**CMPS242 – Machine Learning**

By: Zayd Hammoudeh

[Lecture 1 Introduction to Machine Learning 2](#_Toc495263200)

[Lecture 2 Probability Theory 3](#_Toc495263201)

[Lecture 3 Coding Theory, Relative Entropy, Convexity, and Jensen’s Inequality 5](#_Toc495263202)

# Introduction to Machine Learning

|  |  |  |  |
| --- | --- | --- | --- |
| **Univariate Polynomial Curve Fitting**   * - Learned weight vector * - Input Value * - Target Value * - Predicted value | **Overfitting** – The model is so “good” it just learns the training data.  **Techniques to Prevent Overfitting**   * Use more training data * Regularization * Select a simpler model. | **Sum of Squares/Total Error** | **Regularization**   * Penalizes large coefficient values. * Training Error always minimum when there is no regularization (i.e., ) |
| **Mean-Squared Error** |
| **Root-Mean-Square Error** |
| **Regularized Error** |

**Training Paradigms**

|  |  |  |
| --- | --- | --- |
| **Single Split Method**   * Randomly split the training set 70% training, 30% validation * Perform training on the training subset * Select the model based on their performance on the validation set. * **Advantages**:   + Simplest to implement (**easy**).   + Least computationally expensive (**cheap**). * **Disadvantages**:   + **Wastes data** as selecting best method using 30% less data.   + Highly affected by validation split profile (**unreliable**) | **-Fold Cross Validation**   * Randomly partition the training set into disjoint subsets of roughly equal size. * Perform training on training subsets and validation on the left out split. * Repeat the process holding out a different split. * Use the mean error to select the best model. * Middle approach * **Advantages**:   + Train on all the data   + Less variance than single split. * **Disadvantages:**   + **Wastes of the data** (better than single split if   + Moderate computational approach | **Leave-One-Out Cross Validation**   * Select a single record as the validation set. * Train on all other data points. * Repeat the process holding out a different single record until all have been held out. * Use the mean error to select the best model. * **Advantages**:   + Trains and validates on all data (**no data waste**) * **Disadvantages:**   + Most computationally **expensive**.   + High variance (**weird behavior**) |

|  |  |  |
| --- | --- | --- |
| **CV-Based Model and Hyperparameter Selection**   * Compute Cross Validation error for all model/hyperparameter combinations. * Select the model/hyperparameter that returned the best CV error. * Train the model again using all the training data. | **Key Takeaway**: “Not enough to memorize, but need to be able to generalize.” |  |

# Probability Theory

|  |  |  |
| --- | --- | --- |
| * **Sample Space** () – Set all of possible outcomes. * **Event** – Set of outcomes, i.e., any subset of * Probability Distribution – Function * **Example:** Given event , its probability distribution is: | **Axioms of Probability**   * Axiom #1: * Axiom #2: * **Axiom #3:** For an infinite sequence of disjoint events, , | **Properties of Union**   * Disjoint Union: * Non-disjoint Union: |
| **Conditional Probability**   * Amount of “A” in “B” * Property of Intersection: * Rewriting Conditional Probability: |

|  |  |  |
| --- | --- | --- |
| **Independence**   * Two events are independent if and only if: * Alternate Definition of Independence: * Only holds if * Definition implies: | **Partition**   * **Partition:** A set of **mutually exclusive** events whose union equals the sample space. * **Combining Conditional Probability and Partitioning:**    + Define the sequence as a partition   + Essentially a type of weighted average. | **Conditional Independence**   * Given a set of **independent** events: |
| **Multivariate Probability Distributions**   * **Joint Probability**: Given **at least two random variables**, it is a single composite probability distribution in the form: * **Marginal Probability**: Univariate probability distribution derived from a joint distribution: * **Conditional Probability**: Distribution conditioned on another variables value: |
| **Additional Properties of Conditional Probability**   * Each event is independent with the **sample space** as: * Independent with the **Empty Set** |
| **Bayes’ Rule**   * Relies on being part of a **partition**. * Prior Probability: and * Likelihood: and * Posterior Probability: |

|  |  |  |
| --- | --- | --- |
| **Rules of Probability**  **for Discrete Variables**   * **Sum Rule:** * **Product Rule:** * **Bayes Theorem:** | **Probability Densities**   * – Probability Density Function   + **Continuous:**   + **Discrete:** * – **Cumulative Distribution Function**   + Monotonically increasing   + **Bounded:**   + Continuous Definition: | **Expectations**   * Definition:   + **Discrete:**   + **Continuous:** * **Conditional Expectation:** * **Approximate Expectation**: Applies for both discrete and continuous random variables |

|  |  |  |
| --- | --- | --- |
| **Variances and Covariances**   * **Variance:** * **Covariance:** Between two random variables and * **Covariance** (Vector): | **Gaussian Distributions** | |
| **Univariate Gaussian** |  |
| **Multivariate Gaussian**   * - Dimension of and | * - **Covariance matrix** |

|  |  |  |
| --- | --- | --- |
| **Gaussian Parameter Estimation**   * Normal distribution is a likelihood function since it is given parameters (, ) * For i.i.d. samples: | **Maximum (Log) Likelihood**   * Find the parameters that maximize the likelihood * Easier if maximizing the log rather than the original function * Easier to determine with the | **MAP** – Maximizing Posterior Probability |

# Coding Theory, Relative Entropy, Convexity, and Jensen’s Inequality

**Coding Theory**

|  |  |
| --- | --- |
| Given a random variable, :   * Measure of Surprise: * : Very surprised * : Not surprised * **Entropy:** Weighted average of the measure of surprise. Formally:   + Entropy is the “expectation of the surprise”   + Entropy is largest when is uniform for all possible | **Huffman Code**   * **Prefix Code**: No code is a prefix of another code.   **Basic Pseudocode**  Initialize Each Outcome as a Subtree  **While** |Subtrees| > 1  Pick the two subtrees with smallest root values  Combine smallest subtrees with root node that is sum of previous roots  Start at root of the single tree and assign “0” and “!” at each branch |

|  |  |  |  |
| --- | --- | --- | --- |
| * **Code**: Assigns **symbols** a bitstring (i.e., **codeword**). It must be uniquely decodable. * **Unique Decodability** – Symbols are assigned in such a way to ensure there is only single possible way to decode any string. | **Expected Code Length**   * – Code * – Length of codeword for symbol (in bits) | **Theorem #1:** Given is the **optimal encoding**, then the Huffman code’s length is:  **Theorem #2:** Huffman codes are optimal. |  |

**Relative Entropy/Kullback-Leibler Divergence**

|  |  |  |
| --- | --- | --- |
| **Relative Entropy/**  **Kullback-Leibler Divergence**  **Properties:**   * Non-negative * **Asymmetrical** (i.e., ) * Only equals 0 if | **Relative Entropy/Kullback-Leibler Divergence**  **Note**: is the expected code length for distribution | **L’Hopital’s Rule**  **If :**  **Then**   * Uses derivatives to help determine limits in indeterminate form. * Rule may be applied multiple times to the same expression.   **Example:** |

**Convexity and Jensen’s Inequality**

|  |  |  |  |
| --- | --- | --- | --- |
| **Convex Function**: A function is said to be *convex* *over an interval*  if for every and it holds:   * **Strictly Convex:** Only inequality holds (i.e., never equality) * Sum of two convex functions is also convex. | **Concave:** A function, , is *concave* if is convex. | **Properties of Convex Functions**   * Sum of two convex functions is also convex. * Maybe be highly susceptible to outliers. * Local minimum is a global minimum. * Tangent line is always below the curve. * Linear and constant functions are both concave and convex. | **Sufficient Conditions**   * **Convex**: Second derivative always positive. * **Concave**: Second derivative is always negative. * Note a necessary condition. |

|  |  |  |
| --- | --- | --- |
| **Theorem:** If the function has a second derivative which is non-negative (*positive*) everywhere, then the function is convex (*strictly convex*).   * **Counterexample for the Necessity of Second Derivative:** is at | **Expectation**   * Denoted as * Discrete form: * Continuous Form: | **Jensen’s Inequality:** If is a convex function and is a random variable, then:  Moreover, if is strictly convex, then equality for implies that with probability 1 (i.e., is a constant).  If is concave, then the inequality sign is reversed. |

|  |  |  |
| --- | --- | --- |
| **Jensen’s Inequality for Discrete Random Variables** | **Jensen’s Inequality for Continuous Random Variables** | **Mutual Information**   * Quantifies the mutual dependence between two variables.   + Quantifies how similar the joint distribution is to the product of the marginal distributions. * – Non-negativity with equality only when . |

|  |  |  |  |
| --- | --- | --- | --- |
| **Relationships Between Entropies** | **Quasi-Convex** | |  |
| **Conditional Entropy** | * Addresses the high (i.e., unbounded) susceptibility of convex functions to outliers. * Inverse of function is convex in the range |  |

# Basic Distributions and Exponential Families

**Binary Random Variables**

|  |  |  |  |
| --- | --- | --- | --- |
| **Binary Random Variables** | | **Parameter Estimation:**  **Maximum Likelihood for BernoullI**   * – Observations | **Overfitting:** As with other techniques, parameter estimation is vulnerable to overfitting.   * **Example:** Flip a coin and get three heads. |
| **Bernoulli Distribution** | **Binomial Distribution** |

**Non-Binary Random Variables**

|  |  |  |  |
| --- | --- | --- | --- |
| **Beta Distribution** | **Multinoulli Distribution** | | Multinomial Distribution |
| * “1 to K” Coding Scheme: | ML Parameter Estimation   * Solvable using a Lagrange Multiplier where the constraint is: |

|  |  |  |
| --- | --- | --- |
| **Multinomial Distribution** | **Single-Variate Gaussian Distribution** | **Multivariate Gaussian Distribution**   * - Dimension of and |

|  |  |  |  |
| --- | --- | --- | --- |
| **Central Limit Theorem** – The distribution of the sum of i.i.d. random variables becomes increasingly Gaussian as grows. |  |  |  |

**Exponential Family**

|  |  |  |  |
| --- | --- | --- | --- |
| **Form of Exponential Family**   * - Maybe a vector or scalar * – Distribution parameter(s) | **Bernoulli Distribution in the Exponential Family** | **Logistic Sigmoid Function** |  |

**Non-Parametric Models**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nonparametric Models**   * Requires storing and computing with the entire dataset. * Less efficient than parametric models for storage and computation (once parametric model is set). | **-Nearest-Neighbors**   * - Acts as a smoother * For , the error rate of 1-nearest neighbors classifier is never more than twice the optimal. | **Domain Partitioning**   * Partition the input ranges of each parameter into “**bins**” * Define classification based on the most numerous in the grid cell. * **Disadvantage:** Not scalable. Number of locations grows exponentially with feature dimension. |  |

# Bregman Divergence

|  |  |  |  |
| --- | --- | --- | --- |
| **Bregman Divergences**   * Used to construct losses and regularizers * Convex by design * **Definition:** Relative entropy between *two distributions of the same family with different parameters* | **Training of Parameters**   * **Online:** Get one example at a time.   + All data may not be able to fit into memory. * **Batch:** Training based on loss of all examples at once * **Mini-Batch:** Multiple examples at a time, but not all at the same time. |  | **Stochastic Gradient Descent**   * Online update on random example(s)   + Could be a minibatch |