McAloon, K., Yap, R. (1987). Constraint logic programming and options trading. *IEEE Expert, 2* (3) 42–50.
6. Dincbas, M., Simonis, H., Van Hentenryck, P. (1990). Solving large combinatorial problems in logic programming. *Journal of Logic Programming 8*, 75–93.
7. Jaffar, J., Michaylov, S., Yap, R. (1991). A methodology for managing hard constraints in CLP systems. *Proc. ACM-SIGPLAN Conference on Programming Language Design and Implementation*, 306–316.
8. Jaffar, J. Michaylov, S., Stuckey, P., and Yap, R. (1992). The CLP(R) language and system. *ACM transactions on Programming Languages, 14*(3), 339–395.
9. Jaffar, J. Michaylov, S., Stuckey, P., Yap, R.H.C. (1992). An abstract machine for CLP(R). *Proceedings of the ACM-SIGPLAN Conference on Programming Language Design and Implementation.* 128–139.
10. Heintze, N., Michaylov, S., Stuckey, P. J. (1992). CLP(R) and some electrical engineering problems. *Journal of Automated Reasoning, 9* 231–260.
11. Jaffar, J., Maher, M. J. (1994). Constraint logic programming: a survey. *Journal of Logic Programming.* To appear.