Lecture 2

* Structured label
  + {1,2,…,K}^N
* Multiclass
  + {1, 2, 3, …, K}
  + Only 1 class per instance
* Multilabel
  + Multiple labels per instance
* Questions
  + How to define space
  + Pick the right model
* Hypothesis space
  + Function space (input -> output)
* For n binary input function, there are 2^n^n possible functions to define (count all possible output configurations)

Lecture 3

* For label space Y and instance space X, there are |Y|^|X| possible functions f(x)
* The considered functions in this space is the hypothesis space
  + E.g., hypothesis space for all Boolean functions of 4 variables could be only conjunctive functions: x1 and x2, x1 and x3, …
  + E.g., m-of-n rule functions
  + E.g., neural networks, decision trees, nested collections, etc.
  + E.g., Linear classification
* Hypothesis: a specific function in the hypothesis space == model
* Goal: find best hypothesis in hypothesis set that fits the data
  + Hope it generalizes well to data never seen before
* Overfitting
  + Due to high complexity of model
  + Making choices based on very little data
  + Good performance on training data, but does not generalize well to testing data
* Prevent overfitting
  + Simpler, less-expressive models
  + Regularization
    - Promote simpler models – quantitatively define what a simple model is first
  + Data perturbation – add noise in training
  + Stop the optimization process earlier – don’t stop at the best one (that fits perfectly on training data)
* How to learn: how to find a good hypothesis/model form the hypothesis space?
  + Determine a hypothesis space.. use brute-force to find the best hypothesis/model/function
  + But the hypothesis space could be infinite
    - Use gradient descent, or other techniques
    - Local search
      * Start with a linear threshold function (random?)
      * See how well the function is and compute a gradient
      * Correct it
      * Repeat until converge
* K-Nearest Neighbor
* Algo
  + Learning: store all training instances
  + Predication of an input instance x: Average of k nearest neighbors to x (add a distance weight)
* K is the hyperparameter
  + A model with hyperparameter k (and specific way of computing distance (e.g., p-norm)) is a hypothesis in the k-NN hypothesis space
* Distance
  + 1-norm: Manhattan distance
  + 2-norm: Euclidean distance
  + Hamming distance: Number of features that have a different value (used in discrete cases)