**Statistical Learning: Overview**

* **Definition**: Application of statistical methods to make predictions about unseen data.
* **Applications**:
  + Predicting weather using meteorological data
  + Naval ship classification from satellite images
  + Estimating likelihood of a second stroke post initial occurrence

**Course Goals**

1. **Conceptual Understanding**:
   * Grasp underlying intuition, concepts, and thought process behind statistical/machine learning.
2. **Theory Mastery**:
   * Understand theoretical underpinnings of algorithms/methods.
3. **Practical Skills**:
   * Capability to implement machine learning methods on real datasets.

**Example: Spam Detection**

* **Objective**: Craft a spam filter for emails.
* **Dataset**: 4601 emails.
* **Features**: Relative frequencies of 57 most common words and punctuation marks.

**Basic Terminology**

* **Features**: Variables utilized for predictions, e.g., weight, height, diet.
* **Response/Outcome**: Target variable one aims to predict, e.g., lifespan.

**Learning Problems**

* **Regression**: Aimed at predicting continuous variables, e.g., height or sales.
* **Classification**: Aimed at predicting categorical variables, e.g., color or success/failure.

**Data Types**

* **Training Data**: Data with known outcomes, utilized for model training.
* **Testing Data**: Data with unknown outcomes, utilized for model evaluation.

**Learning Types**

1. **Supervised Learning**:
   * **Definition**: Predicting a specific response using labeled training data.
   * **Example**: Classifying images as "cars" or "fish".
2. **Unsupervised Learning**:
   * **Definition**: Discovering patterns without a predetermined response variable.
   * **Example**: Determining distinct customer groups.

**Terminology: Statistics vs. Machine Learning**

| **Statistics** | **Machine Learning** |
| --- | --- |
| Regressor/x-variable | Feature |
| Weights | Parameters |
| Fitting a model | Learning |
| Response variable | Outcome |
| Prediction model | Learner |

**Week 2: Linear Models**

**1. Introduction**

* Goal: Predict response *Y* using features *X*1​,*X*2​,…,*Xp*​.
* Assumption: Linear relationship between *Y* and *X*.
* Quote: “Essentially, all models are wrong, but some are useful” - George Box

**2. Simple Linear Regression**

* Model: *Y*=*β*0​+*β*1​*X*+*ϵ*
* Predict for new observation: *Y*^new​=*β*^​0​+*β*^​1​*x*new​

**3. Multiple Linear Regression**

* Model: *Y*=*β*0​+∑*j*=1*p*​*βj*​*Xj*​+*ϵ*
* Predict for new observation: *Y*^=*β*^​0​+∑*j*=1*p*​*β*^​*j*​*Xj*,new​

**4. Flexibility**

* Linear models can include various transformations (quadratic, cubic, logarithm) and interactions.
* Suitable for linear or near-linear relationships.

**5. Interpretability**

* Linear models provide insight into relationships between variables.
* Useful for testing causal relationships.

**6. Example on Interpretability**

* Modeling product sales as a function of advertising.
* Multiple variable consideration (e.g., TV advertising) to avoid false correlations.

**7. Matrix Form**

* *Y*=*Xβ*+*ϵ*
* Predictions: *Y*^=*Xβ*^​

**8. Loss Functions**

* Aims to minimize deviations between predictions and observed responses.
* Examples:
  + Squared Error Loss
  + Absolute Error Loss
  + Huber Loss

**9. Loss Function Comparisons**

* Ordinary Least Squares (OLS) uses squared error loss.
* Robust regression uses absolute error loss.
* Huber Loss combines the two.

**10. Least Squares Estimation**

* Single Variable: *β*^​0​=*y*ˉ​−*x*ˉ*β*^​1​ and*β*^​1​=Var(*x*)Cov(*x*,*y*)​
* Multiple Variables: *β*^​=(*XTX*)−1*XTy*

**11. Statistical Properties**

* OLS is unbiased under certain assumptions.
* Variance calculations provided.

**12. Mean Squared Error (MSE)**

* Defined as the expected value of the squared deviation of an estimator from the actual value.
* OLS provides the lowest variance among unbiased linear estimators.

**13. Residuals**

* Difference between observed and predicted values.
* OLS minimizes the sum of squared residuals.

**14. Fitted Values**

* Predicted values from the regression model.

**15. Prediction Error**

* Unreliable predictions outside the range of training data.
* OLS error increases with distance from the mean.

**16. Classification**

* Using linear regression for classification results in a linear decision boundary.

**Week 3 Study Notes**

**Introduction**

* The lecture covers three main topics: KNN, Bias-Variance Tradeoff, and Cross Validation.
* The Bias-Variance Tradeoff and Cross Validation are foundational concepts that will be utilized throughout the course.
* KNN is introduced as a crucial algorithm to elucidate these concepts.

**K Nearest Neighbors (KNN)**

* A simple estimator predicting values based on their proximity to other data points.
* Formula for prediction:

*Y*^(*x*new​)=*k*1​*j*∈*Nk*​(*x*new​)∑​*yj*​

where *Nk*​(*x*new​) is the "neighborhood" of *x*, defined by the *k* closest points.

* Closeness is often measured using the Euclidean distance.

**KNN Regression Example:**

* Predict an individual's weight based on their height using *k*=2.
* For a height of 69, the two closest data points (by height) are 70 and 67, with weights 180 and 190, respectively. The predicted weight is the average, 185.

**KNN Classification:**

* For classification problems, if *Y*^>0, predict Group 1; if *Y*^<0, predict Group 2.
* The group with the majority of "close" points becomes the prediction.

**KNN Neighborhoods:**

* Different values of *k* lead to varied decision boundaries and prediction regions.

**Practical Considerations and Uses:**

* KNN is widely applied despite its simplicity.
* Useful when decision boundaries are irregular or when classes are in complex non-linear groupings.
* Standardizing features ensures comparable distances when constructing neighborhoods.
* KNN can be computationally intensive for large datasets because it requires the entire training dataset for predictions.
* In tie scenarios, decisions are often random.

**KNN Prediction Errors:**

* Training Error: Misclassification rate on the training data. It's zero when *k*=1.
* Test/Prediction Error: Misclassification rate on test data. Model performance is typically gauged by the misclassification percentage on the test set.

**Bias-Variance in KNN**:

* **Low *k* (e.g., *k*=1)**:
  + Many distinct regions for predictions, indicating a complex/overfit model.
  + Model fits noise in the data.
  + Characteristics: Low Bias, High Variance
* **High *k* (e.g., *k*=*N*)**:
  + Few distinct regions for predictions, indicating a simple/underfit model.
  + The model misses trends by averaging over many observations.
  + Characteristics: High Bias, Low Variance

**Bias-Variance Tradeoff**:

* **High Bias/Low Variance Models**:
  + Simple models missing some data trends due to many assumptions.
  + Insensitive to small training data changes.
  + Example: Using the response mean as a prediction results in "Under-fitting".
* **Low Bias/High Variance Models**:
  + Complex models with weak assumptions closely fitting the data.
  + Sensitive to small training data changes.
  + Example: Using the closest training data observation for prediction (KNN with *k*=1) results in "Over-fitting".

**Cross Validation Introduction**:

* Challenge: Selecting parameters like *k* in KNN and assessing model performance on test data.
* Expected Prediction Error (EPE) is ideal but isn't directly observable. It's estimated using test data.
* Two main performance assessment approaches:
  1. **Holdout Method**
  2. **Cross Validation**

**Holdout Method**:

* Split data into training and testing sets (commonly 75-25).
* Estimate EPE on the test set.
* If the performance of the final model needs assessment, take a second holdout sample (e.g., 50-25-25 split).

**Cross Validation**:

* Randomly split data into *k* groups.
* Train the model using all data except one group.
* Use that group as the test data and save the average error.
* Repeat for all groups.
* Compute the average of the *k* error estimates.

**Week 4 Study Notes**

**Introduction**

* The session delves into shrinkage and selection methods in linear models.
* When estimating high-degree polynomials with OLS, the model might fit the data too closely, resulting in high variance. This means the model fits the noise in the data and not the overall trend. For instance, even if the overall trend is only quadratic, a higher-degree polynomial might overfit.

**Possible Remedies:**

1. Make the function estimate smoother (Example: Ridge Regression).
2. Drop some polynomial terms to prevent the model from fitting the data too closely (Example: Best Subsets Regression).
3. Combine both of the above approaches (Example: Lasso Regression).

**Ridge Regression Introduction**

* To make the function smoother, we need to manage the coefficients' magnitudes. Larger coefficients indicate that the function *f*(*x*) changes rapidly with *x*.
* Ridge Regression aims to reduce the magnitude of the coefficients to achieve a smoother function.

**Ridge Regression: Lagrangian Formulation**

* Ridge regression estimates aim to minimize:

*β*^​Ridge​=arg*β*min​⎣⎡​*i*=1∑*N*​(*yi*​−*β*0​−*j*=1∑*p*​*βj*​*xij*​)2+*λj*=1∑*p*​*βj*2​⎦⎤​

* *λ* is a parameter that determines the amount of "shrinkage" or reduction in the coefficients.
  + Larger *λ* leads to smaller coefficients.
  + Smaller *λ* leads to larger coefficients.
  + As *λ*→∞, all coefficients approach 0.
  + As *λ*→0, coefficients approach the least squares estimates.

**Ridge Regression: Constrained Formulation**

* Ridge regression can also be formulated as:

*β*^​Ridge​=arg*β*min​⎣⎡​*i*=1∑*N*​(*yi*​−*β*0​−*j*=1∑*p*​*βj*​*xij*​)2⎦⎤​

subject to:

*j*∑​*βj*2​≤*C*

Where *C* is a constraint or limit on the size of the coefficients.

* There's a one-to-one correspondence between *C* and *λ*.

**Lasso Introduction**

* Lasso regression involves penalizing the coefficients based on their absolute values.
* **Drawbacks**:
  + The objective function is no longer smooth and differentiable concerning *β*.
* **Benefits**:
  + It forces some parameters to be shrunk and others to be dropped entirely. This feature can be more effective in certain scenarios.

**Lasso: Lagrangian Formulation**

* Lasso regression is similar to ridge regression but uses an absolute value penalty instead of a squared penalty:

*β*^​Lasso​=arg*β*min​⎣⎡​*i*=1∑*N*​(*yi*​−*β*0​−*j*=1∑*p*​*βj*​*xij*​)2+*λj*=1∑*p*​∣*βj*​∣⎦⎤​

* The intercept is not included in the penalty.
* A higher *λ* leads to smaller *β* values, with more coefficients set to zero.
* A lower *λ* results in larger *β* values, with fewer coefficients set to zero.
* *λ* is typically chosen to minimize expected prediction error, often through cross-validation.

**Lasso: Constrained Formulation**

* Lasso regression can also be formulated as:

*β*^​Lasso​=arg*β*min​⎣⎡​*i*=1∑*N*​(*yi*​−*β*0​−*j*=1∑*p*​*βj*​*xij*​)2⎦⎤​

subject to:

*j*∑​∣*βj*​∣≤*C*

Where *C* is a constraint or limit on the size of the coefficients.

* There's a one-to-one correspondence between *C* and *λ*.

**Lasso Intuition**

* Unlike ridge regression, Lasso will set some parameters equal to zero when *C* or *λ* is sufficiently small.
* Lasso has two primary effects compared to OLS:
  1. Dropping parameters
  2. Shrinking parameters
* Lasso is a combination of subset selection and shrinkage approaches, like ridge regression.

**Best Subsets Selection (BSS)**

* BSS finds the subset of size *k* (where *k* ranges from 1 to *p*) that results in the lowest sum of squared error on the training data.
* Of the remaining *k* models, one model is selected using criteria such as cross-validation and the expected prediction error (like AIC).
* Selecting the *k* that results in the lowest sum of squared errors on training data isn't ideal because this error always declines with the number of variables, possibly leading to overfitting.
* One benefit of selection methods is interpretation. More parsimonious models are more interpretable.

**Drawbacks of BSS**

1. **Computationally Demanding**: Running many regressions for each *k*. For *p*=30, over a billion regressions must be run. It's impractical for *p*>40 even with optimized methods.
2. **High Variance**: The discrete nature of BSS can result in high variance. Small changes in the training data might lead to significant changes in selected variables.

**Best Subsets Selection: Alternatives**

A more constrained search might be computationally easier and yield lower variance. Some alternatives are:

1. **Forward Stepwise Selection**: Start with a model containing only the intercept and sequentially add the variable that reduces the sum of squared error the most. Continue until all variables are included.
2. **Backward Stepwise Selection**: Start with a full model and sequentially drop the variable with the least impact until only the intercept remains.

**Elastic Net**

* Elastic Net is a compromise between Lasso and Ridge:

*β*^​ElasticNet​=arg*β*min​⎣⎡​*i*=1∑*N*​(*yi*​−*β*0​−*j*=1∑*p*​*βj*​*xij*​)2+*λ*(*αj*=1∑*p*​*βj*2​+(1−*α*)*j*=1∑*p*​∣*βj*​∣)⎦⎤​

Where:

* + *α*=0 corresponds to Lasso
  + *α*=1 corresponds to Ridge
* The Elastic Net shrinks coefficients like Ridge and selects variables like the Lasso.

**Week 5 and 6 Study Notes**

**Introduction to Classification**

* Classification is a subset of supervised learning where the goal is to predict the categorical class labels of new instances based on past observations.
* Unlike regression where the output is numerical, classification output is categorical. The primary objective is to identify which category (or class) an unknown data point belongs to.
* Linear classifiers classify observations based on a linear combination of inputs.

**Classification: Goals and Bayes' Classifier**

* The Bayes' Classifier provides a theoretical approach to classification. It assigns an observation to the class for which the posterior probability *P*(*Y*=*k*∣*X*=*x*) is highest.
* The Bayes Error rate, represented as:

*E*(*Y*=*Y*^∣*X*=*x*)

denotes the lowest possible error rate we can achieve, even if we knew the true probabilities. This serves as a gold standard against which to compare other methods.

**Linear Regression for Classification**

* While linear regression is traditionally used for predicting a continuous outcome, it can be employed for binary outcomes by coding one category as "1" and the other as "0".
* The model is given by:

Pr(*Y*=1∣*X*)=*β*0​+*β*1​*X*

Predictions are made by inputting features into the regression equation and rounding the output to the nearest integer: values below 0.5 are rounded to 0, and values above 0.5 are rounded to 1.

* This method's limitation is that it can produce predicted values outside the [0,1] range, which doesn't make sense for probabilities.

**Decision Boundaries in Classification**

* The decision boundary is the hyperplane that separates different classes in the feature space.
* For a two-dimensional feature space, the decision boundary is a line. As the number of features increases, the boundary becomes a hyperplane in a multi-dimensional space.
* The decision boundary is crucial for understanding how a model distinguishes between different classes.

**Logistic Regression**

* Logistic regression estimates the probability that an instance belongs to a particular category.
* Unlike linear regression, which can predict values outside the [0,1] range, logistic regression produces outputs strictly between 0 and 1.
* The model for probability is:

Pr(*Y*=1∣*X*)=1+*eβ*0​+*β*1​*Xeβ*0​+*β*1​*X*​

This formula utilizes the logistic function to squeeze the output of a linear equation between 0 and 1.

**Understanding Odds and Logits in Logistic Regression**

* The odds ratio is a measure of the odds of an event happening compared to it not happening. It's defined as:

Odds=1−Pr(*Y*=1∣*X*)Pr(*Y*=1∣*X*)​=*eβ*0​+*β*1​*X*

* The log-odds or logit transformation is the logarithm of the odds ratio:

Logit=log(1−Pr(*Y*=1∣*X*)Pr(*Y*=1∣*X*)​)=*β*0​+*β*1​*X*

In logistic regression, coefficients represent the change in the log odds for a one-unit change in the predictor. Exponentiating these coefficients gives the change in odds for a one-unit change in the predictor.