Embedded Methods

Matthew Miers, SooJung Lee, DaeHo Kim

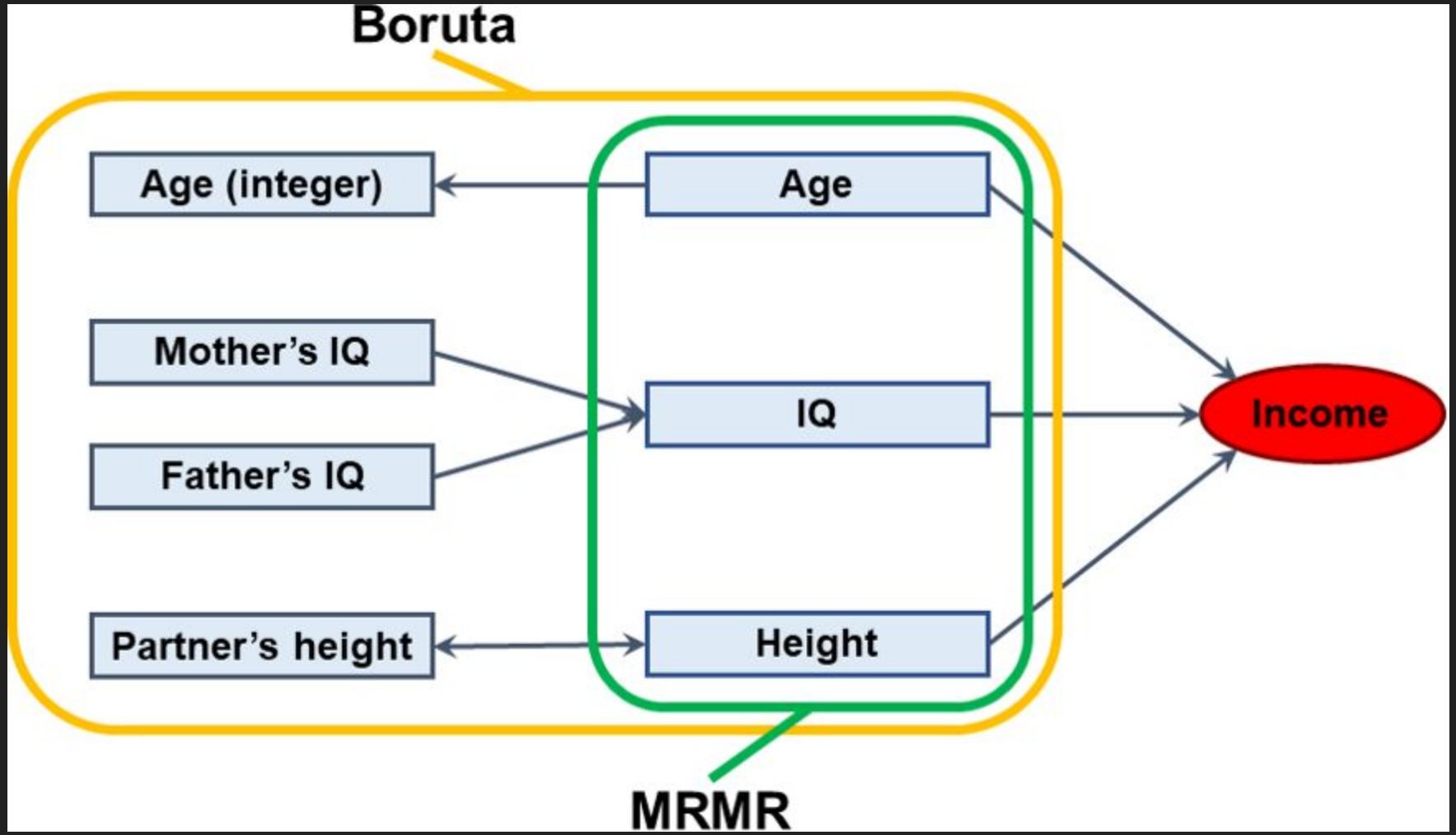
What are feature selection methods?

For many machine learning applications, often times there is a wealth of the number of features given.

While more features is typically a good thing, it can lead to overfitting, as well as an increase in the computational power needed to run the model, or even just be random noise or be basically the same as some other features

The concept of feature selection is to reduce some features to the ones that reduce noise, but still provide valuable prediction results, and there are three main types

- Filter methods
- Wrapper methods
- Embedded methods


Boruta
Age (integer)
Mother's IQ
Father's IQ
Partner's height
Age
IQ
Height
Income
MRMR

What are embedded methods?

Builds off of the ideas presented in wrapper methods, but aims to include the variable selection as a part of the training process

Some types of embedded selection:

- Greedy selection
- mRMR
- Classifier weight rankings

Background Definitions

Mutual Information (MI) - a measure of dependency between two variables

$$I(Y,X) = H(Y) - H(Y|X)$$

Entropy and conditional entropy

$$H(Y) = -\sum_{y} p(y) log(p(y))$$

$$H(Y|X) = -\sum_{x}\sum_{y} p(x,y) log(p(y|x))$$

Greedy search

Goal: choosing a feature will maximize the MI between the feature and the class output with a negative weight given to features that have high MI with already selected features

$$I(Y,f) - \beta \sum_{s \in S} I(f;s)$$

Y - the output
f - the current selected feature
s - a feature in the already selected subset (S)
β - the weight of the MI and the current feature set (S)

Greedy search simply

- Select the most ‘informative’ feature based on the output
- Add that feature to the selected feature set
- For each additional feature choose one that is a balance of being ‘informative’ to the output and not ‘informative’ to already selected features
- Stop when the value falls below a chosen threshold

mRMR - min Redundancy Max Relevance

Goal: Much the same as the greedy selection (features that will maximize the MI with the class output minimize MI with already selected features) but changes the redundancy weight value

$$I(x_j; C) - \frac{1}{m-1} \sum_{x_l \in S_{m-1}} I(x_j; x_l)$$

$I(x_j;C)$ - MI between feature and classifier
m - number of features
S_{m-1} - set of features (excluding current feature)

Understanding the formula

$$score_i(f) = \frac{relevance(f \mid target)}{redundancy(f \mid features\ selected\ until\ i-1)}$$

$$score_i(f) = \frac{F(f, target)}{\sum_{s \in features\ selected\ until\ i-1} |corr(f, s)| \ / \ (i-1)}$$

F = F-Statistics
f = feature
corr = Pearson Correlation
i = i-th iteration
s = Subset

mRMR - Many flavors

MRMR Selection Schemes

Type	Acronym	Full Name	Formula
Discrete	MID	Mutual information difference	$\max_{i \in \Omega_S}[I(i,h) - \frac{1}{\lvert S \rvert} \sum_{j \in S} I(i,j)]$
	MIQ	Mutual information quotient	$\max_{i \in \Omega_S}\{I(i,h) / [\frac{1}{\lvert S \rvert} \sum_{j \in S} I(i,j)]\}$
Continuous	FCD	F-test correlation difference	$\max_{i \in \Omega_S}[F(i,h) - \frac{1}{\lvert S \rvert} \sum_{j \in S} \lvert c(i,j) \rvert]$
	FCQ	F-test correlation quotient	$\max_{i \in \Omega_S}\{F(i,h) / [\frac{1}{\lvert S \rvert} \sum_{j \in S} \lvert c(i,j) \rvert]\}$
	FDM	F-test distance multiplicative	$\max_{i \in \Omega_S}[F(i,h) \cdot \frac{1}{\lvert S \rvert} \sum_{j \in S} d(i,j)]$
	FSQ	F-test similarity quotient	$\max_{i \in \Omega_S}\{F(i,h) / [\frac{1}{\lvert S \rvert} \sum_{j \in S} \frac{1}{d(i,j)}]\}$

Ding & Peng, CSB2003, JBCB2005

Classifier weight rankings

Goal: To build a vector (w) that has the same length as the number of features. The values for each feature should a large positive number if the feature is strongly correlated with the positive classification group (ex has cancer) and should have a large negative number if the feature is strongly correlated with the negative classification group.

$$w_j = \frac{\mu_j(+) - \mu_j(-)}{\sigma_j(+) + \sigma_j(-)}$$

$\mu_j(+)$ - mean of the input values for feature j that belong to the positive classification group

$\mu_j(-)$ - mean of the input values for feature j that belong to the negative classification group

$\sigma_j(+)$ - std dev of the input values for feature j that belong to the positive classification group

$\sigma_j(-)$ - std dev of the input values for feature j that belong to the negative classification group

w_j - the weight for the element corresponding to feature j in the weight vector

Classifier weight rankings (continued)

To use for prediction (weighted voting scheme): Use the following decision function. If the value is positive, then predict the (+) group if the value is negative predict the (-) group

$$D(\mathbf{x}) = \mathbf{w} \cdot (\mathbf{x} - \mu)$$

D(x) - the decision function for sample x
w - the function weight vector calculated in the previous step
x - the sample's function vector
$\mu = (\mu_j(+) + \mu_j(-)) / 2$ using values from the previous step

Classifier weight rankings (continued)

To use for feature selection: Calculate values DJ(i) using the below formula for each feature (sensitivity analysis). A simpler method involves pruning features with low magnitudes in the w vector

$$DJ(i) = (1/2)\frac{\partial^2 J}{\partial w_i^2}(Dw_i)^2$$

Questions?