Week 3 Video 4

Automated Feature Generation

Automated Feature Selection

Automated Feature Generation

- The creation of new data features in an automated fashion from existing data features

Multiplicative Interactions

- You have variables A and B
- New variable C = A * B

- Do this for all possible variables

Multiplicative Interactions

- A well-known way to create new features
- Rich history in statistics and statistical analysis

Less Common Variant

- A/B
- You have to decide what to do when B=0

Function Transformations

- X^2
- Sqrt(X)
- Ln(X)

Automated Threshold Selection

- Turn a numerical variable into a binary
- Try to find the cut-off point that maximizes your dependent variable
  - J48 does something very much like this
  - You can hack this in the Excel Equation solver or do this using code

Which raises the question

- Why would you want to do automated feature selection, anyways?

- Won't a lot of algorithms do this for you?

A lot of algorithms will

- But doing some automated feature generation before running a conservative algorithm like Linear Regression or Logistic Regression
- Can provide an option that is less conservative than just running a conservative algorithm
- But which is more conservative than algorithms that look for a broad range of functional forms

Also

- Binarizing numerical variables by finding thresholds and running linear regression

- Won't find the same models as J48

- A lot of other differences between the approaches

Another type of automated feature generation

- Automatically distilling features out of raw/incomprehensible data
  - Different than code that just distills well-known data, this approach actually tries to discover what the features should be

Emerging method

- Auto-encoders

- Uses neural network to find structure in variables in an unsupervised fashion

- Just starting to be used in EDM – use by Bosch and Paquette (2018) in automatic generation of features for affect detection

Automated Feature Selection

- The process of selecting features prior to running an algorithm

First, a warning

- Doing automated feature selection on your whole data set prior to building models
- Raises the chance of over-fitting and getting better numbers, even if you use cross-validation when building models
- You can control for this by
  - Holding out a test set
  - Obtaining another test set later

Correlation Filtering

- Throw out variables that are too closely correlated to each other
- But which one do you throw out?
- An arbitrary decision, and sometimes the better variables get filtered
  (cf. Sao Pedro et al., 2012)

Fast Correlation-Based Filtering (Yu & Liu, 2005)

- Find the correlation between each pair of features
  - Or other measure of relatedness – Yu & Liu use entropy despite the name
  - I like correlation personally
- Sort the features by their correlation to the predicted variable

Fast Correlation-Based Filtering (Yu & Liu, 2005)

- Take the best feature
  - E.g. the feature most correlated to the predicted variable
- Save the best feature
- Throw out all other features that are too highly correlated to that best feature
- Take all other features, and repeat the process

Fast Correlation-Based Filtering (Yu & Liu, 2005)

- Gives you a set of variables that are not too highly correlated to each other, but are well correlated to the predicted variable

Example

	A	B	C	D	E	F	Predicted
A		.6	.5	.4	.3	.7	.65
B			.8	.7	.6	.5	.68
C				.2	.3	.4	.62
D					.8	.1	.54
E						.3	.32
F							.58

Cutoff = .65

	A	B	C	D	E	F	Predicted
A		.6	.5	.4	.3	.7	.65
B			.8	.7	.6	.5	.68
C				.2	.3	.4	.62
D					.8	.1	.54
E						.3	.32
F							.58

Find and Save the Best

	A	B	C	D	E	F	Predicted
A		.6	.5	.4	.3	.7	.65
B			.8	.7	.6	.5	.68
C				.2	.3	.4	.62
D					.8	.1	.54
E						.3	.32
F							.58

Delete too-correlated variables

	A	B	C	D	E	F	Predicted
A		.6	.5	.4	.3	.7	.65
B			.8	.7	.6	.5	.68
C				.2	.3	.4	.62
D					.8	.1	.54
E						.3	.32
F							.58

Save the best remaining

	A	B	C	D	E	F	Predicted
A		.6	.5	.4	.3	.7	.65
B			.8	.7	.6	.5	.68
C				.2	.3	.4	.62
D					.8	.1	.54
E						.3	.32
F							.58

Delete too-correlated variables

	A	B	C	D	E	F	Predicted
A		.6	.5	.4	.3	.2	.65
B			.8	.7	.6	.5	.68
C				.2	.3	.4	.62
D					.8	.1	.54
E						.3	.32
F							.58

No remaining over threshold

	A	B	C	D	E	F	Predicted
A		.6	.5	.4	.3	.2	.65
B			.8	.7	.6	.5	.68
C				.2	.3	.4	.62
D					.8	.1	.54
E						.3	.32
F							.58

Note

- The set of features was the best set that was not too highly-correlated

In-Video Quiz: What Variables will be kept? (Cutoff = 0.65)

	G	H	I	J	K	L	Predicted
G		.7	.8	.8	.4	.3	.72
H			.8	.7	.6	.5	.38
I				.8	.3	.4	.82
J					.8	.1	.75
K						.5	.65
L							.42

A) I, K, L

B) I, K

C) G, K, L

D) G, H, I, J

Removing features that could have second-order effects

- Run your algorithm with each feature alone
  - E.g. if you have 50 features, run your algorithm 50 times
  - With cross-validation turned on
- Throw out all variables that are equal to or worse than chance in a single-feature model
- Reduces the scope for over-fitting
  - But also for finding genuine second-order effects

Forward Selection

- Another thing you can do is introduce an outer-loop forward selection procedure outside your algorithm

- In other words, try running your algorithm on every variable individually (using cross-validation)
- Take the best model, and keep that variable
- Now try running your algorithm using that variable and, in addition, each other variable
- Take the best model, and keep both variables
- Repeat until no variable can be added that makes the model better

Forward Selection

- This finds the best set of variables rather than finding the goodness of the best model selected out of the whole data set
- Improves performance on the current data set
  - i.e. over-fitting
  - Can lead to over-estimation of model goodness
- But may lead to better performance on a held-out test-set than a model built using all variables
  - Since a simpler, more parsimonious model emerges

You may be asking

- Shouldn't you let your fancy algorithm pick the variables for you?

- Feature selection methods are a way of making your overall process more conservative
  - Valuable when you want to under-fit

Automated Feature Generation and Selection

- Ways to adjust the degree of conservatism of your overall approach
- Can be useful things to try at the margins
- Won't turn junk into a beautiful model

Next Lecture

- Knowledge Engineering