

## HW0: Survey and basic concepts

## 1 Student Survey

Please note the following information on your assignment:

1. Which of the following courses have you taken: Differential calculus; Integral calculus; Multivariate calculus; Linear algebra; Probability and statistics; Artificial intelligence; Algorithms; Computer vision; Image processing; Natural language processing; Robotics; Optimization (linear, quadratic, convex, etc.)

Integral calculus (in high school), probability and statistics for engineers, and algorithms.

2. List a few (research/CS/math/whatever) topics that interest you.

Formally, I'm interested in large-scale ML and IR. (I'm an undergrad in Ellen Riloff's NLP group.) Less formally, I'm interested in all sorts of things. As a hobby I'm really interested in programming languages; I implemented a Python compiler in Haskell this summer. I enjoy studying and analyzing algorithms, and in particular, streaming and distributed algorithms. I'm interested in formal methods of computation on both the sequential and massively distributed basis. The list goes on and on.

3. How would you rate your programming skills (1-10, 10 best)? How would you rate your math skills?

Programming: 6 or 7 – I can do the work, but I have no idea what these numbers mean.  
Math: 5 – I'm average. Actually, probably worse than average.

4. What are your goals in this class?

I want to have a broad knowledge of the theater of techniques used in ML. A good background on a number of subjects, and possibly a really deep understanding of one or two subjects. I want to be able to implement, analyze, use, and develop ML algorithms in my research. Mostly, I want a theoretical understanding that allows me to extend the current knowledge about ML to fit it into the work that I'm doing.

5. Please provide a 4-8 character identifier that we can use to post grades only pseudo-anonymously. Use only alpha-numerics (no spaces).

collie

6. Please be sure that you have subscribed to the class mailing list.

## 2 Written Problems

Answer the following questions in 50-100 words each:

1. What is the difference between supervised, unsupervised and reinforcement learning?

**Supervised learning** depends on there being *labeled* training data. Typically, a learning algorithm will use such labels to guide some loss function toward estimating a “best” hypothesis function which is meant to correctly predict the label of any valid input. In contrast, **unsupervised learning** usually attempts to uncover some unobserved structure in unlabeled data, which in particular means that there is no loss or reward functions. Good examples of this include dimensionality reduction and data clustering. **Reinforcement learning**, a bit different from both of those things, actually receives no training examples whatsoever, and instead attempts to find the sequence of reactions that maximizes reward by incorporating some sort of loss or reward function to correspond to the various actions it can undertake.

2. List at least two real-world problems (other than those discussed in class) for each of the categories: supervised, unsupervised and reinforcement. For one of the supervised problems, what is the form of the output and what might be a reasonably input? For the reinforcement learning problem, what might be the state space, action space and reward function?

**Supervised learning** Bayes classifiers can be used to automate role recognition in social graphs (e.g., who are the evil executives in the Enron emails?). SL has also been used in speaker identification for a long time. **Unsupervised learning** Topic models (Blei, Ng, and Jordan) are based in Latent Dirichlet Allocation, and basically posit that words in a document can be described as having been generated as a result of being a part of one of some number of topics that a document is comprised of. Barzilay et al won ACL best paper in 2009 for using **reinforcement learning** to map words and phrases in Windows troubleshooting guides to actions (“click mouse”) and things (“button”). RL also has a pretty long history with motion planning.

3. What is overfitting? What is underfitting? How is overfitting controlled?

When we have trained a learning algorithm that does not transfer well to the general case, it is said to be **overfitted**. When we have a model that fits much too loosely, it is said to be **underfitted**. There are a number of ways to prevent overfitting, but the easiest one is just to keep your model simple.

4. Getting labeled data for supervised learning is expensive. What are some of the approaches to deal with this problem?

The obvious thing to do is to annotate the data only partially. Another approach is to use only data that is really useful, which requires less labeled data. You can use transfer learning, in which you leverage learning about something else to learn about the task that you *really* care about.

## 3 Additional Exercises

The following are true/false questions. You don’t need to answer the questions. Just tell us which ones you can’t answer confidently in less than one minute. (You won’t be graded on this.) If you can’t answer at least 6, you should probably spend some extra time outside of class beefing up on elementary math.

1.  $\log x + \log y = \log(xy)$
2.  $\log[ab^c] = \log a + (\log b)(\log c)$
3.  $\frac{\partial}{\partial x}[5x^2 + 3x] = 10x + 3$
4.  $\frac{\partial}{\partial x} \log x = -\frac{1}{x}$
5.  $p(a \mid b) = p(a, b)/p(b)$
6.  $p(x \mid y, z) = p(x \mid y)p(x \mid z)$
7.  $\|\alpha \mathbf{u} + \mathbf{v}\|^2 = \alpha^2 \|\mathbf{u}\|^2 + \|\mathbf{v}\|^2$ , where  $\|\cdot\|$  denotes Euclidean norm,  $\alpha$  is a scalar and  $\mathbf{u}$  and  $\mathbf{v}$  are vectors
8.  $|\mathbf{u}^\top \mathbf{v}| \geq \|\mathbf{u}\| \times \|\mathbf{v}\|$ , where  $|\cdot|$  denotes absolute value and  $\mathbf{u}^\top \mathbf{v}$  is the dot product of  $\mathbf{u}$  and  $\mathbf{v}$
9.  $\int_{-\infty}^{\infty} dx \exp[-(\pi/2)x^2] = \sqrt{2}$

I wasn't sure about 2 or 9.