What Am I Looking For in a Prospective Student?

TL;DR (Even though this is only a page and a half long...)  

My work lies in the intersection of theory and machine learning.

Any future student should consider me only if s/he is willing to do theoretical work and build the skills of proving lemmas and theorems, and has the basic knowledge of analyzing randomized algorithms.

I do NOT care about prior work detailing “we’ve used deep learning to cure cancer.” I DO care about your conversion of the original problem into a mathematical model, and the reason for choosing one technique over another.

Do NOT send me your generic email. Send me an email that should inform me as to why we would be a good advisor-advisee match.

I would describe myself as someone who’s interest lie in the intersection of theory and machine learning. While I am willing to be stretch further in each direction (if I were to co-supervise with someone from the theory side / the ML side), the core of what I consider to be a publication-worthy work is a (i) sound and (ii) interesting theorem. To this theorem one can add experiments, more theorems, incentive-compatibility design (game theoretic view) and more, however – I do expect my students to have the capability to prove a theorem and to design a novel algorithm on their own, and upon graduation. Namely, this is a trait that one needs to build up to, slowly, starting with small HW-like questions and building to a whole research project – but this is the path I intend on taking my students. And so - the potential for theorem proving is the key property I look for in a prospective student. In particular, if you wish you grad-studies to involve only programming work and only run experiments – we are a poor match unfortunately.

As core prerequisites, I require my students to be familiar with basic algorithms, asymptotic notation (big-O) and probability. Any prior knowledge about differential privacy (my current main field of research) is encouraged. Any prior knowledge with the theory of machine learning (PAC-learning, [convex-]function optimization, any theorems about optimization or proven approximations, clustering) is encouraged. Any prior knowledge with statistical theory is encouraged (again, proofs, not “I know how to use SAS”).

Knowledge in applicable machine learning can be neat, but I am far more curious as to the learning model you’ve used: Do you understand the model? Can you tell me when it does and when it doesn’t work? Can you tell me if you tried other models and what did they fail to capture? How rigorously did you abstract the problem you were solving? Are there interesting open problems about the technique you’ve used (not its performance over a specific dataset, but rather its guaranteed performance over all datasets of certain nature)? What I mainly care about is the process of converting the abstract problem into a formal mathematical conjecture / problem that can be written on a whiteboard. What problem statement did you end up deciding on, and why?
So, if you are considering emailing me about advising your PhD / MSc studies – do NOT send me your generic email which you send to any professor in CS. Send me an email that informs me as to why you and I might be a good advisor-advisee match. Here are some suggestions as to what will pique my interest:

- **Prior experience in research:** Did you do an undergrad / intern / MSc research project? What was it about?
  Do not elaborate about the project’s data and concrete results (“we’ve discovered 7 distinct speech patterns in a corpus of 591,132 sentences”), but rather on the research question and the model you’ve used to analyze the data (“In analyzing structured sentences, we found that logistic time-series performs roughly as well as a neural-network; and managed to prove that if the input was generated according to model X then logistic-time series actually retrieves the model parameters”). Why was this model better than other models? Did you just plug-in a neural-networks package, or did you understand the underlying phenomena in your analysis? Did you encounter any modelling difficulties, and if so – how did you overcome them?

- **Prior educational experience:** What courses did you take? What courses in theory / math / ML did you take? What courses will you want to take?
  (Undergrads:) Was there a particularly project / HW task that you solved in a particularly interesting way? That made you do serious theory work?
  (MSc-students:) What high-level algo/ML classes did you take and what did they cover? (LPs, random approximation algorithms, lower bounds, Hoeffding/Chernoff-bounds, etc)
  Did any of those courses make you do substantial theory work on your own (e.g. prove something from scratch, design an algorithm to solve a specific problem) and how did you rise to the challenge?

- **Any prior knowledge about privacy in general / differential privacy?**
  If you’ve done anything at all in differential privacy, which admittedly remains somewhat of a niche field, I would like to know about it.

- **What problem / type-of-problems do you consider interesting?**
  It shouldn’t be a big problem (P=NP, proving neural-networks bounds etc). It can be a small problem, and presumably about something you know some background material about. A follow-up question on something you worked on is great.

- **Finally, do NOT waste space listing all the programming languages you are fluent with.** All I need is someone who can code a few experiments, even in Python/MATLAB/R (let alone C or Java). Even if you have no programming skills, I could be interested in advising you (a pure math background).