Random
CS 700 Stuff !

By An AI Researcher !

AI Experiments

- Often require both experiment and proof.
- Often involve stochastic (semi-random) experiments.
- Often involve comparing multiple techniques over several objectives.
- Often use simulations with large numbers of parameters.

Experimental Issues Common in AI

* Good Experiments (randomness, open source verification, replicability, honesty)
* Reporting: LaTeX, gnuplot, R, good writing
* Making Valid Claims (T-tests/ANOVAs, nonparametric tests, multiobjective)
* Proving Generality (multiple problem domains, train/test methodologies)
* Optimization and Experimental Design

Good Experimental Methodology

Verification and Replicability

* Report more than enough information necessary for others to replicate your experiments or verify your proof.

* What if your system is so large and complex that no one can reasonably be expected to replicate it just to verify your crazy claim?

Open Source And Good Experimental Claims

- Make all your code available so others can verify your claims.
- Reasons not to:
  - You're hiding something
  - Your code is embarassingly bad
  - You want to sell your code

Random Number Generators

* Absolutely critical for good stochastic experimental work

* If you have relied on java.util.Random or C/C++'s rand() for your work, it's time to redo all of your experiments.

Random Number Generators

* Statistically Random
* Long Period
* Fast
* Replicable (pseudorandom)
* $R_1 \ldots R_{n-1}$ cannot predict R_n (crypto)

Random Number Generators

- Horrifying: Linear Congruential
  - $R_n = aR_{n-1} + b \pmod{m}$
  - Used in Java, C, C++

alife.co.uk/nonrandom/

- Better
  - Knuth Subtractive, Linear Feedback Shift Register, Lagged Fibonacci
  - Mersenne Twister

Plagiarism and Faking or Misreporting Results

Plagiarism and Faking or Misreporting Results

* Your career is over.

Ways to End Your Career

- Your technique did poorly in some tests. Report only where it did well.
- Your technique did poorly in all tests. Modify the test results.
- You find out that someone else already invented the method. Don't cite them.
- Your results were due to an error. Don't report it.

Ways to End Your Career

- Plagiarize.
  - Use figures without permission.
  - Someone else said something better than you can. Use his text instead.
    - This includes former co-authors.
  - Quote someone without making it brutally clear that it's a quote.

Example

* Since most jobs have small memory requirements, relatively fine-grain (or short-term) time-slicing among several memory-resident jobs is distinctly possible.

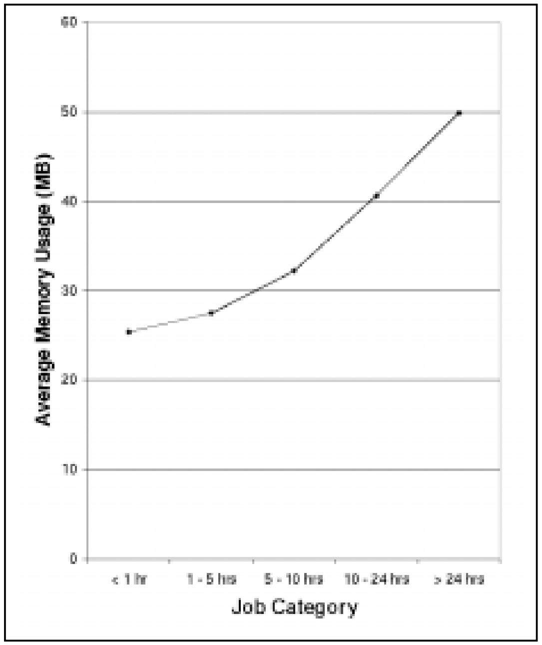


Figure 1

Example

- Since most jobs have small memory requirements, relatively fine-grain (or short-term) time-slicing among several memory-resident jobs is distinctly possible [Setia et al, 1999].

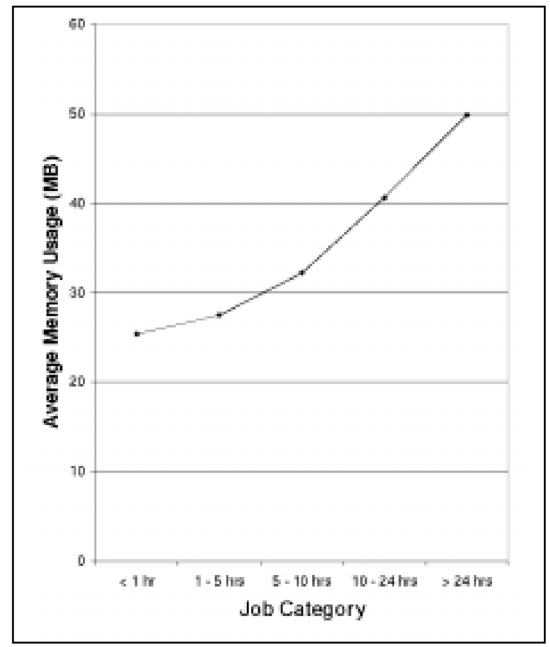


Figure 1 from [Setia et al, 1999]

Example

* [Setia et al 1999] argued that since most jobs have small memory requirements, relatively fine-grain (or short-term) time-slicing among several memory-resident jobs is distinctly possible.

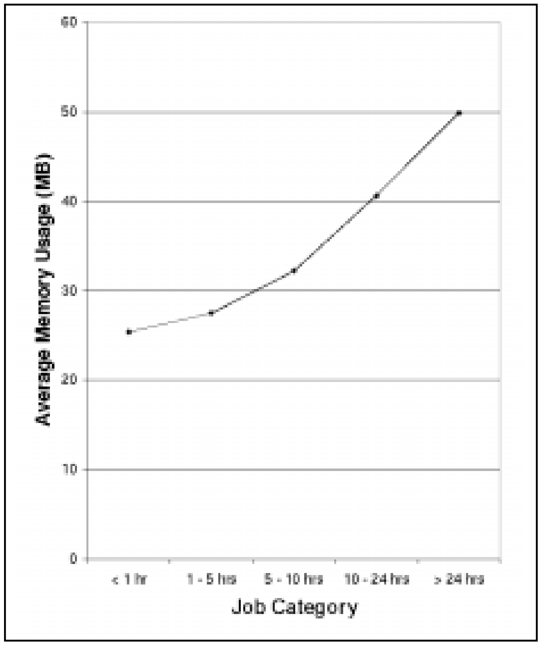


Figure 1 from [Setia et al, 1999]

Example

* [Setia et al 199] have argued: "Since most jobs have small memory requirements, relatively fine-grain (or short-term) time-slicing among several memory-resident jobs is distinctly possible."

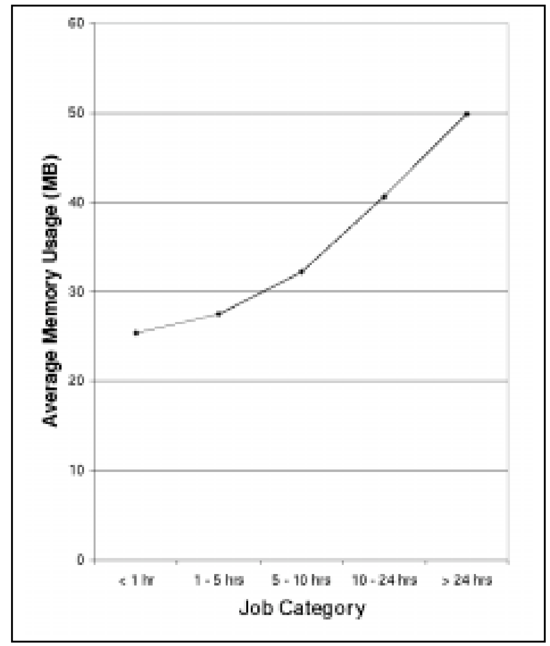


Figure 1 from [Setia et al, 1999] with permission

Example

* Setia et al have argued that the small size of many memory-resident jobs enables fine-grain time-slicing. [Setia et al 1999]
* (This is dangerous)

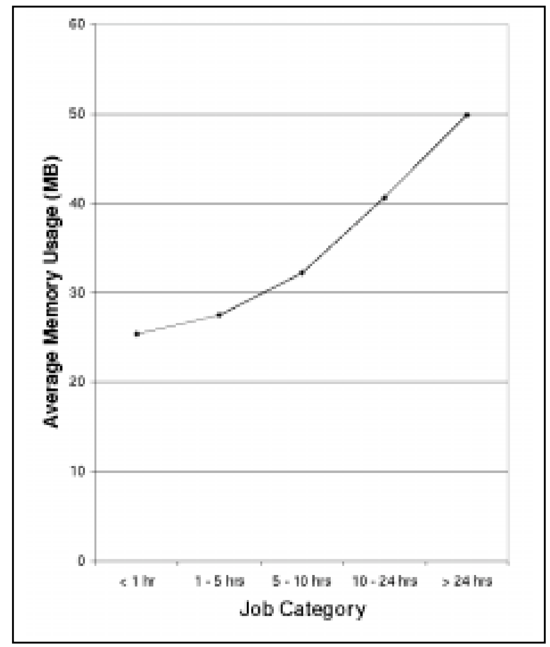


Figure 1 from [Setia et al, 1999] with permission

Reporting Tools

Do Not Use MS Word

* Math looks horrible in Word
* Word does not properly typeset paragraphs of text
* Word does not handle PDF/EPS files well
* Word does not lend itself to long documents or to customization
* Word is not portable

LaTeX

- Nearly all computer science, mathematics, physics, and engineering journals and conferences permit LaTeX
- Many require LaTeX
- Your dissertation should almost certainly be in LaTeX
- If you use Word, you will be noted as someone unable to learn LaTeX

LaTeX has:

* A huge open source community and library of tools
* The best math typesetting anywhere
* The best bibliographic reference system
* Excellent long-document handling (macros, sophisticated style files, etc.)
* Extensibility to presentations, reports...

LaTeX does not have...

* Good unicode handling
  * (though XeTeX is great)
* Good modern font handling
  * (though XeTeX is great)
* A GUI
  * (it's a programming language)

Do Not Use Excel

- Excel is famous for math errors
- Excel's statistics are primitive
  - Use R
- Excel's chart facilities are very poor for scientific publishing, and cannot output to PDF
  - Use R, Mathematica, gnuplot

Official Style

* A long time ago, scientists used to write well.
* Then the Victorian period occurred.
* Administrators and politicians developed a unique style of writing called "Official Style" which was designed to obsfucate and avoid responsibility.

Official Style

* Use I and Me
* Avoid passive voice unless it is awkward
* Do not use a word if there exists a shorter, more common word which means the same thing
    * Utilize → Use

Official Style

* Four experiments were performed utilizing the technique.
* We utilized the technique in performing four experiments.

* I ran four experiments using the technique.

Note to Foreign Students

* Regularly schedule ESL writing tutoring
  * writingcenter.gmu.edu/eslservices.html

* Read all of Strunk and White
  * www.bartleby.com/141/

Making Valid Claims

Two Primary Goals

* Demonstrating that my technique is best
* Demonstrating why my technique is best

Comparisons

* T-Test is the absolute minimum
* Nonparametric tests are preferred (data is rarely normally distributed)
    * Rank all data, do T-test on ranks
* LARGE sample sizes: 50 is minimum
* ANOVA for multiple comparisons
* Bonferroni Correction: $\alpha' = \alpha/N$

Good Tutorial

* Statistics for EC
  * http://www.cis.uoguelph.ca/~wineberg/

Picking a Metric

- Scenario: I want to show that the output of my stochastic optimization technique is better than existing technique A.
- Average best result over N runs?
- How often I found the optimum?
- How often I got within ε of the optimum?

Multiobjective Tests

* Example: Tree "Bloat" in Genetic Programming
* Technique must produce trees which are both highly fit and very small
* Technique A makes smaller trees
* Technique B makes fitter trees
* Which is better? What if N techniques?

Combining to a Single Quality Value

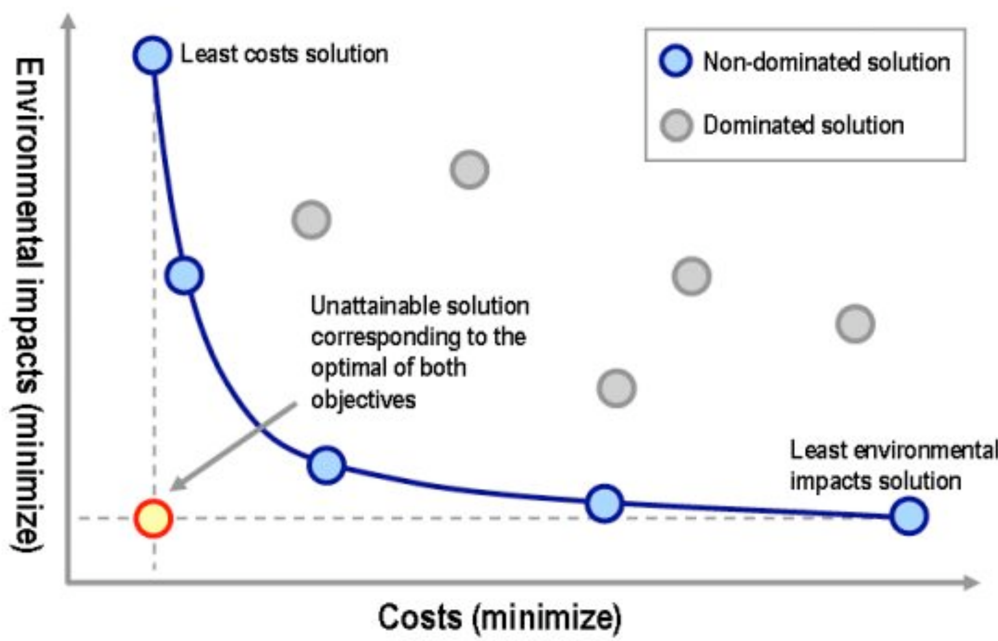
* s = tree size, f = tree fitness, q = quality
* $q = as + bf$ (linear)
* $q = s^a f^b$ (nonlinear)

* Yuck: how do we know what the experimenter wants?

Pareto Optimality

- Solution A pareto-dominates Solution b iff both are true:
  - 1. for all criteria c, A(c) is not inferior to B(c)
  - 2. exists a criterion d for which A(d) is superior to B(d)
- Pareto Front: all pareto-nondominated solutions in your collection

The Front



* 2-dimensional front shown

Comparing Pareto Methods

* Find the Pareto front
* Identify "unacceptable" regions of each criterion
* Trim the Pareto front
* How do we know front-technique A is statistically significantly superior to non-front-technique B? Not easy.

Proving Generality

Testing over multiple test problems

* Scenario: I wish to show that my new constraint satisfaction system beats branch-and-bound techniques.

Testing over multiple problem domains

* MANY test problems
* STANDARDIZED test problems
  * At least ones that others have done in the past so you can be relevant
* REAL test problems
* HIGHLY VARIANT test problems

Train/Test Methodologies

- Scenario: two machine learning techniques A and B each generate rules describing the world based on a limited set of N samples from the world.
  - Neural networks, decision trees, support vector machines, etc.
- How do I verify that technique A has figured out how the world works better than technique B?

Train/Test Methodologies

- Gather a set U of uniformly distributed samples.
- Divide U into two sets, the training set and the testing set.
- Feed each technique the training set.
- Verify the degree to which each technique then got the testing set correct.

Train/Test Methodologies

* Performance on the training set doesn't really matter.
* All that matters is performance on the testing set, demonstrating generalization.

Optimization and Experimental Design

Experimental Design

* Scenario: a large multiagent simulation with parameters with many possible settings and likely significant nonlinear parameter interaction.
* The Scientist's Goal: understanding and verifying the model's parameter space.
* The Engineer's Goal: optimizing the parameter space.

Optimization

* I need to optimize N parameters for my method.
* Local Optima?
* Expensive to perform Experiments?
* Experiments can be performed in parallel?

Hill-climbing

* S ← random solution
* 1. S′ ← tweak(copy(S))
* 2. If S′ better than S: S ← S′
* 3. Go to 1

Simulated Annealing

- S ← random solution, T ← High Value
- 1. S′ ← tweak(copy(S))
- 2. If S′ better than S
  or with P(T, Q(S), Q(S′)): S ← S′
- 3. Decrease T $P(T, S', S) = e^{\frac{Q(S') - Q(S)}{T}}$
- 4. Unless T ≤ 0, Go to 1

Evolutionary Computation Methods

* $P \leftarrow \{ S_1 \ldots S_n \}$ random solutions
* 1. $P' \leftarrow \{ \}$
* 2. Until P' is filled,
  * 3. $P' \leftarrow P' \cup \{ \text{tweak(copy(select}(P))) \}$
* 4. $P \leftarrow P'$
* 5. Go to 1

What does it mean to...

* tweak?
* copy?
* select?
* Assess the quality of a solution?
    * ... how would you do this for multiple objectives?

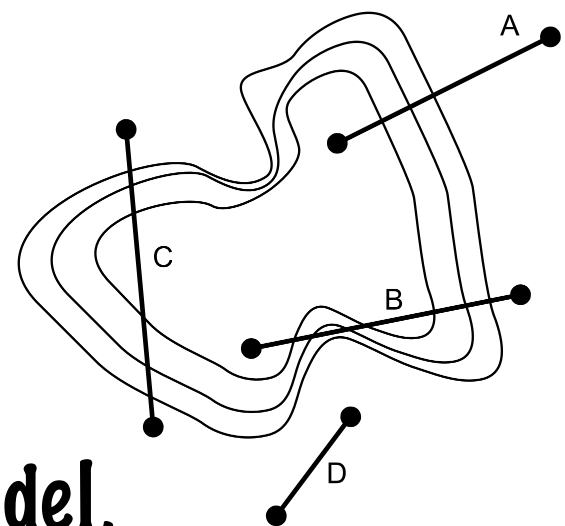
EC Methods

* Genetic Algorithms
* Genetic Programming
* Evolution Strategies
* Friends of the family:
    * Particle Swarm Optimization
    * Ant Colony Optimization

Parameter Search

* Situation: I have a simulation with a real-valued parameter space for which there are complex interactions among the parameters. Testing is costly.
* I wish to test N times to get a feel for what the parameter space looks like.
* How do I focus tests on areas of the space mostly like to be "interesting"?

What is Interesting?



- Interesting is: a steep slope in parameter space
  - Differentiate the space?
  - Take M samples, fit a model, sample more where the model is steep?
  - Iteratively sample along a line between two very different quality samples?

Algorithm

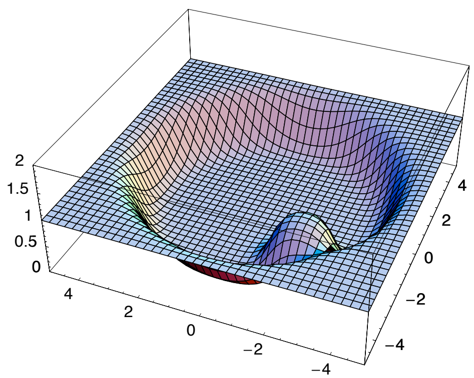
* Generate $P = \{ S_1 \ldots S_m \}$ random samples
* 1. $S \leftarrow$ pick at random from P
* 2. $S' \leftarrow$ select(P), which is close to S and very different in quality from S
* 3. Generate S'' along line segment between S and S'
* 4. $P \leftarrow P \cup \{S''\}$
* 5. Go to 1

Iterated Bracketing (replaces steps 3 and 4)

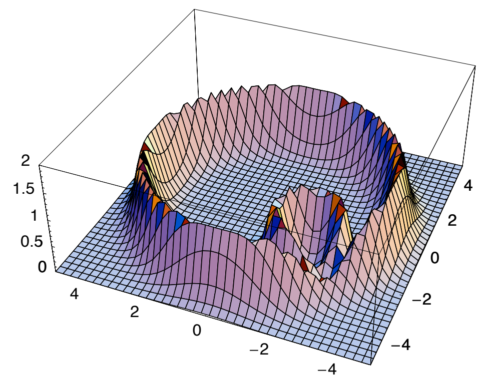
* Given S and S′, iterate W times:
  * Generate S′′ along line segment between S and S′
  * $P \leftarrow P \cup \{S''\}$
  * If $\Delta(S,S'') > \Delta(S',S'')$ then $S' = S''$ else $S = S''$

$$\Delta(S_1, S_2) = \frac{|Q(S_1) - Q(S_2)|}{||S_1 - S_2||}$$

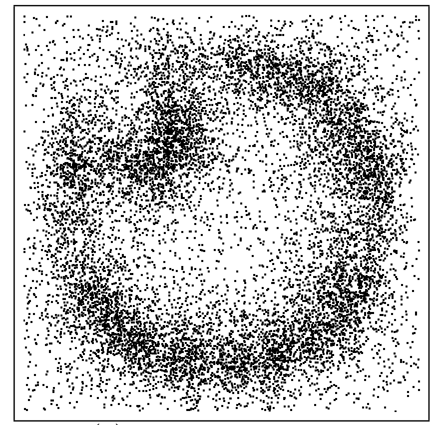
Results



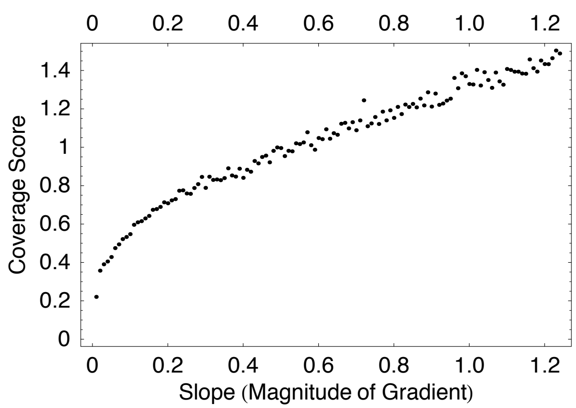
(a) Function $Circ(x, y)$



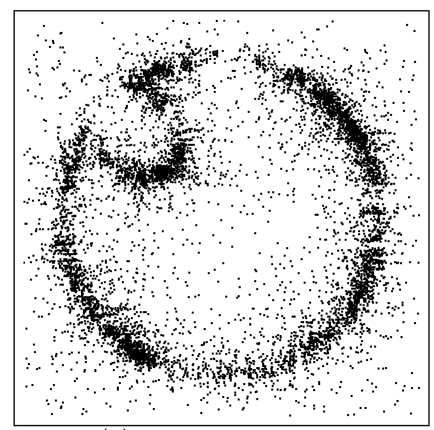
(b) Slope: $||\nabla Circ(x, y)||$



(c) 10000 Iterations, $numBrackets = 1$



(d) Coverage Score by Slope, 10000 iterations, $numBrackets = 1$, average of 50 runs



(e) 2000 Iterations, $numBrackets = 5$