Week 2 – Features & Data Processing

1. Properties of Features:

1. Distinctness (= OR !=)
2. Order: (>, <, <=, >=)
3. Meaningful Differences (+ -)
4. Meaningful Ratios (x /)

2. Types of Features:

1. Nominal (Distinctness)
    - gender, postal codes
2. Ordinal (D + Order)
    - grades
3. Interval (+ Meaninful diff)
    - temperatures, dates
4. Ratio (+ Meaningful ratio)
    - Heights, time

All 4 of them can be represented by**discrete** or **continuous** values

Categorical Qualitative: Nominal & Ordinal

Numeric Quantitative: Interval & Ratio

Exercise 1:

```
 Postal codes
     - Discrete
     - Nominal
Gender
     - Binary
     - Nominal
Height
     - Continuous
     - Ratio
Student ID
     - Discrete
     - Ordinal
Grading System
     - Discrete (if its A, B, etc)
     - Ordinal
Date
     - Discrete
     - Interval
```

3. Dataset Characteristics

1. Dimensionality

- Challenges of high-dimensional data, "Curse of dimensionality"

2. Sparsity

- E.g. In bag-of-words, most words will be zero (not used)
- Advantage for computing time and space

3. Resolution

- Patterns depend on the scale
- E.g. travel patterns on scale of hours, days, weeks

4. Possible Issues with Dataset

Low quality dataset/features lead to poor models

- E.g., a classifier build with poor data/features may incorrectly diagnose a patient as being sick when he/she is not

Possible issues with dataset/features

1. Noise

- Refers to random error/variance in original values

- recording of a concert with background noise
  - Check-in data on social media with GPS errors
- Want to **remove** them (e.g. noise reduction/removal)

2. Outliers

- **Anomalous objects:** *observations* with characteristics that are considerably different than most other observations in data set
- **Anomalous values:** *Feature values* that are unusual w.r.t. to typical values for that feature
- E.g. sudden increase in web traffic, large & odd online purchases
- Want to identify them (anomaly detection)

3. Missing values

Reasons - Incomplete data collection - e.g. People not providing annual income - Features not applicable to certain observations - e.g. annual income not applicable to students

Types - Missing completely at random - e.g. data collection is randomly lost - Missing at Random - Missing values related to **some other** features - e.g. older adults not providing annual income - Missing Not at random - Missing values related to **unobserved** features - e.g. not knowing age & income

What to do? - Eliminate observations/variable - OK for missing completely at random. But may not be ok for the other two - Need to **understand** the effects of this elimination - Estimate missing values - Using averages in time series or spatial data - Ignore missing values during analysis - e.g. KNN using features with values

4. Duplicate data

- Deal with these duplicates during data cleaning

5. Wrong/Inconsistent data

- Examples
  - user-provided street name and postal code not matching
  - user-provided street name and postal code not matching
- Ways to overcome:
  - More stringent data collection
  - E.g., drop-down list for specific data input
- Detect potentially wrong data values
  - E.g., allowable range for specific features
- Correction of wrong/inconsistent values
  - E.g., correct postal code based on block number and street name

Exercise 2

Consider a dataset with the issues of noise, outliers, duplicate observations, missing values and wrong/inconsistent data.

What would be the possible problems of applying the k-nearest neighbors (KNN) algorithm on this dataset?

Note: KNN assigns an observation to the class label of the k-nearest neighbor with majority voting

```
 - Noise/Outliers:
    If k-value is too small, may be overly sensitive to noise/outliers
- Duplicate observations:
    K-nearest neighbors may be all duplicates
- Missing/wrong/inconsistent:
    Distance measure may be inaccurate
```

5. Data Preprocessing

Aggregation

- Combining two or more features (or observations) into a**single** feature
- Purpose:
  - Data reduction (reduce #features)
  - Change scale (days aggregated to weeks/months/year)
  - More 'stable' data (less variability)

Sampling

- Main technique for **data reduction**
- Expensive to **obtain** entire set of relevant data (use random survey instead)
- Expensive to **process** entire set
- A sample is **representative** if it has approximately the same properties as the original set of data
- Types of Sampling
  - Simple Random Sampling
  - equal probability of selecting any item
  - Stratified sampling
  - split data into several partitions and draw random samples from each partitions

Dimensionality Reduction

- Curse of Dimensionality
  - As #features increases, more data is needed for an accurate model (as data gets increasingly sparse in the space it occupies)
- Purpose
  - Avoid curse of dimensionality
  - Reduce amt of time/memory required
  - Allow for easier visualisation
  - Eliminate noise/irrelvant features
- Techniques:
  - PCA, SVD
  - Feature selection

Feature Subset Selection

- Another way to reduce dimensionality of data
- Redundant features
  - Duplicate much information contained in one or more features
  - E.g., income level, CPF contributions and income tax
- Irrelevant features
  - Contain no useful information for the data science task
  - E.g. NRIC for predicting person's chance of falling sick
- Approaches:
  - Embedded approaches:
  - As part of classification algorithm (e.g. selection of features when building decision trees)
  - Filter Approaches:
  - Independent feature selection process before applying algorithm
  - E.g. based on their correlation with class labels
  - Wrapper Approaches:
  - Search for best feature subset for a specific algorithm (more expensive than filter)
  - E.g. recursive feature elimination

Feature Creation

- Create new attributes that can capture the important information in a data set much more efficiently than the original attributes

Discretization & Binarization

- Discretization:
  - Converting a continuous attribute into an ordinal attribute
  - Many classification algorithms work best if both the independent and dependent variables have only a few values
  - E.g. instead of using (continuous) savings, we have (discrete) savings levels like low, med, high
- Binarization:
  - Mapping a categorical attribute into one or more binary variables

6. Text Processing

Tokenization

- segment text into tokens
- **Token** = a sequence of characters in a particular document at a particular position
- **Tokenization is language dependent**

Stopwords

- Generally exclude **high-frequency** words (e.g. "a", "the", "in", etc)
- Stopwords are language dependent

Token Normalization

- Reducing multiple tokens to the **same canonical term**, such that matches occur despite superficial differences.
- E.g. USA = U.S.A = usa

Lemmatization

- Reduce inflectional/variant forms to base form
- Direct impact on vocabulary size
- Examples:
  - am, are, is -> be
  - car, cars, car's, cars' -> car
- Need a list of grammatical rules + a list of irregular words

Stemming

- Reduce tokens to “root” form of words to recognize morphological variation
  - E.g. “computer”, “computational”, “computation” all reduced to same token “compute”
- Stemming “blindly” strips off known affixes (prefixes and suffixes) in an iterative fashion

Lemmatization	Stemming
Need to have detailed dictionaries	Cut off end or beginning of word
Lemma is base form of all its inflectional forms but stem isn't	Considers a list of common prefix / suffix
Slower	**Faster**

7. Text Representation

- Some basic terms:
  - **Syntax**: the allowable structures in the language: sentences, phrases, affixes (-ing, -ed, -ment, etc.).
  - **Semantics**: the meaning(s) of texts in the language.
  - **Part-of-Speech (POS)**: the category of a word (noun, verb, preposition etc.).
  - **Bag-of-words (BoW)**: a featurization that uses a vector of word counts (or binary) ignoring order.
  - **N-gram**: for a fixed, small N (2-5 is common), an n-gram is a consecutive sequence of words in a text.

7.1 Bag of Words Featurization

```
 Sentence: The   cat     sat     on      the     mat
word id:  #1     12     5       #3      #1      14

BOW featurization would be the vector:
Vector:      2, 0,  1, 0, 1 , 0, ...
position    #1     #3
```

- Original word order is lost

7.2 Document Collection

- Collection of *n* documents can be represented in the vector space model by a **term-document matrix**
- Entry in matrix corr to 'weight' of a term in the document
    - Zero = term has no significance in document or term does not exist in document
- row --> D1, D2 ...
- col --> T1, T2 ...

7.3 N-grams

- N-grams tries to capture the **word order** by modeling tuples of **consecutive** words
- The unigrams have higher counts and are able to detect influences that are **weak**, while bigrams and trigrams capture **strong** influences that are more specific.
    - e.g. “the white house” will generally have very different influences from the sum of influences of “the”, “white”, “house”.
- N-grams pose some challenges in feature set size. If the original vocabulary size is `|V|`, the number of 2-grams is `|V|2` While for 3-grams it is `|V|3`.
- Luckily natural language n-grams (including single words) have a **power law frequency** structure. This means that most of the ngrams you see are common. A dictionary that contains the **most common** n-grams will cover most of the n-grams you see
    - Because of this you may see values like this:
    - Unigram dictionary size: `40,000`
    - Bigram dictionary size: `100,000`
    - Trigram dictionary size: `300,000`
    - With coverage of `> 80%` of the features occurring in the text.

7.4 Skip-grams

- We can also analyze the meaning of a particular word by looking at the contexts in which it occurs.
- A skip-gram is a set of non-consecutive words (with specified offset), that occur in some sentence.

7.5 TF-IDF

- More frequent terms in a document are more indicative of the topic.
- `tf= f / max(freq_ls)`
- Terms that appear in many different documents are less indicative of overall topic

- TFIDF -> gives a weight to the terms