Lecture 14: Feature Selection and Analysis

COMP90049

Semester 2, 2020

Lida Rashidi, CIS

Acknowledgement: Jeremy Nicholson, Tim Baldwin & Karin Verspoor

Features in Machine Learning

Where we're at so far

We want to get knowledge out of a data set:

Outlook	Temperature	Humidity	Windy	Play
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
overcast	hot	high	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
⋮	⋮	⋮	⋮	⋮

Machine Learning, revisited

We want to get knowledge out of a data set:

- Data mining
- Machine learning
  - Supervised machine learning $\leftarrow$ today (mostly)
  - Unsupervised machine learning

Machine Learning, revisited

How to do (supervised) Machine Learning:

0. Get hired!
1. Pick a feature representation
2. Compile data
3. Pick a (suitable) model
4. Train the model
5. Classify development data, evaluate results
6. Probably: *go to (1)*

Machine Learning, revisited

Our job as Machine Learning experts:

- Choose a model suitable for classifying the data according to the attributes
- Choose attributes suitable for classifying the data according to the model
    - Inspection
    - Intuition

Machine Learning, revisited

Our job as Machine Learning experts:

- Choose a model suitable for classifying the data according to the attributes
- Choose attributes suitable for classifying the data according to the model

  - ~~Inspection~~
  - ~~Intuition~~
  - Neither possible in practice

Feature Selection

What makes features good?

Better models!

- Better performance according to some evaluation metric

Side-goal:

- Seeing important features can suggest other important features
- Tell us interesting things about the problem

Side-goal:

- Fewer features $\rightarrow$ smaller models $\rightarrow$ faster answer
  - More accurate answer $>>$ faster answer

Choosing a good feature set

"Wrapper" methods:

- Choose subset of attributes that give best performance on the development data
- For example: for the Weather data set:
  - Train model on {`Outlook`}
  - Train model on {`Temperature`}
  - ...
  - Train model on {`Outlook`, `Temperature`}
  - ...
  - Train model on {`Outlook`, `Temperature`, `Humidity`}
  - ...
  - Train model on {`Outlook`, `Temperature`, `Humidity`, `Windy`}

Choosing a good feature set

"Wrapper" methods:

- Choose subset of attributes that give best performance on the development data
- For example: for the Weather data set:
  - Evaluate model on {`Outlook`}
  - Evaluate model on {`Temperature`}
  - ...
  - Evaluate model on {`Outlook, Temperature`}
  - ...
  - Evaluate model on {`Outlook, Temperature, Humidity`}
  - ...
  - Evaluate model on {`Outlook, Temperature, Humidity, Windy`}
- Best performance on data set $\rightarrow$ best feature set

Choosing a good feature set

"Wrapper" methods:

- Choose subset of attributes that give best performance on the development data
- Advantages:
  - Feature set with optimal performance on development data
- Disadvantages:
  - Takes a **long** time

Aside: how long does the full wrapper method take?

Assume we have a fast method (e.g. Naive Bayes) over a data set of non-trivial size ($\sim$10K instances):

- Assume: train–evaluate cycle takes 10 sec to complete

How many cycles? For m features:

- 2^m subsets $= \frac{2^m}{6}$ minutes
- $m = 10 \rightarrow 3$ hours
- $m = 60 \rightarrow$ heat death of universe

Only practical for very small data sets.

More practical wrapper methods

Greedy approach:

- Train and evaluate model on each single attribute
- Choose best attribute
- Until convergence:
  - Train and evaluate model on best attribute(s), plus each remaining single attribute
  - Choose best attribute out of the remaining set
- Iterate until performance (e.g. accuracy) stops increasing

More practical wrapper methods

Greedy approach:

- Bad news:
  - Takes $\frac{1}{2}m^2$ cycles, for m attributes
  - In theory, 386 attributes $\rightarrow$ days
- Good news:
  - In practice, converges much more quickly than this
- Bad news again:
  - Convergences to a sub-optimal (and often very bad) solution

More practical wrapper methods

“Ablation” approach:

- Start with all attributes
- Remove one attribute, train and evaluate model
- Until divergence:
  - From remaining attributes, remove each attribute, train and evaluate model
  - Remove attribute that causes least performance degredation
- Termination condition usually: performance (e.g. accuracy) starts to degrade by more than ϵ

More practical wrapper methods

"Ablation" approach; for example:

- Start with all features
  - Train, evaluate model on {`Outlook, Temperature, Humidity, Windy`}
- Consider feature subsets of size 3:
  - Train, evaluate model on {`Outlook, Temperature, Humidity`}
  - Train, evaluate model on {`Outlook, Temperature, Windy`}
  - Train, evaluate model on {`Outlook, Humidity, Windy`}
  - Train, evaluate model on {`Temperature, Humidity, Windy`}
- Choose best of previous five (let's say THW):
- Consider feature subsets of size 2:
  - Train, evaluate model on {`Temperature, Humidity`}
  - Train, evaluate model on {`Temperature, Windy`}
  - Train, evaluate model on {`Humidity, Windy`}
- etc...

More practical wrapper methods

"Ablation" approach:

- Good news:
  - Mostly removes irrelevant attributes (at the start)
- Bad news:
  - Assumes independence of attributes (both approaches)
  - Actually does take $O(m^2)$ time; cycles are slower with more attributes
  - Not feasible on non-trivial data sets.

Feature filtering

Intuition: possible to evaluate “goodness” of each feature, separate from othe features

- Consider each feature separately: linear time in number of attributes
- Possible (but difficult) to control for inter-dependence of features
- Typically most popular strategy

Filtering methods

Feature "goodness"

What makes a ~~feature set~~ single feature good?

- ~~Better models!~~
- Well correlated with class

a_1	a_2	c
Y	Y	Y
Y	N	Y
N	Y	N
N	N	N

Which of a_1, a_2 is good?

Toy example

a_1	a_2	c
Y	Y	Y
Y	N	Y
N	Y	N
N	N	N

a_1 is probably good.

Toy example

a_1	a_2	c
Y	Y	Y
Y	N	Y
N	Y	N
N	N	N

a_2 is probably not good.

Pointwise Mutual Information

Recall independence, equivalently:

$$P(C|A) = P(C)$$

We clearly want attributes that are **not** independent from class.

$$\frac{P(A, C)}{P(A)P(C)} = 1$$

- If LHS $>>$ 1, attribute and class occur together much more often than randomly.
- If LHS $\sim$ 1, attribute and class occur together as often as we would expect from random chance
- (If LHS $<<$ 1, attribute and class are negatively correlated. More on this later.)

Pointwise Mutual Information

Pointwise mutual information:

$$PMI(A, C) = \log_2 \frac{P(A, C)}{P(A)P(C)}$$

Attributes with greatest PMI: best attributes

Toy example, revisited

a_1	a_2	c
Y	Y	Y
Y	N	Y
N	Y	N
N	N	N

Calculate PMI of a_1, a_2 with respect to c

Toy example, revisited

a_1	a_2	c
Y	Y	Y
Y	N	Y
N	Y	N
N	N	N

$P(a_1) = \frac{2}{4}$; $P(c) = \frac{2}{4}$; $P(a_1, c) = \frac{2}{4}$

Toy example, revisited

a_1	a_2	c
Y	Y	Y
Y	N	Y
N	Y	N
N	N	N

$$
\begin{aligned}
PMI(a_1, c) &= \log_2 \frac{\frac{1}{2}}{\frac{1}{2} \cdot \frac{1}{2}} \\
&= \log_2(2) = 1
\end{aligned}
$$

Toy example, revisited

a_1	a_2	c
Y	Y	Y
Y	N	Y
N	Y	N
N	N	N

$P(a_2) = \frac{2}{4}$; $P(c) = \frac{2}{4}$; $P(a_1, c) = \frac{1}{4}$

Toy example, revisited

a_1	a_2	c
Y	Y	Y
Y	N	Y
N	Y	N
N	N	N

$$
\begin{aligned}
PMI(a_2, c) &= \log_2 \frac{\frac{1}{4}}{\frac{1}{2} \cdot \frac{1}{2}} \\
&= \log_2(1) = 0
\end{aligned}
$$

Feature "goodness", revisited

What makes a single feature good?

- Well correlated with class
  - Knowing a lets us predict c with more confidence
- Reverse correlated with class
  - Knowing $\bar{a}$ lets us predict c with more confidence
- Well correlated (or reverse correlated) with not class
  - Knowing a lets us predict $\bar{c}$ with more confidence
  - Usually not quite as good, but still useful

Mutual Information

Mutual information: combine each a, $\bar{a}$, c, $\bar{c}$ PMI

Aside: Contingency tables

Contigency tables: compact representation of these frequency counts

	a	$\bar{a}$	Total
c	$\sigma(a, c)$	$\sigma(\bar{a}, c)$	$\sigma(c)$
$\bar{c}$	$\sigma(a, \bar{c})$	$\sigma(\bar{a}, \bar{c})$	$\sigma(\bar{c})$
Total	$\sigma(a)$	$\sigma(\bar{a})$	N

$P(a, c) = \frac{\sigma(a,c)}{N}$, etc.

Aside: Contingency tables

Contingency tables for toy example:

a_1	$a =$Y	$a =$N	Total
$c =$Y	2	0	2
$c =$N	0	2	2
Total	2	2	4

a_2	$a =$Y	$a =$N	Total
$c =$Y	1	1	2
$c =$N	1	1	2
Total	2	2	4

Mutual Information

Mutual information: combine each a, $\bar{a}$, c, $\bar{c}$ PMI

$$\begin{aligned} MI(A,C) &= P(a,c)PMI(a,c) + P(\bar{a},c)PMI(\bar{a},c) + \\ &\quad P(a,\bar{c})PMI(a,\bar{c}) + P(\bar{a},\bar{c})PMI(\bar{a},\bar{c}) \end{aligned}$$

$$\begin{aligned} MI(A,C) &= P(a,c)\log_2 \frac{P(a,c)}{P(a)P(c)} + P(\bar{a},c)\log_2 \frac{P(\bar{a},c)}{P(\bar{a})P(c)} + \\ &\quad P(a,\bar{c})\log_2 \frac{P(a,\bar{c})}{P(a)P(\bar{c})} + P(\bar{a},\bar{c})\log_2 \frac{P(\bar{a},\bar{c})}{P(\bar{a})P(\bar{c})} \end{aligned}$$

Often written more compactly as:

$$MI(A, C) = \sum_{i \in \{a, \bar{a}\}} \sum_{j \in \{c, \bar{c}\}} P(i, j) \log_2 \frac{P(i, j)}{P(i)P(j)}$$

(This representation can be extended to different types of attributes more intuitively.)
Note that $0 \log 0 \equiv 0$.

Mutual Information Example

Contingency table for toy example:

a_1	$a =$Y	$a =$N	Total
$c =$Y	2	0	2
$c =$N	0	2	2
Total	2	2	4

Mutual Information Example

Contingency table for toy example:

a_1	$a =$Y	$a =$N	Total
$c =$Y	2	0	2
$c =$N	0	2	2
Total	2	2	4

$P(a, c) = \frac{2}{4}$; $P(a) = \frac{2}{4}$; $P(c) = \frac{2}{4}$
$P(\bar{a}, \bar{c}) = \frac{2}{4}$; $P(\bar{a}) = \frac{2}{4}$; $P(\bar{c}) = \frac{2}{4}$
$P(\bar{a}, c) = 0$; $P(a, \bar{c}) = 0$

Mutual Information Example

$$
\begin{aligned}
MI(A_1, C) &= P(a_1, c)\log_2 \frac{P(a_1, c)}{P(a_1)P(c)} + P(\bar{a}_1, c)\log_2 \frac{P(\bar{a}_1, c)}{P(\bar{a}_1)P(c)} + \\
&\quad P(a_1, \bar{c})\log_2 \frac{P(a_1, \bar{c})}{P(a_1)P(\bar{c})} + P(\bar{a}_1, \bar{c})\log_2 \frac{P(\bar{a}_1, \bar{c})}{P(\bar{a}_1)P(\bar{c})} \\
&= \frac{1}{2}\log_2 \frac{\frac{1}{2}}{\frac{1}{2}\frac{1}{2}} + 0\log_2 \frac{0}{\frac{1}{2}\frac{1}{2}} + 0\log_2 \frac{0}{\frac{1}{2}\frac{1}{2}} + \frac{1}{2}\log_2 \frac{\frac{1}{2}}{\frac{1}{2}\frac{1}{2}} \\
&= \frac{1}{2}(1) + 0 + 0 + \frac{1}{2}(1) = 1
\end{aligned}
$$

Mutual Information Example

Contingency table for toy example:

a_2	$a =$Y	$a =$N	Total
$c =$Y	1	1	2
$c =$N	1	1	2
Total	2	2	4

$P(a, c) = \frac{1}{4}$; $P(a) = \frac{2}{4}$; $P(c) = \frac{2}{4}$
$P(\bar{a}, \bar{c}) = \frac{1}{4}$; $P(\bar{a}) = \frac{2}{4}$; $P(\bar{c}) = \frac{2}{4}$
$P(\bar{a}, c) = \frac{1}{4}$; $P(a, \bar{c}) = \frac{1}{4}$

THE UNIVERSITY OF MELBOURNE

Mutual Information Example

$$
\begin{aligned}
MI(A_2, C) &= P(a_2, c)\log_2 \frac{P(a_2, c)}{P(a_2)P(c)} + P(\bar{a}_2, c)\log_2 \frac{P(\bar{a}_2, c)}{P(\bar{a}_2)P(c)} + \\
&\quad P(a_2, \bar{c})\log_2 \frac{P(a_2, \bar{c})}{P(a_2)P(\bar{c})} + P(\bar{a}_2, \bar{c})\log_2 \frac{P(\bar{a}_2, \bar{c})}{P(\bar{a}_2)P(\bar{c})} \\
&= \frac{1}{4}\log_2 \frac{\frac{1}{4}}{\frac{1}{2}\frac{1}{2}} + \frac{1}{4}\log_2 \frac{\frac{1}{4}}{\frac{1}{2}\frac{1}{2}} + \frac{1}{4}\log_2 \frac{\frac{1}{4}}{\frac{1}{2}\frac{1}{2}} + \frac{1}{4}\log_2 \frac{\frac{1}{4}}{\frac{1}{2}\frac{1}{2}} \\
&= \frac{1}{4}(0) + \frac{1}{4}(0) + \frac{1}{4}(0) + \frac{1}{4}(0) = 0
\end{aligned}
$$

Chi-square

Similar idea, different solution:
Consider contingency table:

	a	$\bar{a}$	Total
c	$\sigma(a, c)$	$\sigma(\bar{a}, c)$	$\sigma(c)$
$\bar{c}$	$\sigma(a, \bar{c})$	$\sigma(\bar{a}, \bar{c})$	$\sigma(\bar{c})$
Total	$\sigma(a)$	$\sigma(\bar{a})$	N

Consider contingency table (shorthand):

	a	$\bar{a}$	Total
c	W	X	$W + X$
$\bar{c}$	Y	Z	$Y + Z$
Total	$W + Y$	$X + Z$	$N = W + X + Y + Z$

If a, c were independent (uncorrelated), what value would I expect to be in W ($E(W)$)?

Chi-square

a, c independent $\rightarrow P(a,c) = P(a)P(c)$

$$
\begin{aligned}
P(a,c) &= P(a)P(c) \\
\frac{\sigma(a,c)}{N} &= \frac{\sigma(a)}{N}\frac{\sigma(c)}{N} \\
\sigma(a,c) &= \frac{\sigma(a)\sigma(c)}{N} \\
E(W) &= \frac{(W+Y)(W+X)}{W+X+Y+Z}
\end{aligned}
$$

Chi-square

Check the value we actually observed $O(W)$ with the expected value $E(W)$:

- If the observed value is much greater than the expected value, a occurs more often with c than we would expect at random — predictive
- If the observed value is much lesser than the expected value, a occurs less often with c than we would expect at random — predictive
- If the observed value is close to the expected value, a occurs as often with c as we would expect randomly — not predictive

Similarly with X, Y, Z

Chi-square

Actual calculation (to fit to a chi-square distribution):

$$\chi^2 = \frac{(O(W) - E(W))^2}{E(W)} + \frac{(O(X) - E(X))^2}{E(X)} + \frac{(O(Y) - E(Y))^2}{E(Y)} + \frac{(O(Z) - E(Z))^2}{E(Z)}$$

Because the values are squared, χ^2 becomes much greater when $\mid O - E \mid$ is large, even if E is also large.

Chi-square

Actual calculation (written more compactly):

$$\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}}$$

(i sums over rows and j sums over columns.)

In practice, there are simpler ways to calculate this for 2×2 contingency tables.

Chi-square Example

Contingency table for toy example (observed values):

a_1	$a =$Y	$a =$N	Total
$c =$Y	2	0	2
$c =$N	0	2	2
Total	2	2	4

Contingency table for toy example (expected values):

a_1	$a =$Y	$a =$N	Total
$c =$Y	1	1	2
$c =$N	1	1	2
Total	2	2	4

Chi-square Example

$$
\begin{aligned}
\chi^2(A_1, C) &= \frac{(O_{a,c} - E_{a,c})^2}{E_{a,c}} + \frac{(O_{\bar{a},c} - E_{\bar{a},c})^2}{E_{\bar{a},c}} + \\
&\quad \frac{(O_{a,\bar{c}} - E_{a,\bar{c}})^2}{E_{a,\bar{c}}} + \frac{(O_{\bar{a},\bar{c}} - E_{\bar{a},\bar{c}})^2}{E_{\bar{a},\bar{c}}} \\
&= \frac{(2-1)^2}{1} + \frac{(0-1)^2}{1} + \frac{(0-1)^2}{1} + \frac{(2-1)^2}{1} \\
&= 1 + 1 + 1 + 1 = 4
\end{aligned}
$$

$\chi^2(A_2, C)$ is obviously 0, because all observed values are equal to expected values.

Common Issues

Types of Attribute

So far, we've only looked at binary (Y/N) attributes:

- Nominal attributes
- Continuous attributes
- Ordinal attributes

Types of Attribute

Nominal attributes (e.g. `Outlook`={sunny, overcast, rainy}).
Two common strategies:

1. Treat as multiple binary attributes:
    - e.g. `sunny`=Y, `overcast`=N, `rainy`=N, etc.
    - Can just use the formulae as given
    - Results often difficult to interpret
        - For example, `Outlook`=sunny is useful, but `Outlook`=overcast and `Outlook`=rainy are not useful... Should we use `Outlook`?

Types of Attribute

Nominal attributes (e.g. `Outlook`=$\{$sunny, overcast, rainy$\}$).
Two common strategies:

2. Modify contigency tables (and formulae)

O	s	o	r
c =Y	U	V	W
c =N	X	Y	Z

Types of Attribute

Modified MI:

$$
\begin{aligned}
MI(O,C) &= \sum_{i \in \{s,o,r\}} \sum_{j \in \{c,\bar{c}\}} P(i,j) \log_2 \frac{P(i,j)}{P(i)P(j)} \\
&= P(s,c) \log_2 \frac{P(s,c)}{P(s)P(c)} + P(s,\bar{c}) \log_2 \frac{P(s,\bar{c}}{P(s)P(\bar{c})} + \\
&\quad P(o,c) \log_2 \frac{P(o,c)}{P(o)P(c)} + P(o,\bar{c}) \log_2 \frac{P(o,\bar{c})}{P(o)P(\bar{c})} + \\
&\quad P(r,c) \log_2 \frac{P(r,c)}{P(r)P(c)} + P(r,\bar{c}) \log_2 \frac{P(r,\bar{c})}{P(r)P(\bar{c})}
\end{aligned}
$$

- Biased towards attributes with many values. (Why?)

Types of Attribute

Chi-square can be used as normal, with 6 observed/expected values.

- To control for score inflation, we need to consider "number of degrees of freedom", and then use the significance test explicitly (beyond the scope of this subject)

Types of Attribute

Continuous attributes:

- Usually dealt with by estimating probability based on a Gaussian (normal) distribution
- With a large number of values, most random variables are normally distributed due to the **Central Limit Theorem**
- For small data sets or pathological features, we typically need to use messy binomial/multinomial distributions

All of this is (unsurprisingly) beyond the scope of this subject

Types of Attribute

Ordinal attributes (e.g. low, med, high or 1,2,3,4).
Three possibilities, roughly in order of popularity:

1. Treat as binary
   - Particularly appropriate for frequency counts where events are low-frequency (e.g. words in tweets)
2. Treat as continuous
   - The fact that we haven't *seen* any intermediate values is usually not important
   - Does have all of the technical downsides of continuous attributes, however
3. Treat as nominal (i.e. throw away ordering)

Multi-class problems

So far, we've only looked at binary (Y/N) classification tasks.

Multiclass (e.g. LA, NY, C, At, SF) classification tasks are usually much more difficult.

Multi-class problems

What makes a single feature good?

- Highly correlated with class
- Highly reverse correlated with class
- Highly correlated (or reverse correlated) with not class

... What if there are many classes?

Multi-class problems

What makes a feature **bad**?

- Irrelevant
- Correlated with other features
- Good at only predicting one class (but is this truly bad?)

Multi-class problems

Consider multi-class problem over LA, NY, C, At, SF:

- PMI, MI, χ^2 are all calculated *per-class*
- (Some other feature selection metrics, e.g. Information Gain, work for all classes at once)
- Need to make a point of selecting (hopefully uncorrelated) features for *each* class to give our classifier the best chance of predicting everything correctly.

Multi-class problems

Actual example (MI):

LA	NY	C	At	SF
la	nyc	chicago	atlanta	sf
angeles	york	bears	atl	httpdealnaycom
los	ny	il	ga	francisco
chicago	chicago	httpbitlyczmk	lol	san
hollywood	atlanta	cubs	u	u
atlanta	yankees	la	georgia	lol
lakers	sf	chi	chicago	save

Multi-class problems

Intuitive features:

LA	NY	C	At	SF
la	**nyc**	**chicago**	**atlanta**	**sf**
angeles	**york**	**bears**	**atl**	`httpdealnaycom`
los	**ny**	**il**	**ga**	**francisco**
`chicago`	`chicago`	`httpbitlyczmk`	`lol`	**san**
hollywood	`atlanta`	**cubs**	`u`	`u`
`atlanta`	**yankees**	`la`	**georgia**	`lol`
lakers	`sf`	**chi**	`chicago`	`save`

Multi-class problems

Features for predicting not class (MI):

LA	NY	C	At	SF
la	nyc	chicago	atlanta	sf
angeles	york	bears	atl	httpdealnaycom
los	ny	il	ga	francisco
chicago	**chicago**	httpbitlyczmk	lol	san
hollywood	**atlanta**	cubs	u	u
atlanta	yankees	**la**	georgia	lol
lakers	**sf**	chi	**chicago**	save

Multi-class problems

Unintuitive features:

LA	NY	C	At	SF
la	nyc	chicago	atlanta	sf
angeles	york	bears	atl	**httpdealnaycom**
los	ny	il	ga	francisco
chicago	chicago	**httpbitlyczmk**	**lol**	san
hollywood	atlanta	cubs	**u**	**u**
atlanta	yankees	la	georgia	**lol**
lakers	sf	chi	chicago	**save**

What's going on with MI?

Mutual Information is biased toward rare, uninformative features

- All probabilities: no notion of the raw frequency of events
- If a feature is seen rarely, but always with a given class, it will be seen as "good"
- Best features in the Twitter dataset only had MI of about 0.01 bits; 100$^{\text{th}}$ best for a given class had MI of about 0.002 bits

So... Feature selection isn't so great?

No way!

- Even marginally relevant features usually a vast improvement on an unfiltered data set
- Some models **need** feature selection
  - k-Nearest Neighbour, hugely
  - Naive Bayes/Decision Trees, to a lesser extent
- Machine learning experts (us!) need to think about the data!

Summary

Summary

- Wrappers vs. Filters
- Popular filters: PMI, MI, χ^2, how should we use them and what are the results going to look like
- Importance of feature selection for different methods (even though it often isn't the solution we were hoping for)

References

Guyon, Isabelle, and Andre Elisseeff. 2003. An introduction to variable and feature selection. *The Journal of Machine Learning Research*. Vol 3, 1157–1182.

John, George, Ron Kohavi, and Karl Pfleger. 1994. Irrelevant features and the subset selection problem. In *Proceedings of the 11th International Conference on Machine Learning*, 121–9.

Tan, Pang-Ning, Michael Steinbach, and Vipin Kumar. 2006. *Introduction to Data Mining*. Addison Wesley.

Witten, Ian, and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. San Francisco, USA: Morgan Kaufmann.

Yang, Yiming, Jan Pedersen. 1997. A Comparative Study on Feature Selection in Text Categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*, 412–20.

THE UNIVERSITY OF MELBOURNE