ECE 5424: Introduction to Machine Learning

Topics:

- Midterm Review

Stefan Lee

Virginia Tech

Format

- Midterm Exam
  - When: October 6th, class timing
  - Where: In class

  - Format: Pen-and-paper.
  - Open-book, open-notes, closed-internet.
    - No sharing.

  - What to expect: mix of
    - Multiple Choice or True/False questions
    - “Prove this statement”
    - “What would happen for this dataset?”

  - Material
    - Everything from beginning to class to Tuesday’s lecture

How to Prepare

- Find the “What You Should Know” slides in each lecture powerpoints and make sure you know those concepts
- This presentation provides an overview but is not 100% complete.
- Review class materials and your homeworks.
- We wont ask many questions you can just look up so get a good nights rest and come prepared to think.

Summary of Topics Covered

- K Nearest Neighbor Classifier / Regressor
  - Distance Functions (L1, L2, Mahalanobis)
  - Weighted k-NN & Kernel Regression
- Statistical Estimation
  - Basic Probability
    - Random Variables, Bayes Rule, Chain Rule, Marginalization, Independence, Conditional Independence, Entropy, KL Divergence
  - Maximum Likelihood Estimation (MLE)
    - General MLE strategy
    - Bernoulli
    - Categorical
    - Normal/Gaussian
  - Maximum A Posteriori (MAP)
    - Effect of Priors
    - Conjugate Priors
      - Bernoulli * Beta = Beta
      - Categorical * Dirichlet = Dirichlet
      - Gaussian* Gaussian = Gaussian

Summary of Topics Covered (Cont'd)

- Linear Regression
  - Ordinary Least Squares
  - Robust Least Squares and Ridge Regression
- Naïve Bayes
- Logistic Regression
  - Regularized Logistic Regression

- General Machine Learning Know-how
  - General Train/Val/Test Strategy
  - Underfitting / Overfitting
  - Error Decomposition
    - Modelling, Estimation, Optimization, & Bayes
    - Bias / Variance Tradeoff
  - Model Classes
  - Algorithm Evaluations and Diagnostics
    - Loss Functions, Confusion Matrices, ROC Curves, Learning Curves, Cross Validation
  - Curse of Dimensionality
  - Generative vs. Discriminative Models

Summary of Topics Covered (Cont’d)

- Other Important Mathematic Concepts
  - Vector Algebra
  - Basic Calculus
  - Convexity / Concavity
  - Gradient Descent / Ascent

Know Your Models: kNN Classification / Regression

- **The Model:**
  - Classification: Find nearest neighbors by distance metric and let them vote.
  - Regression: Find nearest neighbors by distance metric and average them.
- **Weighted Variants:**
  - Apply weights to neighbors based on distance (weighted voting/average)
  - Kernel Regression / Classification
    - Set k to n and weight based on distance
  - Smoother than basic k-NN!
- **Problems with k-NN**
  - Curse of dimensionality: distances in high d not very meaningful
  - Irrelevant features make distance != similarity and degrade performance
  - Slow NN search: Must remember (very large) dataset for prediction

Know Your Models: Linear Regression

- **Linear model of Y given X:**
  - **Assume:** $Y\,|X = x_i \sim N(w^T x_i, \sigma^2)$ then $w_{MLE} = argmax\; P(D \mid w) = argmin \sum (w^T x_i - y_i)^2 = (X^T X)^{-1} X^T Y$
  - Another name for this method is ordinary least squares or OLS.
- **Other Variants:**
  - Robust Regression with Laplacian Likelihood ($Y\,|X = x_i \sim Lap(w^T x_i, \sigma^2)$
  - Ridge Regression with Gaussian Prior (w $\sim N(o, \tau^2)$)
  - General Additive Regression
    - Learn non-linear functions in the original space by solving linear regression in a non-linear space i.e. $Y\,|X = x_i \sim N(w^T \Phi(x_i), \sigma^2)$
    - Example $x_i = [x_1, x_2, x_3]\;\; and\; \Phi(x_i) = [x_1, x_2, x_3, x_1 x_2, x_1, x_3, x_2, x_3]$
- **Problems with Linear Regression**
  - $(X^T X)^{-1}$ may not be invertible (or is huge!)
  - OLS is not particularly good with outliers

Know Your Models: Naïve Bayes Classifier

- **Generative Model $\boldsymbol{P(X\,|Y)\,P(Y)}$:**
  - Optimal Bayes Classifier predicts $\mathrm{argmax}_y\, P(X\,|Y=y)\,P(Y=y)$
  - Naive Bayes assume $P(X \mid Y) = \prod P(X_i| \, Y)$ i.e. features are ***conditionally independent*** in order to make learning $P(X \mid Y)$ tractable.
  - Learning model amounts to statistical estimation of $P(X_i| \, Y)'s$ and $P(Y)$
- **Many Variants Depending on Choice of Distributions:**
  - Pick a distribution for each $P(X_i \mid Y=y)$ (Categorical, Normal, etc.)
  - Categorical distribution on $P(Y)$
- **Problems with Naïve Bayes Classifiers**
  - Learning can leave 0 probability entries – solution is to add priors!
  - Be careful of numerical underflow – try using log space in practice!
  - Correlated features that violate assumption push outputs to extremes
- **A notable usage: Bag of Words model**
- **Gaussian Naïve Bayes** with class-independent variances representationally equivalent to Logistic Regression - Solution differs because of objective function

Know Your Models: Logistic Regression Classifier

- **Discriminative Model $P(Y\ |X)$:**
  - Assume $P(Y\ |X = x) = \frac{1}{1+e^{-w^T x}}$ ← sigmoid/logistic fnction
  - Learns a linear decision boundary (i.e. hyperplane in higher d)

- **Other Variants:**
  - Can put priors on weights w just like in ridge regression

- **Problems with Logistic Regression**
  - No closed form solution. Training requires optimization, but likelihood is concave so there is a single maximum.
  - Can only do linear fits…. Oh wait! Can use same trick as generalized linear regression and do linear fits on non-linear data transforms!

Know: Difference between MLE and MAP

- Both are estimate of distribution parameters based on data but MAP includes a prior specified by the model without respect to the data

$$\theta_{MLE} = argmax \overbrace{P(D|\theta)}^{Likelihood}$$

$$\theta_{MAP} = argmax \underbrace{P(\theta|D)}_{Posterior} = argmax \overbrace{P(D|\theta)}^{Likelihood} \underbrace{P(\theta)}_{Prior}$$

- If $P(\theta)$ is uniform, $\theta_{MLE} = \theta_{MAP}$

Be Familiar: Distribution We Discussed

If random variable X is distributed as ___________________.

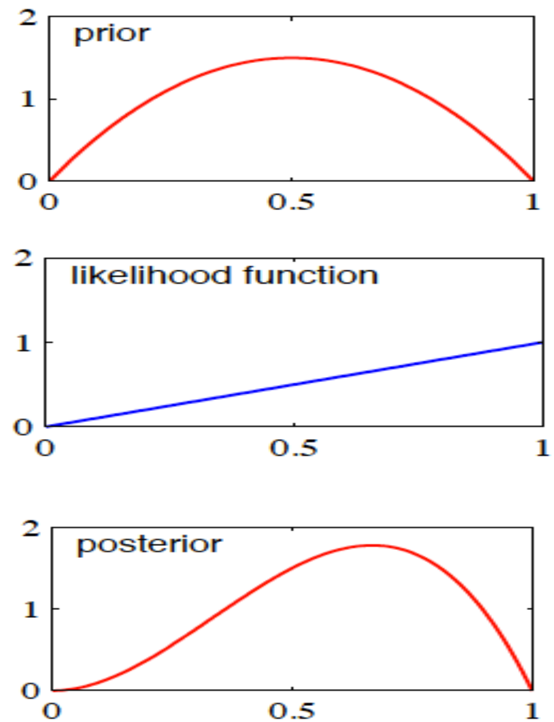
- **Bernoulli**(θ) then X is binary and P(X=1) = θ , P(X=0) = 1 - θ
- **Beta**(α_1, α_o) then X between 0 and 1 and $P(X = x) = \frac{x^{\alpha_1 - 1}\,(1-x)^{\alpha_0 - 1}}{B(\alpha_1, \alpha_0)}$
- **Categorical**($p_1, \ldots, p_k$) then X is discrete {1,…,k} and P(X=k) = p_k
- **Dirichlet**($\alpha_1, \ldots, \alpha_k$) then $X \in \mathbb{R}^k, \sum x_i = 1$, and $P(X = x) = B(\alpha) \prod_{i=1}^{k} x_i^{\alpha_i - 1}$
- **Gaussian**(μ, σ^2) then X is continuous and $P(X = x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$
- **Laplacian**(μ, b) then X is continuous and $P(X = x) = \frac{1}{2b} e^{-\frac{|x-\mu|}{2b}}$

Know: Conjugate Priors / Effect of Priors

Likelihood	Prior	Posterior
Bernoulli	Beta	Beta
Categorical	Dirichlet	Dirichlet
Gaussian	Gaussian	Gaussian

Example: Bernoulli with a Beta Prior

- Prior = Beta(2,2)
  - $\theta_{prior} = 0.5$
- Dataset = {H}
  - $L(\theta) = \theta$, $\theta_{MLE} = 1$
- Posterior = Beta(3,2)
  - $\theta_{MAP} = (3-1)/(3+2-2) = 2/3$

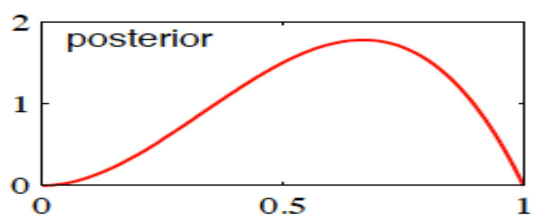


Know: Bayesian Inference (aka appreciating posteriors)

Example: I want to estimate the chance I'll lose money on a bet.

- MLE strategy: find MLE estimate for chance of success under a Bernoulli likelihood and look at expected loss on my gambling.
  - This is a point estimate and requires that my MLE estimate is pretty good

- Bayesian strategy: find posterior over the chance of success and compute expected loss over my beliefs of this chance

$\int$ 

$* \, Cost \; d\theta$

- Lets us reason about the uncertainty of our estimate though the integral of the posterior might be mess… conjugate priors ensure it isn't!

Skills: Be able to Compute MLE of Parameters

- Given i.i.d samples D ={ x_1, …, x_n} from P(X; θ)

1. Write likelihood of D under P(X; θ) as a function of θ
   - Likelihood L(θ) = P(D | θ) = $\prod_{i=1}^{n} P(x_i|\theta)$
2. Take log to get LL(θ) = $\sum_{i-1}^{n} \log(P(x_i \,|\theta))$
3. Solve for argmax $\mathrm{LL}(\theta)$
   - First order methods sometimes give closed form solutions

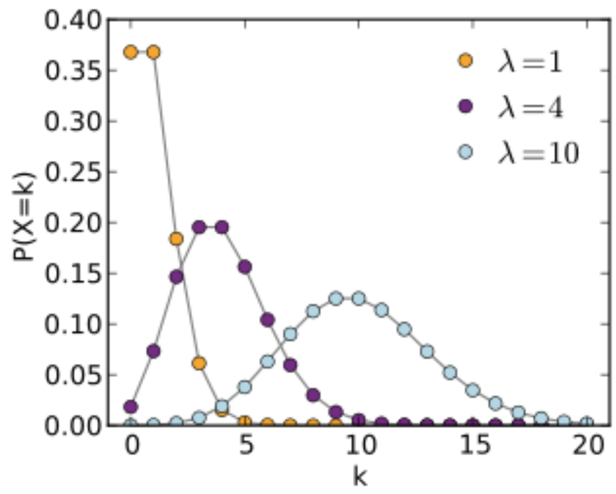
Practice: Compute MLE for Poisson Distribution

- Given i.i.d samples D ={x_1, …, x_n} from P(X; λ) = $\frac{\lambda^x e^{-\lambda}}{x!}$

1. Write likelihood of D under P(X; λ) as a function of λ
   - L(λ) = P(D | λ) = $\prod_{i=1}^{n} \frac{\lambda^{x_i} e^{-\lambda}}{x_i!} = \frac{\lambda^{\sum x_i} e^{-n\lambda}}{x_1! * \cdots * x_n!}$

2. Take log to get LL(λ) = $-\mathrm{n}\lambda + \log(\lambda) \sum_{i-1}^{n}(\mathrm{x_i} \; - \log(x_i!))$

3. Solve for argmax $\mathrm{LL}(\lambda)$
   - $\frac{\delta LL(\lambda)}{\delta\lambda} = -n + \frac{\sum x_i}{\lambda} = 0$
   - $\lambda_{MLE} = \frac{1}{n}\sum x_i$



Skills: Be able to Compute MAP of Parameters

- Given i.i.d samples D ={ x_1, …, x_n} from P(X; θ) with prior P(θ)

1. Write posterior of θ under P(X; θ) as a function of θ
   - P(θ) ∝ P(D | θ)P(theta) = $\prod_{i=1}^{n} P(x_i | \theta) P(\theta)$

2. Take log to get LP(θ) = $\sum_{i-1}^{n} \log(P(x_i \mid \theta)) + \log(P(\theta))$

3. Solve for argmax $\mathrm{LP}(\theta)$
   - First order methods sometimes give closed form solutions

Practice: Compute Map for Poisson Distribution with Gamma Prior

- Given i.i.d samples D ={x_1, …, x_n} from P(X; λ) = $\frac{\lambda^x e^{-\lambda}}{x!}$ and $\lambda \sim Gamma(\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} e^{-\beta\lambda}$

1. Write posterior under P(X; λ) and P(λ) as a function of λ

   - P(λ|D) ∝ P(D | λ) P(λ) ∝ $\underbrace{\prod_{i=1}^{n} \frac{\lambda^{x_i} e^{-\lambda}}{x_i!}}_{P(D|\lambda)} \lambda^{\alpha-1} e^{-\beta\lambda} \propto \boldsymbol{\lambda^{\alpha-1+\sum x_i} e^{-(n+\beta)\lambda}}$

2. LP(λ) ∝ $-(\mathrm{n} + \beta)\lambda + \log(\lambda)\ (\alpha - 1 + \sum_{i-1}^{n} \mathrm{x_i}\ )$

3. Solve for argmax $\mathrm{LL}(\lambda)$

   - $\frac{\delta LL(\lambda)}{\delta\lambda} = -(n + \beta) + \frac{a-1+\sum x_i}{\lambda} = 0$
   - $\lambda_{MAP} = \frac{1}{n+\beta}(\alpha - 1\ +\ \sum x_i)$

Practice: What distribution is the posterior and what are the parameters in terms of X,α, β?

- Given i.i.d samples D ={x_1, …, x_n} from P(X; λ) = $\frac{\lambda^x e^{-\lambda}}{x!}$ and $\lambda \sim Gamma(\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} e^{-\beta\lambda}$

1. P(λ|D) ∝ P(D | λ) P(λ) ∝ $\underbrace{\prod_{i=1}^{n} \frac{\lambda^{x_i} e^{-\lambda}}{x_i!}}_{P(D|\lambda)} \lambda^{\alpha-1} e^{-\beta\lambda} \propto \lambda^{\alpha-1+\sum x_i} e^{-(n+\beta)\lambda}$

$$Gamma(\sum x_i + \alpha, n + \beta)$$

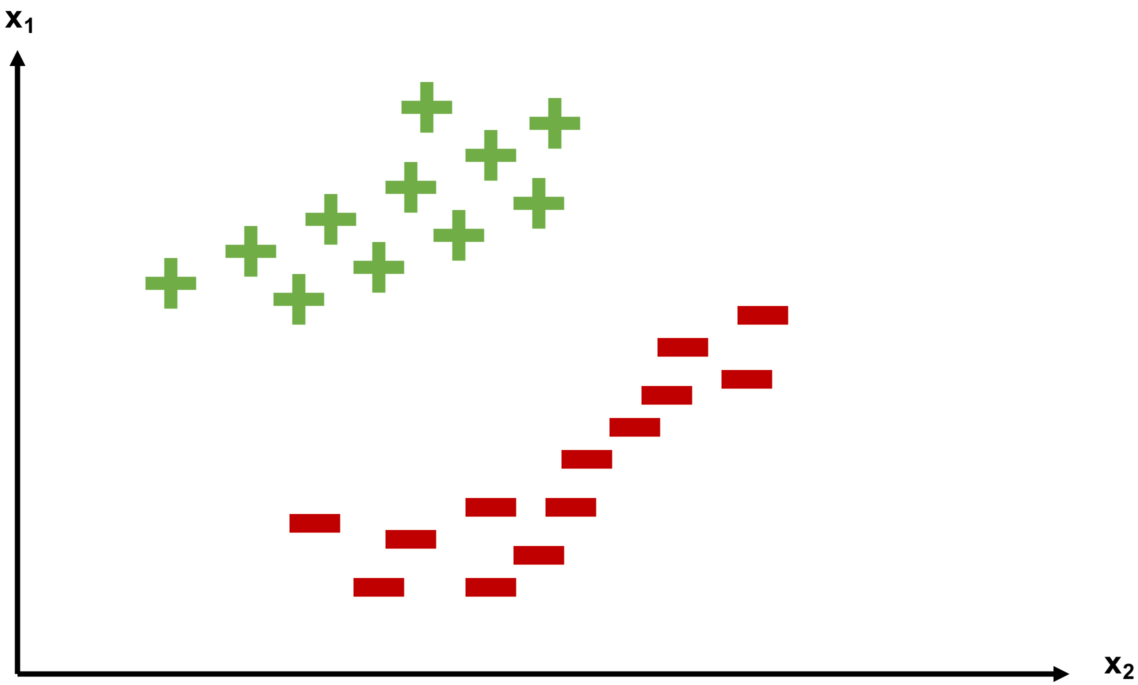
Skills: Be Able to Compare and Contrast Classifiers

- **K Nearest Neighbors**
  - Assumption: f(x) is locally constant
  - Training: N/A
  - Testing: Majority (or weighted) vote of k nearest neighbors

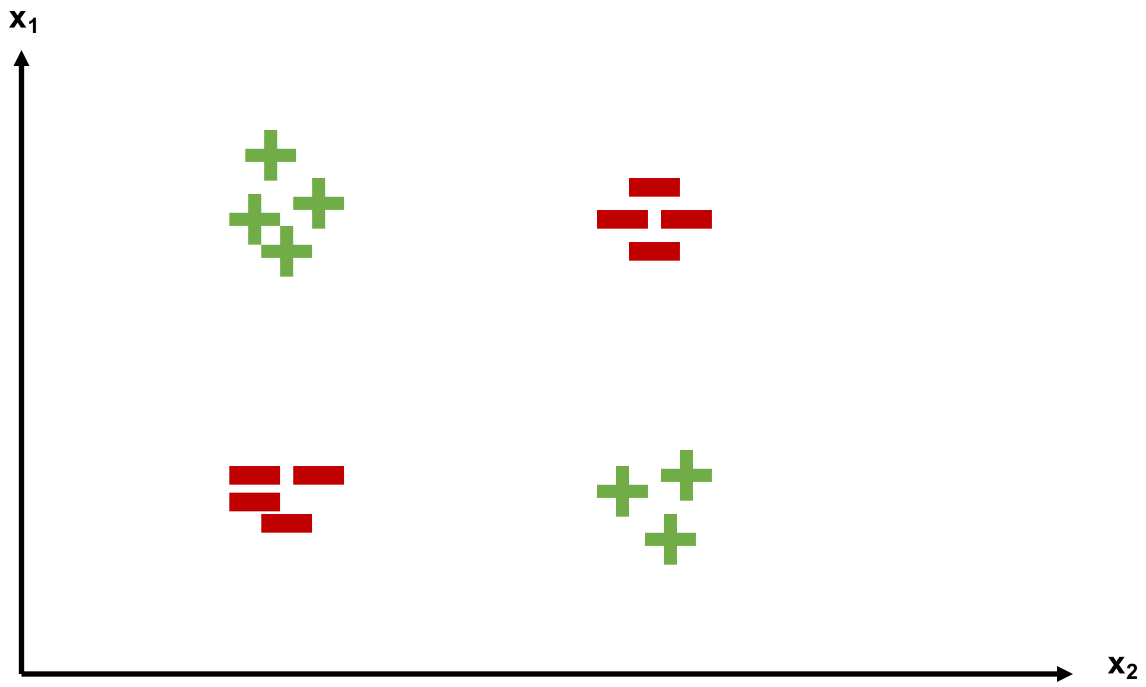
- **Logistic Regression**
  - Assumption: $P(Y|X=x_i) = \text{sigmoid}(w^T x_i)$
  - Training: SGD based
  - Test: Plug x into learned P(Y | X) and take argmax over Y

- **Naïve Bayes**
  - Assumption: $P(X_1,..,X_j \mid Y) = P(X_1 \mid Y)*\ldots* P(X_j \mid Y)$
  - Training: Statistical Estimation of P(X | Y) and P(Y)
  - Test: Plug x into P(X | Y) and find argmax P(X | Y)P(Y)

Practice: What classifier(s) for this data? Why?



Practice: What classifier for this data? Why?



Know: Error Decomposition

- Approximation/Modeling Error
  - You approximated reality with model
- Estimation Error
  - You learned a model with finite data
- Optimization Error
  - You were lazy and couldn’t/didn’t optimize to completion
- Bayes Error
  - there is a lower bound on error for all models, usually non-zero

Know: How Error Types Change w.r.t Other Things

	Modelling	Estimation	Optimization	Bayes
More Training Data		↓		Reality Sucks
Larger Model Class	↓	↑	(maybe) ↑	Reality Still Sucks

How to change model class?

- Same model with more/fewer features
- Different model with more/fewer parameters
- Different model with different assumptions (linear? Non-linear?

How much data do I need?

- Depends on the model.. Gaussian Naïve Bayes and Logistic regression give same result in the limit if GNB assumptions hold
- GNB typically needs less data to approach this limit but if the assumptions don't hold LR is expected to do better.

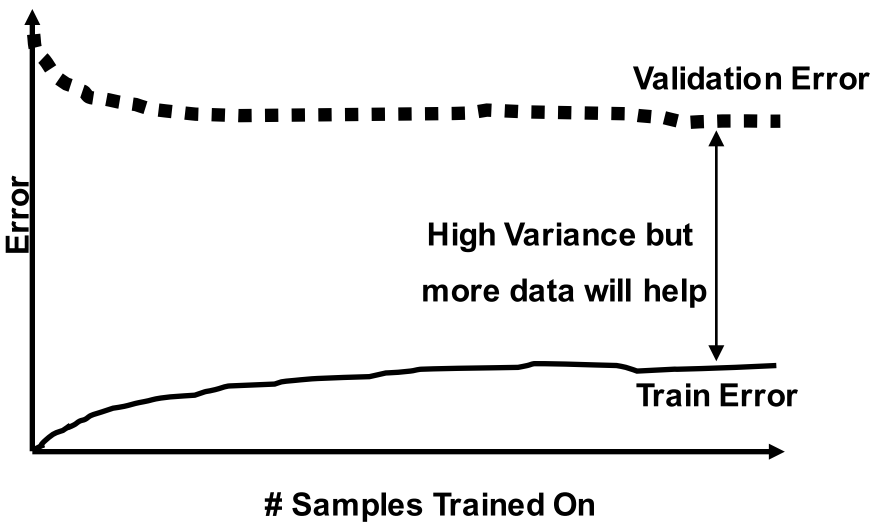
Know: Bias vs Variance

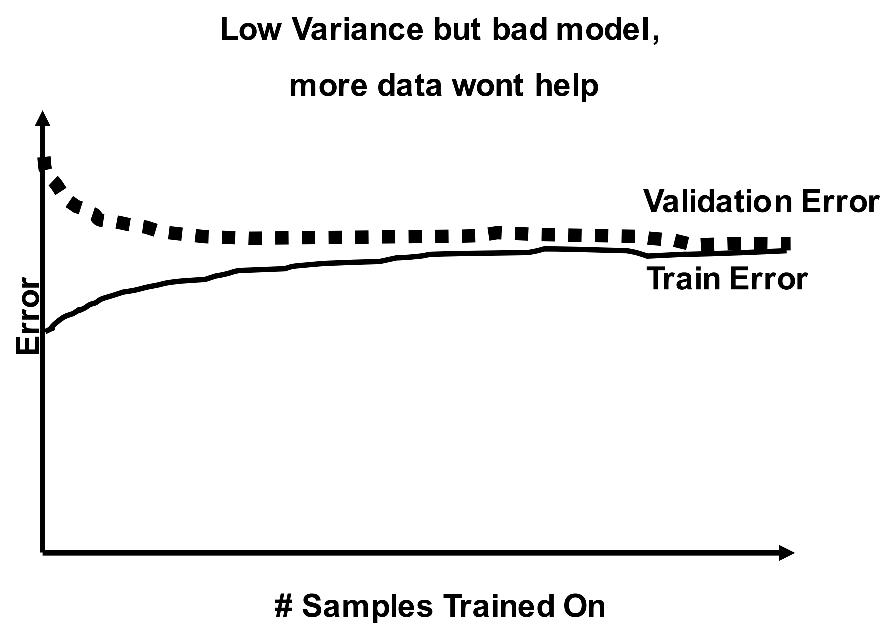
- **Bias:** difference between what you expect to learn and truth i.e. $E[\theta] - \theta^*$
  - Measures how well you expect to represent true solution
  - Decreases with more complex model

- **Variance:** difference between what you expect to learn and what you learn from a from a particular dataset i.e $E[(\theta - E[\theta])^2]$
  - Measures how sensitive learner is to specific dataset
  - Increases with more complex model

Know: Learning Curves

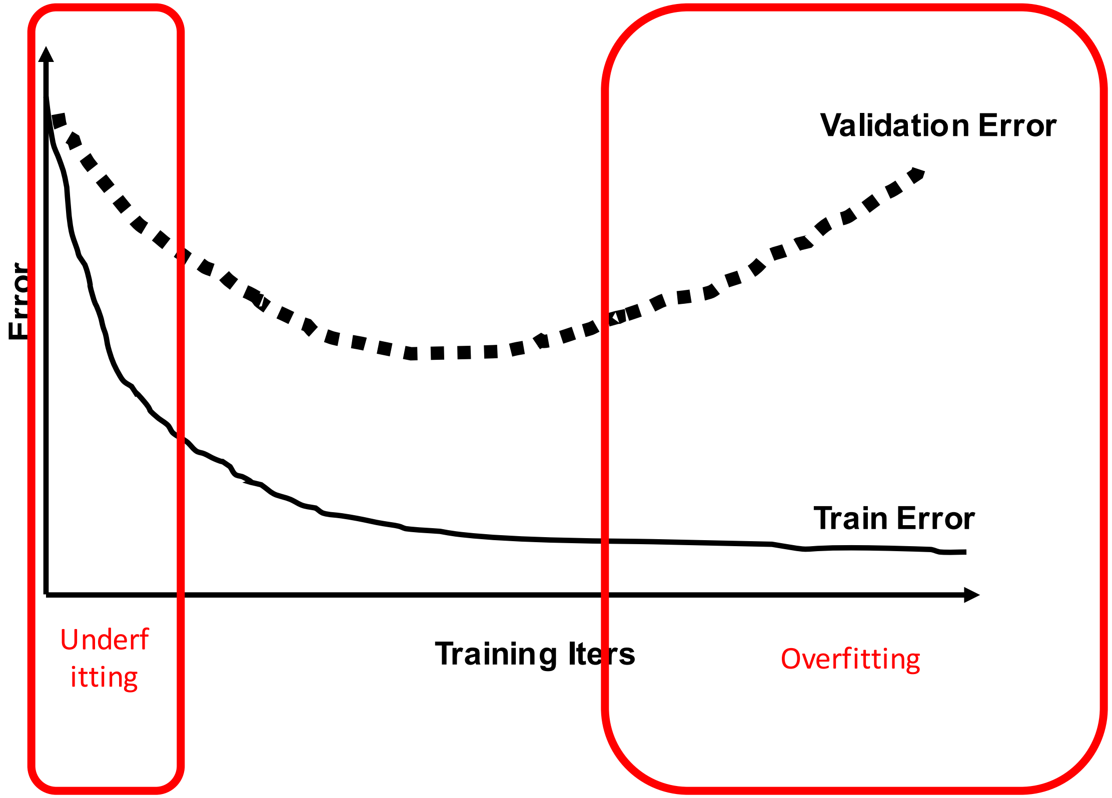
- **Plot** error as a function of training dataset size





Know: Underfitting & Overfitting

- **Plot** error through training (for models without closed form solutions



- Overfitting is easier with more complex models but is possible for any model
- More data helps avoid overfitting as do regularizers

Know: Train/Val/Test and Cross Validation

Train – used to learn model parameters

Validation – used to tune hyper-parameters of model

Test – used to estimate expected error

- The improved holdout method: k-fold *cross-validation*
    - Partition data into k roughly equal parts;
    - Train on all but j-th part, test on j-th part

- An extreme case: *leave-one-out* cross-validation

$$\hat{L}_{\text{cv}} = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - f(\mathbf{x}_i; \hat{\mathbf{w}}_{-i})\right)^2$$

where $\hat{\mathbf{w}}_{-i}$ is fit to all the data but the i-th example.

Skills: Be Able to Argue for Concavity/Convexity

- Today's readings help a great deal!
- $f : \Re^d \to \Re$ is a convex function if domain of f is a convex set and for all $\lambda \in [0, 1]$

$$f(\lambda w_1 + (1 - \lambda) w_2) \leq \lambda f(w_1) + (1 - \lambda) f(w_2)$$

- **Alternative:** show the Hessian matrix is positive semidefinite
- **Alternative:** argue with properties of convexity i.e. affine functions are convex, min of convex functions are convex , sum of convex functions is convex, etc..

Practice: Show if f(x) is convex

- $f(x) = x^2$
  - $H = \left[\frac{\delta f}{dx^2}\right] = 2.$ $\quad a * 2 * a = 2a^2 \geq 0 \; \forall a$, therefore convex

- $f(x, y) = x^2 - \log(y)$
  - $H = \begin{bmatrix} \frac{\delta f}{\delta x^2} & \frac{\delta f}{\delta y \, \delta x} \\ \frac{\delta f}{\delta x \, \delta y} & \frac{\delta f}{\delta y^2} \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & \frac{1}{y^2} \end{bmatrix}$, $a^T H a = 2a_1^2 + \frac{a_2^2}{y^2} \geq 0 \; \forall a, y, \therefore convex!$

- $f(x, y) = \log(x/y)$

- $H = \begin{bmatrix} \frac{\delta f}{\delta x^2} & \frac{\delta f}{\delta y \, \delta x} \\ \frac{\delta f}{\delta x \, \delta y} & \frac{\delta f}{\delta y^2} \end{bmatrix} = \begin{bmatrix} -\frac{1}{x^2} & 0 \\ 0 & \frac{1}{y^2} \end{bmatrix}$, $\mathrm{a^T H a} = -\frac{\mathrm{a}_1^2}{\mathrm{x}^2} + \frac{\mathrm{a}_2^2}{\mathrm{y}^2} < 0 \; if \; a_1 > a_2$
  - Non-convex!