Lecture 2 - Chapter 5: Machine Learning Basics

- (One) Definition of Machine Learning
    - “A computer program is said to learn from experience E with respect to some class of tasks and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.” p84
- Tasks
    - classification
    - regression
    - transcription
    - density or probability function estimation
- Performance Measures
    - not always obvious which measure is best
    - not always feasible to implement / compute
    - training set vs test set
    - Examples:
        - mean squared error
        - loss functions (hinge loss, logistic loss)
- Experience
    - dataset
        - “A dataset is a collection of many objects called examples, with each example containing many features that have been objectively measured.” p89
            - e.g. features for object recognition could be the brightness of the pixel in each image
        - Unsupervised learning - no labels on training data
            - best example: density estimation (learn the probability distribution that generated the data)
            - learning features and representations are also good examples
        - Supervised learning - labels on training data
            - good examples are classification and transcription
        - supervised vs unsupervised - not a always clear distinction when it comes to stating the machine learning problem
            - e.g. learning joint distribution of a vector in R^n
- Classic Example: Linear Regression
    - Task: given training data **x** with labels **y** , find a vector **w** so that **y** = **w**^T **x**
        - not always possible
    - Performance Measure: mean squared error (no explicit regularization)
        - use the L2 norm (euclidean distance) to measure the average squared distance between **w**^T **x** to **y**
    - Experience:
        - **x**, **y**

    - Solution: boring algebra to solve for where the gradient with respect to **w** of the mean squared error is equal to zero. good to note that the gradient appears again here and is ubiquitous in deep learning
- Generalization
    - we want the generalization to be low as well
        - i.e. what will our error be on new data?
        - good to note assumption that training and test data is i.i.d.
    - Underfitting
        - "How low is the machine learning algorithm expected to drive the training error?" p95
    - Overfitting
        - "How big is the gap between training and test error expected to be?" p95
    - Capacity
        - neat idea of capacity not in the chapter: Rademacher Complexity
            - http://en.wikipedia.org/wiki/Rademacher_complexity
    - "Machine learning algorithms will generally perform best when their capacity is appropriate in regard to true complexity of the task they need to perform and the amount of training data they are provided with. Models with too low capacity are unable to solve complex tasks. Model with high capacity can solve complex tasks, but when their capacity is too high they may overfit." p96. see figure on p97
- Validation and Cross-validation:
    - get more out of the your data set by training and validating on many partitions.
        - note: validation sets are different that test sets since the learning algorithm nevers train on any examples from a test set
- Point Estimator
    - A parameter, vector of parameters, or function chosen as a representative, e.g. for a distribution
        - usually used for prediction, e.g. in the case when we want to predict label **y** from vector **x**
    - bias of an estimator
        - Example: mean of a gaussian
        - error from bias exists even if the training set is perfectly representative of population distribution.
        - perhaps more closely related to underfitting
    - variance of an estimator
        - variance measures the sensitivity of the estimator to particular samples of data
        - think overfitting
    - trade off of bias vs variance
        - mean squared error
            - bias^2 + variance
- Maximum Likelihood Estimator

    - suppose we have a sample **X** ~ P(**X**) where P is the true data generating distribution. Now consider a family of functions P parametrized by θ. Now we may consider P(**X**; θ), i.e. the probability of observing sequence **X** given parameters θ.
    - Maximum likelihood estimator is the theta that maximises P(**X**; θ)
    - Consider the case of supervised learning. Now we may be interested in the theta that maximises P(**Y**|**X**;θ)
    - Maximum likelihood has the property of consistency, i.e. that as the sample size grows, the estimation of the true parameter θ improves
- Maximum A Posteriori Estimator
    - less variance than ML estimator at cost of more bias. estimator seeks to employ data from outside the training set, i.e. by choosing a prior. usually the prior distribution favors smoothness and simplicity
- Regularization
    - standard ML and MAP estimators cannot always be readily applied in machine learning problems. However we can still reduce variance to combat overfitting.
    - Regularization, similar to the prior, introduces information outside the training set about which solutions are preferred, e.g. to constrain model capacity to combat overfitting. In fact many regularizers can be interpreted as priors.
    - e.g. minimizing the norm of a hyperplane used for linear classification
- Supervised Learning
    - non-deep learning example: SVMs and Kernel Tricks
    - use kernels to find linear separators in higher-dimensional space which project to nonlinear separator in original space
    - some deep learning algorithms can be said to be SVMs with learned kernels
        - important because in usual kernel SVMs, kernel is constant
- Unsupervised Learning
    - non-deep learning example: PCA
    - a linear transformation of data such that the resulting covariance matrix is diagonal, i.e. the elements are mutually uncorrelated
    - also allows one to identify the dimensions which account for the most variance in the data and thus reduce dimensionality by considering only the most impactful dimensions
    - deep learning may achieve similar effect by complex nonlinear transformations
- Curse of Dimensionality & Local Generalization
    - see figure on p130
    - smoothness prior
- Manifold learning
    - “Manifold learning algorithms assume that the data distribution is concentrated in a small number of dimensions, i.e., that the set of high-probability configurations [of a very high dimension set of parameters] can be approximated by a low-dimensional manifold.”
    - example: image recognition