CSE 250B: Machine learning **Winter 2019**

Programming project 1 — Prototype selection for nearest neighbor

One way to speed up nearest neighbor classification is to replace the training set by a carefully chosen subset of "prototypes".

Think of a good strategy for choosing prototypes from the training set, bearing in mind that the ultimate goal is good classification performance. Assume that 1-NN will be used.

Then implement your algorithm, and test it on the MNIST dataset, available at:

`http://yann.lecun.com/exdb/mnist/index.html`

On the due date, upload (to `gradescope`) a **typewritten** report containing the following elements (each labeled clearly).

1. *A short, high-level description of your idea for prototype selection.*

   A few sentences should suffice. These should be crystal clear: they should communicate the key idea to the reader.

2. *Concise and unambiguous pseudocode.*

   (Please do not submit any actual code.) Once again, clarity and conciseness are of the essence. Your scheme should take as input a labeled training set as well as a number M, and should return a set of M prototypes.

3. *Experimental results.*

   A (clearly labeled) table or graph of results showing classification performance on MNIST for a few values of M, including $M = 100, 500, 1000, 5000, 10000$. In each case, you should compare the performance to that of uniform-random selection (that is, picking M of the training points at random). For any strategy with randomness, you should do several experiments and give error bars – give all relevant details, including the formulas you used for computing confidence intervals.

   The pseudocode and experimental details must contain all information needed to reproduce the results.

4. *Critical evaluation.*

   Is your method a clear improvement over random selection? Is there further scope for improvement? What would you like to try next?