F20DL Data Mining and Machine Learning

Diana Bental
(with material from David Corne and slides from http://www.cs.waikato.ac.nz/ml/weka/book.html)

Lecture 4 Preparing Data for Input

- We talked about Input
  - Concepts, Instances and Attributes
  - Practical issues - we already mentioned sparse data and denormalization (in the database sense of the word)
- Now we cover some more practical issues
- Look at a simple classification learning method (so we can use it) 1-Nearest Neighbour
- Data preparation
  - Normalizing values
  - Discretization (binning)
  - Missing Values
- inspection

The main issue for today

- **Dataset preparation**
- There are various things we can and should do to our dataset so that we make it *easier*, or indeed *possible*, to apply our chosen machine learning method
- E.g. we may want to apply the ID3 Decision Tree learning algorithm – but this only works with **categorical** data.
- We may want to apply a Neural Network, but this needs the data to be all numeric, and for all values to be relatively small (e.g. between -1 and 1).
- etc

Example: 1-NN classification

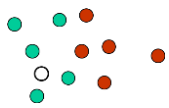
- Almost the simplest possible classification method.
- The `classifier' is just the dataset itself
- Predict the class of a new instance *new* by finding the instance *c* in the dataset that is closest to *new*, and predicting that it will have the same class value as *c*

Visualisation of 1-NN

- Suppose you have data with two numeric attributes and one class attribute , which is either *red* or *green*. We can treat the data as 2D points, coloured by the class. Suppose this is the pre-classified dataset:

Visualisation of 1-NN

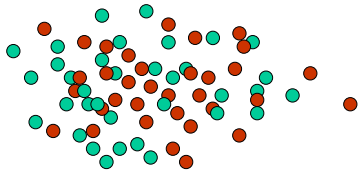
- Suppose you have data with two numeric attirbutes and one class attribute, which is either *red* or *green*. We can treat the data as 2D points, coloured by the class. Suppose this is the pre-classified dataset:



A new unclassified instance comes along – what is its class?

But life is often more interesting (and harder)

- Just an aside to point out that, in many interesting and important classification problems, the situation is more like this:



A bit of notation

A dataset has *N* instances (records) $\{R_1,R_2,\ldots,R_N\}$, … each of which has *F* attributes $\{A_1,A_2,\ldots,A_F\}$.

Attribute A_F is the class attribute – the one we want to predict.

$v(i,j)$ is the value of attribute A_j in record R_i

So, for example, what is

$$\sum_{i=1}^{N}\sum_{j=1}^{F-1} v(i,j)$$

How to work out distance between two instances?

- Depends on the data, and on your common sense. Let's say we are only dealing with numeric data. One way to work out distance is the SSD (sum-squared distance)
- If you have two records a and b, then the distance between them can be given by:

$$\sum_{i=1}^{F-1} (v(a,i) - v(b,i))^2$$

- Just add up the squared differences of the fields, omitting the class field
- Why don't I take the square root of this?

Scaling, Data Normalisation and Discretization

- So now… we will learn these things:
- Ways to do **normalisation** and **scaling** (which may be necessary or sensible, depending on how the data were generated, and on what DM/ML methods you want to use)
- Ways to do **discretization** (for turning numeric fields into categorical fields)

Data Normalisation

- You have data about word-counts for specific words on web pages. You want to predict whether the page is about **sport** or **business**.

"winner"	"game"	"team"	"sales"	"run"	Page content
16	22	81	75	10	business
12	14	44	16	12	business
4	7	20	0	2	sport
2	3	7	6	1	???

- What category would 1-NN predict?

Data Normalisation

- The closest record to the 4th is the 3rd, so it would be predicted to be sport. But this is probably wrong. Why?

"winner"	"game"	"team"	"sales"	"run"	Page content
16	22	81	75	10	business
12	14	44	16	12	business
4	7	20	0	2	sport
2	3	7	6	1	???

Data Normalisation

- The closest record to the 4th is the 3rd, so it would be predicted to be sport. But this is probably wrong. Why?

"winner"	"game"	"team"	"sales"	"run"	Page content
16	22	81	75	10	business
12	14	44	16	12	business
4	7	20	0	2	sport
2	3	7	6	1	???

- Could we pre-process the data to help ML?

(1) Instance Normalisation

- In these data, the sensible thing to do is transform each record so that it is normalised by the total number of words. This makes them more comparable, each providing a "fingerprint" in terms of the relative proportions of the probe words.
- Sometimes this is necessary, sometimes it is useful – it takes common sense.

(1) Instance normalised version

- Scale each row so the numeric fields add up to 1

"winner"	"game"	"team"	"sales"	"run"	Page content
0.0784	0.1078	0.397	0.368	0.049	business
0.12	0.14	0.45	0.16	0.12	business
0.12	0.21	0.61	0	0.061	sport
0.105	0.158	0.368	0.316	0.053	???

- 1-NN now shows that record 4 is closer to 1 and 2

(1) Instance Normalisation

- Using the notation in slide 5 – given a dataset of *N* numeric fields, how would you represent instance normalization of record *r* ?

(2) Attribute Min-Max normalisation

- Min-max is attribute normalisation, done separately for each attribute (column)
- For each numeric attribute, scale it so that each value is in a specific range [a, b]. E.g. usually [0,1], sometimes [-1, 1], etc]
- To scale from 0-1

$$v(r,i) \text{ becomes } \frac{v(r,i) - \min(i)}{\text{range}(i)}$$

- min(*i*) is the smallest value of attribute i in the dataset; max(*i*) is the largest; and range(*i*) is max(i) – min(i)

(2) When min-max normalisation might be useful

	Height (mm)	Weight (kg)	Strength	Game score
1	1856	75	95	70
2	1502	56	101	82
3	1904	86	112	90
4	1775	61	110	90
5	1901	81	92	??

- Which record is closest to record 5?
- Is this a good predictor for game score?
- Which attribute is most likely to predict the score?
- Can min-max normalisation help?

(2) When min-max normalisation might be useful

	Height (mm)	Weight (kg)	Strength (kg)	Game score
1	0.88	0.63	0.15	70
2				82
3	1.00	0.63	1.0	90
4				90
5	0.99	0.83	0.0	??

- Which record is closest to record 5?
- Will this be a good predictor for game mark?
- Which attribute is most likely to be important for predicting mark?
- *How does min-max normalisation help?*

Discretization

	Height (mm)	Weight (kg)	Strength	Game score
1	1856	75	95	70
2	1502	56	101	82
3	1904	86	112	90
4	1775	61	110	90
5	1901	81	92	??

- We can't run the ID3 Decision Tree classifier on this data

Discretization

	Height	Weight	Strength	Game score
1	tall	heavy	medium	low
2	short	light	medium	high
3	tall	heavy	strong	high
4	medium	medium	strong	high
5	tall	heavy	medium	medium

- We can run the ID3 decision tree algorithm on **this** data

Discretization

- Discretization is simply a process that converts a numerical field into a class field.
- How might you do this?

Binning: Equal Width Binning (EWB)

- EWB(5), for example, means you discretize into 5 values, where each value has equal `width'.
- If attribute values range from 0 to 100, then, then each bin has width 20. In the converted dataset, we can just label this bins 1,2,…,5, or we can use appropriate linguistic terms, such as:

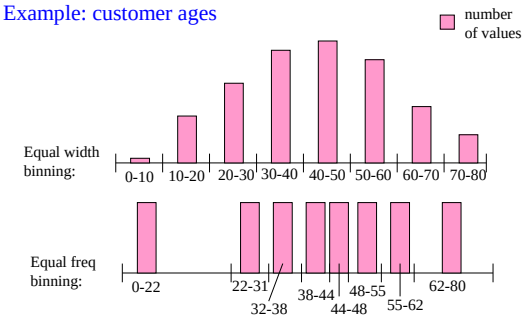
  {very low, low, medium, high, very high}

Binning: Equal Frequency Binning (EFB)

- EFB(3), for example, means you discretize into 3 values, where each bin covers (roughly) the same amount of data.
- So some bins are wider than others but all the same height

Binning

Example: customer ages



Contrived Example for EWB(2)

Height (mm)	Weight (kg)	Strength (kg)	Game score
1856	75	95	70
1502	56	101	82
1904	86	112	90
1775	61	110	90
1901	81	92	??

Height	Weight	Strength	Game score
high	high	low	low
low	high	low	low
high	high	high	high
high	low	high	high
high	high	low	low

- Discretized, 2 equal width bins

Other discretization methods

- There are others, which almost all differ from EWB and EFB in one respect: they use class value information to find bins that make good sense with regard to classification.

Get to know the data

- Use simple visualisation tools e.g.
  - histograms for categorical data
  - graphs for numeric data – any obvious outliers?
  - scatter plots to reveal dependencies
  - Inspect a sample from a very large dataset
- Data is often messy
  - Incomplete
  - Incorrect
- Visualisation can pick up problems that may affect the results

Inaccurate values

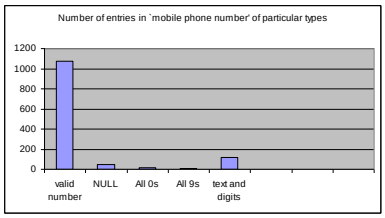
- Data was collected for other purposes
- Comes from several places
- Typographical errors in nominal values
  - Check for consistency
- Typographical errors in numeric attributes
  - Check for outliers (extreme values)
- Deliberate errors
  - E.g. wrong phone numbers, postcodes
- Duplicates, out-of-of-date…

Missing values

- Often, a dataset has missing values for some fields in some records. E.g. a certain blood test was not taken for some patients. A questionnaire response for some question was unreadable, etc…

Missing or Default Values

- Missing values can have different meanings
- E.g. Suppose the histogram of value types in mobile phone no. field is:



What does NULL mean?

A., This record is of someone who does not have a mobile phone?

B. This record is of someone who has a mobile phone, but chose not to supply the number?

C. This record is of someone who has a mobile phone, but who forgot to supply the number, or it was hard to decipher and recorded as NULL?

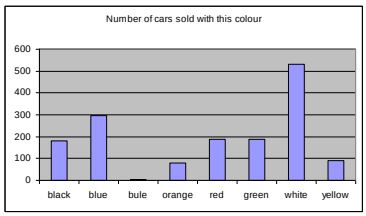
Maybe some are of type A and some are of type B and some are of type C. For some applications/analyses, we may wish to know the breakdown into types.

What about the All zero and All nine entries? Precisely the same can be said of them. Or, perhaps the protocols for recording the entries indicated NULL for type A, 0000000 for type B and 9999999 for type C.

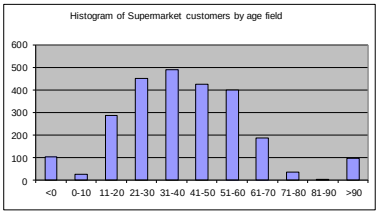
The above relate to a quite simple form of semantic complexity – but what if someone uses this DB to estimate the proportion of people who have never had a mobile phone?

Cleaning via basic data analysis

- Data Profiling: examine the instances to see how the attributes vary. E.g. Automatically generate a histogram of values for that attribute.
- How does the histogram help us in finding problems in this case?



What problems does this analysis alert us to?



Take away

- Prepare data by scaling it and normalising it, and possibly dividing numeric data into categories (discretization)
- Before you try to use a learning mechanism
  - Look at the data critically
  - Consider unusual data
  - And missing data