PROJECT 4: CHOOSE YOUR OWN PROJECT 26

GEORGE WANG, MIN XU, SESHADRI MAHALINGAM
gswang.cs61a@gmail.com
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley

August 5, 2010

All of the projects should be possible using what you know right now plus one of the parallel lectures. In all the projects, we've abstracted away much of the domain-specific knowledge, so don't feel like you don't know enough to do a specific project. We hope that you will pick the one that most interests you and learn something special along the way. All of these are proposed as one-person projects, though I think all of them have natural extensions to include multiple people.

1 Machine Learning: Optical Character Recognition

The goal of this project is to become familiar with a simple Machine Learning system. Statistics and machine learning are becoming increasingly important in computer science and are widely used for applications such as spam filtering, robot movement learning, computer vision, natural language processing, etc.

In this project, we explore a part of computer vision: Optical Character Recognition through Naïve Bayes algorithm. OCR is the problem of analyzing a scanned image of a hand-written number. Naïve Bayes is a common statistical technique to allow us to attack this problem.

The Spec and lecture notes for this project are available here: http://www-inst.eecs.berkeley.edu/~cs61a/su10/proj/4-naive-bayes/. A YouTube video of the lecture (low quality) has also been released, and is available here: http://www.youtube.com/watch?v=Odtf_v5FVRY.

2 Facebook Graph API

Facebook has a Graph API which allows us to be able to query Facebook for any public information. This information ranges from who your friends are to what they like.

This is a extremely open-ended project. We've written Scheme 'hooks' into their API, meaning when you call scheme procedures, requests are going to be sent to the Facebook servers.You will learn the basics of writing and authenticating a Facebook application. What you do with it is up to you.

The lecture notes and skeleton code are available here: http://www-inst.eecs.berkeley.edu/~cs61a/su10/proj/4-facebook-graph/

If you do this project, you must develop a readme file with two paragraphs stating what you are doing and how you attacked your problem. If you are working in groups, you must also write another paragraph describing how you divided your project. Note, we are serious when we mean paragraph: please do not turn in an essay.

Note: This is new project, developed especially for this summer. Bug warnings apply.

3 Bioinformatics: Shotgun Sequencing

The Human Genome Project sequenced the vast majority (not all!) of human DNA. To do this efficiently, they employed the technique of Shotgun Sequencing. This technique combined biochemistry and computing to sequence DNA at an unprecedented speed. Essentially, they were unable to look at an entire very long word at once, so they cut it up into tiny interlocking words (much like a jigsaw puzzle), and then reassembled these pieces in a computer. For example, they would cut up abracadabra into something like: abrac, racad, abra, adabra.

Your job with this project is to become familiar with some basic techniques of reassembling pieces of a word, as well as analyze how these techniques apply when we are dealing with DNA sequences instead of words.

The lecture notes are available here: http://www-inst.eecs.berkeley.edu/~cs61a/su10/proj/4-bioinformatics/

Note: This is new project, developed especially for this summer. Bug warnings apply.

4 Sudoku Solver

A 9x9 Sudoku Solver should be possible, despite there being technically 9^{81} possible board configurations. Eliminating all the not-legal ones, you are left with approximately 6.7×10^{21} possibilities, which is still too much for your computer to handle if you approach in a naive manner.

This project is much more like a 61B project than 61A project. You will need to consider how to approach a problem like this, and how to approach it *efficiently*. You, with advice from your TA, will think about the algorithms and data structures required to handle the project.

5 Graphics

In Chapter 2.2.4 of A&S, they introduce the idea of a program that is able to control some basic graphics.

This project is designed to ensure your understanding of Data Abstraction and Higher Order Procedures. There's no domain-specific knowledge (ie, how to get Scheme to draw things), and that's the point of the project. You have to really trust the data abstraction for things to work out.

This is the project that's typically done for this course, but as project 2. Thus, if you are strapped for time, this may be the shortest project available.

6 Your Choice!

Propose something! Turn in a 1-page summary of your plan, containing three components: what you want to do, how you want to do it, and if you're in a group, how you want to divide the work. If you wish to extend one of our projects, we'd be glad to hear it!

If your project is particularly interesting, you can leave a permanent legacy, because we can offer it as one of the suggested project next summer!