CS4414 Cheat-Sheet

Gurpreet Singh
gsingh95@uwo.ca
6. November 2017

Data Preparation

Data Cleaning

Data cleaning is the process of detecting and correcting (or removing) corrupt or inaccurate records from a record set, table, or database and refers to identifying incorrect parts of the data and then deleting the dirty or coarse data.

Statistics

Definitions

`BoxPlot`: Extended lines mean Min and Max. Box is third quartile, median and first quartile top to bottom. Suspected outliers are empty dots, outliers are solid dots

`Histogram`: Same as a barchart but groups numbers into ranges on the x-axis

`Independence`: Two random variables X and Y are independent if Any conditional distribution of X is the same as the marginal distribution of X and knowing about Y provides no information about X.

`Marginal Distribution`: gives the probabilities of various values of the variables in the subset without reference to the values of the other variables. The sum of the columns.

`Conditional Distribution`: shows the probability that a randomly selected item in a sub-population has a characteristic you're interested in. Moving the column (scanning)

`Joint Distribution`: X and Y have a joint distribution if their realizations come together as a pair. (X, Y) is a random vector, and realizations may be written $(x_1, y_1), (x_2, y_2), ...$, or $\langle x_1, y_1\rangle, \langle x_2, y_2\rangle, ...$

`Sample Mean`: Given a dataset (collection of realizations) $x_1, x_2, ..., x_n$ of X, the sample mean is:

$$\bar{x}_n = \frac{1}{n}\sum_i x_i$$

Given a dataset, $\bar{x}_n$ is a fixed number. It is usually a good estimate of the expected value of a random variable X with an unknown distribution.

`Sample space` $\mathcal{S}$ is the set of all possible events we might observe. Depends on context.
- Coin flips: $\mathcal{S} = \{h, t\}$
- Eruption times: $\mathcal{S} = \mathbb{R}^{\geq 0}$
- (Eruption times, Eruption waits): $\mathcal{S} = \mathbb{R}^{\geq 0} \times \mathbb{R}^{\geq 0}$

`An event` is a subset of the sample space.
- Observe heads: $\{h\}$
- Observe eruption for 2 minutes: $\{2.0\}$
- Observe eruption with length between 1 and 2 minutes and wait between 50 and 70 minutes: $[1, 2] \times [50, 70]$.

Random Variables

`Random variable` is a mapping from the event space to a number (or vector)

`Discrete random variable` take values from a countable set

`Continuous random variable` take values in intervals of $\mathbb{R}$
For a continuous r.v. X, $\Pr(X = x) = 0$ for all x. There is no probability mass function.

Probability Mass Function

For a discrete X, $p_X(x)$ gives $\Pr(X = x)$.
Requirement: $\sum_{x\in\mathcal{X}} p_X(x) = 1$.
Note that the sum can have an infinite number of terms.
- Only works on discrete values
- Sum of probabilities in set needs to be 1

Cumulative Distribution function

For a discrete X, $P_X(x)$ gives $\Pr(X \leq x)$.
Requirements:
- P is nondecreasing
- $\sup_{x\in\mathcal{X}} P_X(x) = 1$

Note: - $P_X(b) = \sum_{x\leq b} p_X(x)$
- $\Pr(a < X \leq b) = P_X(b) - P_X(a)$

Probability Density Function

For continuous X, $\Pr(X = x) = 0$ and PMF does not exist. However, we define the Probability Density Function f_X:
- $\Pr(a \leq X \leq b) = \int_a^b f_X(x)\,\mathrm{d}x$

Requirement:
- $\forall x\ f_X(x) > 0$, $\int_{-\infty}^{\infty} f_X(x)\,\mathrm{d}x = 1$
- Requires a range, doesnt work with discrete values

Cumulative Distribution Function

For a continuous X, $F_X(x)$ gives $\Pr(X \leq x) = \Pr(X \in (-\infty, x])$.
Requirements:
- F is nondecreasing
- $\sup_{x\in\mathcal{X}} F_X(x) = 1$

Note:
- $F_X(x) = \int_{-\infty}^{x} f_X(x)\,\mathrm{d}x$
- $\Pr(x_1 < X \leq x_2) = F_X(x_2) - F_X(x_1)$

Supervised Learning

Definitions

`Supervised Learning` is when correct answers are given to a training model and it uses them to make future perdictions following similar patterns

`Columns` are called input variables or features or attributes

`Labels` or output variables are the outcome we are trying to perdict

`Training Example` or instance is a row in a table

`Data Set` is the whole table

h is called a predictive model or `hypothesis`

`Classification` Is this A, B or C? (Binary if only 2 categories)
`Regression` How much or how many?
`Clustering` How is this organized?
`Reinforcement` What should i do next?

Example Problem

Given a data set $D \subset (\mathcal{X} \times \mathcal{Y})^n$, find a function:

$$h : \mathcal{X} \to \mathcal{Y}$$

such that $h(\mathbf{x})$ is a good predictor for the value of y.

Steps to solve a supervised learning problem

1. Decide what the input-output pairs are.
2. Decide how to encode inputs and outputs. This defines the input space $\mathcal{X}$, and the output space $\mathcal{Y}$.
3. Choose model space/hypothesis class $\mathcal{H}$.
4. Choose an error function (cost function) to define the best model in the class
5. Choose an algorithm for searching efficiently through the space of models to find the best.

Linear Hypothesis

Suppose y was a linear function of $\mathbf{x}$:

$$h_{\mathbf{w}}(\mathbf{x}) = w_0 + w_1x_1 + w_2x_2 + \cdots$$

w_i are called parameters or weights. Typically include an attribute $x_0 = 1$ (also called bias term or intercept term)

Choosing Weights

Error Minimization

Define an error function or a cost function to measure how much our prediction differs from the 'true' answer on the training data. The weights should make the prediction as close to the true values as possible.

Picking the error function

Least Means Squares

Draw a linear line such that the sum of the distance between each the line and each training point (error) is as small as possible.

Linear Regression Summary

The optimal solution (minimizing sum-squared-error) can be computed in polynomial time in the size of the data set.
A very rare case in which an analytical, exact solution is possible

Improving Linear Regression

1. Explicitly transform the data, i.e. create additional features
- Add cross-terms, higher-order terms
- More generally, apply a transformation of the inputs from $\mathcal{X}$ to some other space $\mathcal{X}'$, then do linear regression in the transformed space
2. Use a different model space/hypothesis class

Performance Evaluation

Performance of a Fixed Hypothesis

Define the loss (error) of the hypothesis on an example $(\mathbf{x}, y)$ as

$$L(h(\mathbf{x}), y)$$

Given a model h, (which could have come from anywhere), its generalization error is:

$$E[L(h(\mathbf{X}), Y)]$$

Given a set of data points $(\mathbf{x}_i, y_i)$ that are realizations of $(\mathbf{X}, Y)$, we can compute the empirical error

$$\bar{\ell}_{h,n} = \frac{1}{n}\sum_{i=1}^{n} L(h(\mathbf{x}_i), y_i)$$

Sample Mean

Given a dataset, $\bar{x}_n$ is a fixed number. We use $\bar{X}_n$ to denote the random variable corresponding to the sample mean computed from a randomly drawn dataset of size n.

Statistics, Parameters, and Estimation

A `statistic` is any summary of a dataset. (E.g. $\bar{x}_n$, sample median.)
A statistic is the result of a `function` applied to a dataset.

A `parameter` is any summary of the distribution of a random variable. (E.g. μ_X, median.)
A parameter is the result of a `function` applied to a distribution.

`Estimation` uses a `statistic` (e.g. $\bar{x}_n$) to estimate a `parameter` (e.g. μ_X) of the `distribution` of a `random variable`.
- `Estimate`: value obtained from a specific dataset
- `Estimator`: function (e.g. sum, divide by n) used to compute the estimate
- `Estimand`: parameter of interest

The distribution of an estimator is called a `sampling distribution`

Bias

The expected difference between estimator and parameter. For example,

$$E[\bar{X}_n - \mu_X]$$

If 0, estimator is unbiased.

Sometimes, $\bar{x}_n > \mu_X$, sometimes $\bar{x}_n < \mu_X$, but the long run average of these differences will be zero.

Variance

The expected squared difference between estimator and its mean

$$E[(\bar{X}_n - E[\bar{X}_n])^2]$$

- Positive for all interesting estimators.
- For an unbiased estimator

$$E[(\bar{X}_n - \mu_X)^2]$$

- Sometimes, $\bar{x}_n > \mu_X$, sometimes $\bar{x}_n < \mu_X$, but the `squared differences` are all positive and do not cancel out.

Normal Distribution

Normal distribution is defined by two parameters: μ_X, σ_X^2.
For an estimator like $\bar{X}_n$, if we know $\mu_{\bar{X}_n}$ and $\sigma^2_{\bar{X}_n}$, then we can say a lot about how good it is.

Central Limit Theorem

The sampling distribution of $\bar{X}_n$ is approximately normal if n is big enough.

$$\sigma_n^2 = \frac{\sigma^2}{\sqrt{n}}$$

is called the `standard error` and σ^2 is the variance of X.

Confidence Interval

Typically, we specify confidence given by $1-\alpha$
Use the sampling distribution to get an interval that traps the parameter (estimand) with probability $1-\alpha$.

Test Sets

Training error underestimates generalization error. It is a `biased estimator`.
Create a test set Possibly of size $n = (1.96)^2\frac{\sigma_L^2}{d^2}$ where σ_L^2 is the variance of the loss (which has to be guessed or estimated from training) and d is half-width of a 95% confidence interval.

Model Selection

Overfitting

Larger model spaces *always* lead to lower training error.
Small training error, large generalization error is known as `overfitting`

Strategy 1: A Validation Set

A separate `validation set` can be used for model selection.
- Train on the training set using each proposed model space
- Evaluate each on the validation set, identify the one with lowest *validation* error
- Choose the simplest model with performance < 1 std. error worse than the best.

Problems with Single-Partition Approach

Pros:
- Measures what we want: Performance of the actual learned model.
- Simple
Cons:
- Smaller effective training sets make performance and performance estimates more variable.
- Small validation sets can give poor model selection
- Small test sets can give poor estimates of performance
- For a test set of size 100, with 60 correct classifications, 95

k-fold cross-validation

Divide the instances into k disjoint partitions or folds of size n/k
Loop through the partitions $i = 1...k$:
- Partition i is for evaluation (i.e., estimating the performance of the algorithm after learning is done)
- The rest are used for training (i.e., choosing the specific model within the space)

'Cross-Validation Error' is the average error on the evaluation partitions. Has lower variance than error on one partition.

As with a single validation set, select most parsimonious model whose error is no more than one standard error above the error of the best model.

Summary for Model Selection

The training error decreases with the complexity (size) of the model space
Generalization error decreases at first, then starts increasing
Set aside a validation set helps us find a good model space
We then can report unbiased error estimate, using a test set
Cross-validation is a lower-variance but possibly biased version of this approach. It is standard.

Classification

Linear Methods for Classification

Classification tasks
Loss functions for classification
Logistic Regression
Support Vector Machines

Logistic Regression

Determines the probaability of a feature occuring based on some numeric features.
Perdict the probability (0 to 1) of anything occuring
Specify dependent and independent features
Can classify groups of data using probability categories like all above 0.5 in one and below 0.5 in another

Decision Boundary

Just cut the data into parts using parts some summary of the data
Cut off the data where Y is 1 and x is greater than 0.25
Class = R if $\Pr(Y = 1|X = x) > 0.25$
Cut off the data where Y is 1 and x is greater than 0.5
Class = R if $\Pr(Y = 1|X = x) > 0.5$

Linear Support Vector Machines

Linear classifiers that focus on learning the decision boundary rather than the conditional distribution $P(Y = y|\mathbf{X} = \mathbf{x})$

Perceptrons

If the data is linearly seperatable, the perceptron will find the solution given infinte iterations
Blindingly Fast
Solutions are non-unique

Perceptron Learning Rule

Initialize $\mathbf{w}$ and w_0 randomly
While any training examples remain incorrecty classified
Loop through all misclassified examples
For misclassified example i, perform the updates:

$$\mathbf{w} \leftarrow \mathbf{w} + \delta y_i \mathbf{x}_i, \quad w_0 \leftarrow w_0 + \delta y_i$$

where δ is a step-size parameter.

Support Vector Machines

A optimization criterion (the "margin") guarantees uniqueness and has theoretical advantages
Natural handling nonseparable data by allowing mistakes
An efficient way of operating in expanded feature spaces: "kernel trick"
SVMs can also be used for multiclass classification and regression.

Soft Margin Classifiers

Constraints are relaxed and misclassifiations are allowed.

Visualizations

`Scatterplot Matrix` shows the relation between all variables and is a good first step.
`Correlation Plot/Matrix` shows how two variables are correlated or dependent on each other

Non-Linear Models

Bias-Variance of K-NN (K Nearest Neighbors)

If k is low, very non-linear functions can be approximated, but we also capture the noise in the data Bias is low, variance is high
If k is high, the output is much smoother, less sensitive to data variation High bias, low variance
A validation set can be used to pick the best k

Lazy and Eager Learning

`Lazy` wait for query before generalizing
E.g. Nearest Neighbor
`Eager` generalize before seeing query
E.g. SVM, Linear regression

Pros and Cons of Lazy and Eager

Eager learners must create global approximation
Lazy learners can create many local approximations
An eager learner does the work off-line, summarizes lots of data with few parameters
A lazy learner has to do lots of work sifting through the data at query time
Typically lazy learners take longer time to answer queries and require more space

When to consider Non-parametric methods

When you have: instances that map to points in $\mathbb{R}^p$, not too many attributes per instance (< 20), lots of data
- Advantages:
- Training is very fast
- Easy to learn complex functions over few variables
- Can give back confidence intervals in addition to the prediction
- Often wins if you have enough data
- Disadvantages:
- Slow at query time
- Query answering complexity depends on the number of instances
- Easily fooled by irrelevant attributes
- 'Inference' is not possible

Stop Overfitting in Decision Tree

1. Stop growing the tree when further splitting the data does not yield a statistically significant improvement
2. Grow a full tree, then `prune` the tree, by eliminating nodes

Decision Tree Summary

- Very fast learning algorithms
- Attributes may be discrete or continuous, no preprocessing needed
- Provide a general representation of classification rules
- Easy to understand! Though
- Exact tree output may be sensitive to small changes in data
- With many features, tests may not be meaningful
- In standard form, good for (nonlinear) piecewise axis-orthogonal decision boundaries ÃćâĆňâĂIJ not good with smooth, curvilinear boundaries
- In regression, the function obtained is discontinuous, which may not be desirable
- Good accuracy in practice many applications