Teaching Statement  – Matthew R. Gormley

My passion for teaching stems from a desire to encourage diversity in computer science, to create high-quality new curriculum, and to build excitement for CS among majors and non-majors. I love to learn and build new things; teaching affords me this opportunity and in part motivated me to pursue a doctorate.

Building excitement for computer science and the opportunities for problem solving that it affords is critical to both successful CS education and to encouraging gender, racial, and socio-economic diversity in computer science. By exposing CS students, non-CS majors, and high schoolers to real-world problems and the theory needed to solve them I hope to continue to address these goals. Three facets of computation direct my teaching methodology: (1) computation is a tool for solving real world problems, (2) its use requires theoretical understanding, and (3) system building is enhanced by expertise in the area of application. These three facets are embodied in my areas of research, Natural Language Processing (NLP) and Machine Learning (ML), but are also true of everything from compilers to human-computer interaction.

My teaching experience has ranged over curriculum development, lecturing, mentoring, tutoring, studying recent education research, and academic outreach—and has covered areas such as introductory computer science for non-majors and graduate level machine learning. My experiences divide into four areas:

**Lecturing & Course Design**  At Johns Hopkins University, I co-lectured for and redesigned the curriculum of a course on the fundamentals of practical computing for non-CS majors. Additionally, I developed a new graduate course in machine learning, gave background material lectures, and guest lectures in another graduate course.

**Conference Tutorial**  I presented an advanced tutorial on *Structured Belief Propagation for NLP* to an audience of international researchers and experts in the field of natural language processing (NLP).

**Outreach**  I spent two years on the organizing committee of *NACLO* visiting local Baltimore high schools to teach students about computational linguistics and recruit for the annual Olympiad.

**Mentoring**  I supervised two undergraduates in NLP research via weekly meetings which led to a publication. For over four years, I have volunteered weekly to tutor high schoolers who need extra help outside the classroom.

In these roles as an educator, I have pursued my passion for developing high quality CS curriculum and encouraging diversity in computer science. In all these experiences my teaching has emphasized real-world examples, unification of theoretical concepts, and an understanding of the application area.

1 **Lecturing & Course Design**

While at Johns Hopkins University, I acted as a co-lecturer and curriculum developer for an introductory computer science course for non-CS majors, *Foundations of Practical Computing* (JHU CS 103). The students came from many different backgrounds including economics, molecular biology, and cognitive science. The course was not an intro programming class, but instead focused on abstractions and computational thinking by exposing students to many different programming environments. The primary skill we taught was *how to learn* a new computing environment. My belief is that this skill, more than mastery of a single programming language, would benefit non-majors entering a discipline where computation could serve them for years to come. The environments we covered included a toy-language (Scratch), several declarative
languages (HTML, SQL, regular expressions, Amazon Mechanical Turk), procedural languages (Javascript, Google Spreadsheets, Python, R), and pseudo-code. Our course design emphasized the connections between these languages, but also their individual specializations. This ambitious approach was backed by a heavy emphasis on problem solving. Class time began with a lecture and ended with students solving a simple problem in groups. We assigned two short homework assignments per week to keep up with the rapid pace of the course material. The course responsibilities and lecturing were evenly divided with a fellow PhD student, Ann Irvine, which afforded us the time to develop high quality new curriculum. Graduate students at Hopkins are only rarely given full autonomy to teach and design a course. We redesigned over half the curriculum, creating new topics, lectures, homework assignments, and exams.

My teaching experience has also extended to two graduate level machine learning courses at Johns Hopkins. Alongside professor Suchi Saria and another TA, Michael Paul, I worked to design a new graduate level machine learning course on probabilistic graphical models, *Machine Learning in Complex Domains* (JHU CS 476). My contributions were primarily in developing new homework assignments, exams, and supplemental lectures. In the supplemental lectures, I covered background material in complexity theory, algorithms, optimization, and probabilistic modeling. This material was covered in three lectures in the early weeks of the course. I returned the following year to enhance these supplementary lectures. The course was project based. We developed three large homework assignments and a final exam in the same style. The assignments mirrored a real-world approach to research: most problems consisted of a theoretical component, a programming deliverable, and empirical questions. The problems covered a wide array of machine learning methods: Bayesian inference, the Junction Tree algorithm, parameter sharing and estimation, Gibbs sampling, variational inference, structure learning, CRFs, and Gaussian random fields. Each of these substantial assignments was intended to give the students insight into how to go about designing a course project of their own. Their course projects reflected the diversity of the research interests of the students. Notable applications were clinical decision making for ICU patients, human gesture recognition with a Kinect, and ancestry prediction from genomic sequences.

Over several years, I also gave six guest lectures for another graduate course, *Machine Learning* (JHU CS 475) taught by professor Mark Dredze. For two consecutive years, I lectured on approximate inference and topic modeling; this lecture has since been incorporated back into the curriculum for the course. The following year, I gave two lectures on HMMs, CRFs, and belief propagation. This year, I developed two new lectures on neural networks and deep learning that are to be included in future years of the course.

## 2 Conference Tutorial

I presented a three hour tutorial with professor Jason Eisner on the topic of *Structured Belief Propagation for NLP* to an audience of over 60 experts in the field of computational linguistics. The venue was ACL, the largest international conference on NLP. Probabilistic models form the basis for most state-of-the-art methods in statistical NLP. Yet the models are often overly simplified in order to keep learning and inference tractable. Structured belief propagation and its modern training methods address exactly these problems. The tutorial covered both theory and practice, to show how BP could be applied to new problems. One of the greatest challenges was designing the tutorial to be accessible to both first year graduate students with limited background in machine learning, but also beneficial to researchers who have spent years working with BP. We accomplished this by teaching even the most advanced topics using the simplest examples, so that a newcomer to BP could always keep up. We also designed the content such that even an expert on BP would come away with (1) new theoretical connections to other algorithms and techniques and (2) a better practical understanding of how to cope with the approximations made by BP. Our greatest measure of success was that we received praise from audience members who were experts on the topic of structured BP and from those who had never seen it before.
3 Outreach

The North American Computational Linguistics Olympiad (NACLO) is a puzzle solving competition for middle and high school students. The top students go on to represent the U.S. in the International Linguistics Olympiad. As one of the participating universities, Johns Hopkins hosts local students annually. I spent two years on the organizing committee of NACLO at Hopkins. I visited local Baltimore high schools to teach students about computational linguistics and recruit for the annual Olympiad. Leading up to the Olympiad we conducted practice sessions, where we taught the students strategies for solving the types of linguistic puzzles they would see in the competition. In the years I participated, our team of five increased participation to 79 students, making JHU one of the largest sites for NACLO in the U.S. While participation alone was not the goal, it was a clear testament to our efforts in garnering excitement for linguistics and computation.

For three semesters, I participated in a seminar for professors and graduate students interested in computer science education. We read and discussed papers on various topics ranging from evaluation, massive online open courses, the flipped classroom, core curriculum for CS, and computational thinking. A topic which received considerable attention was encouraging diversity in CS with a focus on practical changes within the JHU computer science department.

4 Mentoring

I have mentored two undergraduate CS majors with an interest in NLP research. Justin Snyder worked with Mark Dredze and I to develop a visualization framework for topic models. This work resulted in a first author publication for Justin and a demo at a major NLP conference (NAACL 2013). For a semester, we met weekly to discuss his progress, software design, and the goals for the visualization framework. Sharon Li worked with Jason Eisner and I to develop a novel approach to unsupervised learning. We met weekly and built a prototype system.

The College and Career Program (CCP) is a tutoring and mentoring program for Baltimore city public high school students who need extra help outside the classroom. The last four years, I volunteered weekly as a tutor at CCP with a focus on math, science and Spanish language skills. One of my mentees, who I worked with weekly on chemistry has gone on to become an engineer at a major research university, where he is now actively engaged in undergraduate research.

5 Looking Forward

My experiences have shaped my values in computer science education. I am excited to continue pursuing diversity through CS outreach. I hope to continue developing new curriculum that emphasizes practical computation for non-CS majors. Within a CS department, I would be thrilled to have new opportunities to teach foundational courses in computer science such as data structures and algorithms and programming courses at any level. My experience working as a software engineer for two years would help me provide real-world context to these courses. Further, I would enjoy teaching advanced undergraduate or graduate courses in artificial intelligence, machine learning, or natural language processing. Finally, I look forward to mentoring undergraduate and graduate students as they pursue exciting research in natural language processing and machine learning applied to other fields.