Chapter 2 : Numeric Features

Dataset

	F1	F2	F3	⋯	Fn
D1					
D2					
D3					
⋮					
Dm					

magnitude matters? — No — Binarization

Yes

feature spans several orders of magnitude? — Yes — Binning

- Fixed-width
- Quantile binning — good if there are large gaps in the counts

feature has a heavy-tailed distribution? — Yes — Log transformation

compresses the range of large numbers
expands the range of small numbers

For models that are smooth functions of the input? ex: linear regression, logistic regression, involves matrix — Yes

- min-max scaling
- standardization
- l^2 normalization

Chapter 3. 4 Text

Bag-of-n-Grams

original text: it is a puppy and it is extremely cute

n = 1

it	is	a	puppy	and	extremely	cute	.	. . .
2	2	1	1	1	1	1	0	0 . . . . .

n = 2

it is	is a	a puppy	puppy and	and it	
1	1	1	1	1	. . . .

⋮

As n increasing,

represent more information

- vectors more sparse
- more expensive to compute

Filtering

- filter out frequent words not easy to set a cutoff number
- filter out rare words
  rare words could be noise
- Stemming

Phrase Extraction

collocation: the concept of a useful phrase

how to find collocations?

- method 1 Frequency-based methods

  Simply look at the most frequently occurring n-grams

   - easy

   - the most frequent combination not useful

- method 2 Hypothesis testing

  null hypothesis: Word 1 appears independent from Word 2.

  alternate hypothesis: Seeing word 1 changes the likelihood of ~~seeing~~ seeing word 2

  final statistic: $\log \lambda = \log \frac{L(\text{Data}; H_{\text{null}})}{L(\text{Data}; H_{\text{alternative}})}$

  algorithm:

  1. compute occurrence probabilities $P(w)$
  2. compute conditional pairwise word occurrence for all bigrams $P(w_2|w_1)$
  3. compute the likelihood ratio $\log \lambda$ for all unique bigrams.
  4. sort the bigrams based on their likelihood ratio
  5. take the bigrams with the smallest likelihood ratio values

- method 3 Chunking and Part-of-Speech Tagging

  rule based models

Tf-Idf (Term Frequency - Inverse Document Frequency)

- bow (w, d) = # times word w appears in document d
  (bag of word)

  tf-idf (w, d) = bow (w, d) / (# documents in which word w appears)

  ↓ refine with log transformation

  $$\text{tf-idf}(w, d) = \text{bow}(w, d) * \log \frac{N}{\text{\# documents in which word } w \text{ appears}}$$

- Tf-idf does not changes the column space of the data matrix

Chapter 5
Categorical Variables

Encoding

- One-Hot Encoding
  - 👍 able to accommodate missing data
  - 👎 redundant

	e_1	e_2	e_3	......	e_n
Category 1	1	0	0	...	0
Category 2	0	1	0	...	0
Category 3	0	0	1	....	0
⋮					
Category n	0	0		- - - -	1

- Dummy Coding
  - 👍 not redundant
  - 👎 cannot handle missing data

	e_1	e_2	...	e_{n-1}
Category 1	1	0	...	0
Category 2	0	1	....	0
⋮				
Category n	0	0	....	0 (reference category)

- Effect Coding
  - 👍 not redundant
  - 👎 cannot handle missing data

	e_1	e_2	....	e_{n-1}
Category 1	1	0	....	0
Category 2	0	1	....	0
⋮				
Category n	-1	-1	....	-1 (reference category)

- When # categories grows ↑

Feature Hashing

Compress the original feature vector into an m-dimensional vector by applying a hashing function to the feature ID

Bin Counting

compute the association staties between that value and the target that we wish to predict.

toy data

user	……	click or not
alice		✓
alice		✓
alice		×
Bob		✓
Sam		×
Sam		✓
Bob		×
Bob		×
Bob		×

$\Rightarrow$

user

$$\begin{gathered} \frac{2}{2} \\ \frac{2}{1} \\ \frac{2}{1} \\ \frac{1}{3} \\ 1 \\ 1 \\ \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{gathered}$$

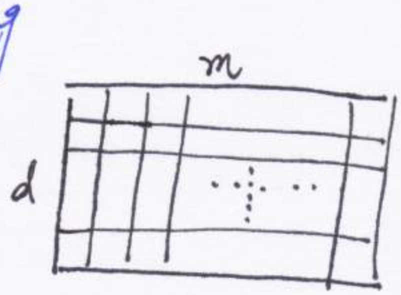
alice $\Rightarrow \left(\frac{2}{3}\right) / \left(\frac{1}{3}\right)$

Sam $\Rightarrow \left(\frac{1}{2}\right) / \left(\frac{1}{2}\right)$

Bob $\Rightarrow \left(\frac{1}{4}\right) / \left(\frac{3}{4}\right)$

Feature Hashing + Bin Counting

$\Rightarrow$ Count-min Sketch



For each data point's feature:

For each row j of the table,

apply the corresponding hash-function to obtain a column index $k = h_j(i)$.

increase the value in row j, column k by 1

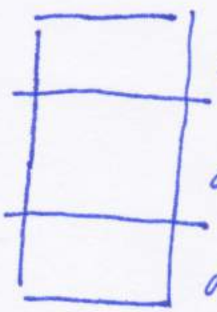
The estimated count is given by the least value in the table

For feature i $\hat{a}_i = \min_j \text{count}[j, h_j(i)]$, where count is the table.

Data Leakage ?

Bin Counting might lead to data leakage since it uses the target variable to compute the statistic. The target variable is what model tries to predict.

Solutions

1. 

2. A statistic is approximately leakage proof if its distribution stays roughly the same with or without any one data point.

   In practice, adding a small noise with distribution Laplace(0, 1) is sufficient to cover up potential leakage from a single data point.

Chapter 6:
Dimensionality Reduction

Projection on a coordinate

$Z = X^T V$ where X: data point
V: new coordinate to project onto

$\downarrow$ more than one data point

$Z = \mathbb{X} V$ where $\mathbb{X}$: data matrix $(m \times d)$ (m → # data points, d → # features)
V: new coordinate to project onto

Variance of a random variable Z

$Var(Z) = E[Z - E(Z)]^2$ intuition for variance:
the degree to which the values of a random variable differ from the expected value.

$\downarrow$ center the data
$E(Z) = 0$

$Var(Z) = E[Z]^2$

PCA

objective function

$$\max_w \sum_{i=1}^{n} (x_i^T w)^2 \quad \text{s.t. } w^T w = 1$$

intuition for the objective function:
maximize the sum of variance of the dataset

$\Rightarrow \max_w w^T \mathbb{X}^T \mathbb{X} w$

$\mathbb{X}^T\mathbb{X}$ is a symmetric matrix (positive semidefinite)
so it has an eigenvalue decomposition form
$S = Q \Lambda Q^T$

$\Rightarrow \max_w w^T Q \Lambda Q^T w$

$\Rightarrow \max_u u^T \Lambda u \quad \|u\| = \|w^T Q\| = 1$

$\Rightarrow \max \sum_{i=1}^{d} \lambda_i u_i^2$ we want to maximize $\sum_{i=1}^{d} \lambda_i u_i^2$ under constraints that $\sum_{i=1}^{d} u_i^2$. then the best is to set $u_1 = 1$ and $u_i = 0$ for $i > 1$

$\Rightarrow w^* = Qe_1 = q_1$ (first column of Q)

Objective Function for $(k+1)$th principal component

$$\max_w w^T X^T X w$$

s.t. $w^T w = 1$

$w_1^T w_1 = w_2^T w_2 = \cdots = w_k^T w_k = 0$

Eigenvector

intuition: every square matrix, no matter what numbers it contains, must map a certain set of vectors back to themselves with some scaling.

square matrix

$$\underset{\text{a certain set of vectors}}{A \underline{v}} = \overset{\text{some scaling}}{\underline{\lambda}} v$$

Singular Vectors

Intuition: a rectangular matrix maps a set of input vectors into a corresponding set of output vectors and its transpose maps those outputs back to the original inputs

rectangular matrix

$$\underline{A}\underbrace{\underline{v}}_{\text{input vector}} = \sigma \overbrace{\underline{u}}^{\text{output vector}} \qquad \overbrace{\underline{A}^T}^{\text{transpose}} u = \sigma v$$

Chapter 7:
K-Means Model Stacking

K-Means

Objective Function

$$\min_{\substack{C_1 \dots C_k \\ \mu_1 \dots \mu_k}} \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|_2$$

use Eucliden distance as metric

$$\mu_i = \sum_{x \in C_i} x / n_i$$

K-Means Featurization

Original features

$$\begin{array}{l} \quad F_1 \ F_2 \ F_3 \cdots F_n \\ d_1 \\ d_2 \\ \vdots \\ d_m \end{array}$$

$\Rightarrow$

K-means

$\Rightarrow$

One-hot vectors

$$\begin{array}{c|ccccc} & F_1' & F_2' & F_3' & \cdots & F_k' \\ d_1 & 1 & 0 & 0 & \cdots & 0 \\ d_2 & 0 & \cdots & & & 0 \\ \vdots & & & & & \\ d_m & 0 & 1 & \cdots & & 0 \end{array}$$

Dense vectors (representing inverse distance to each cluster)

$$\begin{array}{l} \quad F_1' \ F_2' \ F_3' \cdots F_k' \\ d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_m \end{array}$$

$$ele[d_i][F_j] = \frac{1}{\|d_i - \mu_j\|}$$