Dale Markowitz

Statement of Purpose

Every day, we make complex decisions and optimizations about how we should accomplish tasks: how we can juggle problem sets while cutting down growing towers of laundry while at the same time remembering to absorb nutrients. However, if the conscious decisions we make about how to accomplish concrete tasks is difficult, then the problem of controlling our more abstract human needs is harder still. Tiredness, boredom, hunger, stress, distractedness—these and our entire plethora of physiological and emotional states are fundamental in determining how well we accomplish all kinds of goals, yet in this domain, technological innovation has been notably absent. Today, however, as microcontrollers shrink well below the perimeter of a fingernail, we are finally able to measure and interpret the delicate space of human physiological features. We can already see technology take grasp of once-hidden human metrics: wrist-worn heart-rate monitors paired with gyroscopes and accelerometers track our every step to give us a holistic measure of our daily activity. Wearable and wired galvanic skin response meters can recognize when we're startled or stressed and gently remind us to relax. Perhaps one of the most exciting biological sensors that is today just over the horizon is an accurate, portable EEG (electroencephalograph), that is able to read tiny voltage oscillations on our scalps and give us a snapshot of our thoughts. Paired with sophisticated machine learning algorithms, these data can give us real-time insight into our thoughts and feelings, potentially allowing us better control and understanding of our mental processes. At Princeton, I have already begun exploring how we can use portable EEGs to take control of our sustained attention, and it is work like this that I hope to continue in graduate school.

I have always had an interest in creating physical objects, and began playing with electronics during my first year of college. I've built projects atop a host of hardware, from 8-bit microcontrollers to ARM-based Linux Computers to a network of wireless devices. My personal projects have involved not only electronics and circuit board design but also designing and printing 3D components. One of my favorite projects was a necklace that, powered by a microcontroller, tri-color LED, and microphone, lit up in response to “ultrasound.” It alerted me that my cellphone was ringing by glittering in response to a 20,000kHz tone pulse (above the frequency of human hearing) embedded in my cellphone's ringtone. Another personal project I enjoyed creating was a single-channel EEG, consisting of a microcontroller and several instrumentation amplifiers. Although the signal from my home-brew EEG was noisy and the electrode setup was precarious to say the least, the device allowed me to see my brain produce classic alpha waves—modest first steps towards mind-reading!

Hoping to upgrade from my crude­­­ setup, I decided to begin my senior thesis project working with research-grade EEG's. This year, with Professor Kenneth Norman at the Princeton Neuroscience Institute, I am building a Brain-Computer Interface for improving sustained attention. fMRI studies have shown that giving participants real-time feedback on how well they are concentrating during an attention-based task improves their performance on these tasks, even after the neurofeedback is removed. The hope is that these improvements could last even beyond the lab, and that this neurofeedback attention training could be used to help patients better concentrate in their everyday lives. Results from fMRI studies are encouraging, but fMI’s are expensive and impractical for use outside of a neuroscience lab. EEG’s, on the other hand, are getting more inexpensive and accessible by the day, and could act as a practical platform for attention training. In my research, I am trying to create such an EEG platform by applying machine learning techniques to data from inexpensive EEG devices. **Thus far, applying (some algorithm) has given us impressive results, allowing us to classify how well a participant is concentrating with an accuracy of (x) (note: not done yet).** That human thoughts can be decoded from a hundred-dollar, portable EEG headset—and indeed early results show that they can—is a prospect I find extremely exciting, allowing BCI's to evolve from lab-bound, physiologically intrusive devices to a technology that could one day be as commonplace as a Fitbit.

While my current research sits at the boundary of two budding fields—Machine Learning and Neuroscience—I have also spent my time at Princeton doing “classical” Computer Science research, in both networking and content distribution. Both during the first semester of junior year and over the summer, I worked with Professor Vivek Pai to design an Internet accelerator platform for developing world countries. Internet access in developing world countries is both extremely slow and expensive, due largely to a lack of physical infrastructure. As a result, web caching—wherein a central web cache stores data fetched by users in the network and provides subsequent requesters with a cached copy of that data—can be effective in decreasing the lag users experience in these regions. Paired with another strategy called “bandwidth shifting,” in which an accelerator node located in a high-bandwidth region compresses data for and manages the freshness for accelerator nodes in low-bandwidth regions, caching can be more effective still. Professor Pai and I were curious to know if these sophisticated acceleration techniques could run on a cheap, single-board ARM computer, which could ultimately provide a simple plug-and-play platform for developing world network accelerators. In my research, I attempted to create such a platform on a BeagleBone Black (BBB), a $45 Linux computer no larger than a credit card. After testing the BBB's bandwidth in serving disk requests, I determined this cheap computer would have the chutzpa to serve as a web cache, and began developing caching/bandwidth-shifting software for it. Using Node Javascript and C, I created a portable software stack that could run on the BBB, resulting in *BeagleCache: A Low-Cost Caching Proxy for the Developing World*. I hope that this work will serve me in my future research; the BBB is a great platform for many sensor projects that require computation-intensive processing, and it is my hope that networking these stripped-down devices has provided me insight into building wireless, connected sensor devices.

During the second semester of my junior year, inspired by courses I had taken in Networking and Distributed Systems, I decided to pursue a research project in Software-Defined Networking. In my Networking class, I learned that interdomain routing is a major roadblock in allowing Internet Service Providers (ISP's) to provide better service to their customers. The Internet is distributed across many ISP's who are fundamentally in competition with one another, but who must nonetheless work together to deliver data to their customers. It is routing performed between these ISP's that often leads to inefficiencies. For my research project, inspired by new techniques in Software-Defined Networking—that is, networking technology in which backbone Internet routers are able to be reprogramed on the fly—I decided to design router software for ISP's that allows them better control as to how their data is routed across interdomain boundaries. In particular, I built a software-defined overlay network for ISP’s to run on their routers. This software gives each ISP the illusion that they own and control all of the routers and links in the Internet. Routers controlled by other ISP’s are hidden behind the abstraction of a virtual link, and their routing decisions are thought of as variations in congestion and performance of this virtual link. In this way, ISP’s can monitor the performance of the “virtual links” on their network and change their routing policies dynamically to provide data more efficiently and quickly to their customers. I built this software in Pyretic, an SDN language embedded in Python that communicates with network switches via the OpenFlow protocol. I was then able to test the software by simulating a virtual network created in Mininet. I was excited in this project to be able to play with a new technology that will soon—in my opinion—change the landscape of how networks are built. Furthermore, studying networking on the whole redefined the way I thought about systems design. It forced me to consider how unreliable, volatile distributed systems can be built to provide reliable service.

I have explored many dimensions of Computer Science here at Princeton—from systems to mathematics and theory to application—and have hopefully gained a more insightful view of Computer Science as a result of it. As I narrow my focus in graduate studies, I am interested in studying sensor devices not only because I am excited to guide how they will transform our everyday lives, but also because this field integrates many parts of the field. Data analysis will require careful application of Machine Learning algorithms, but implementing those algorithms efficiently on real, fallible hardware will force me to apply what I have learned from network and systems design. I hope that my background designing real systems and willingness to explore new fields will **help me succeed in graduate school (insert extremely compelling closing here).**