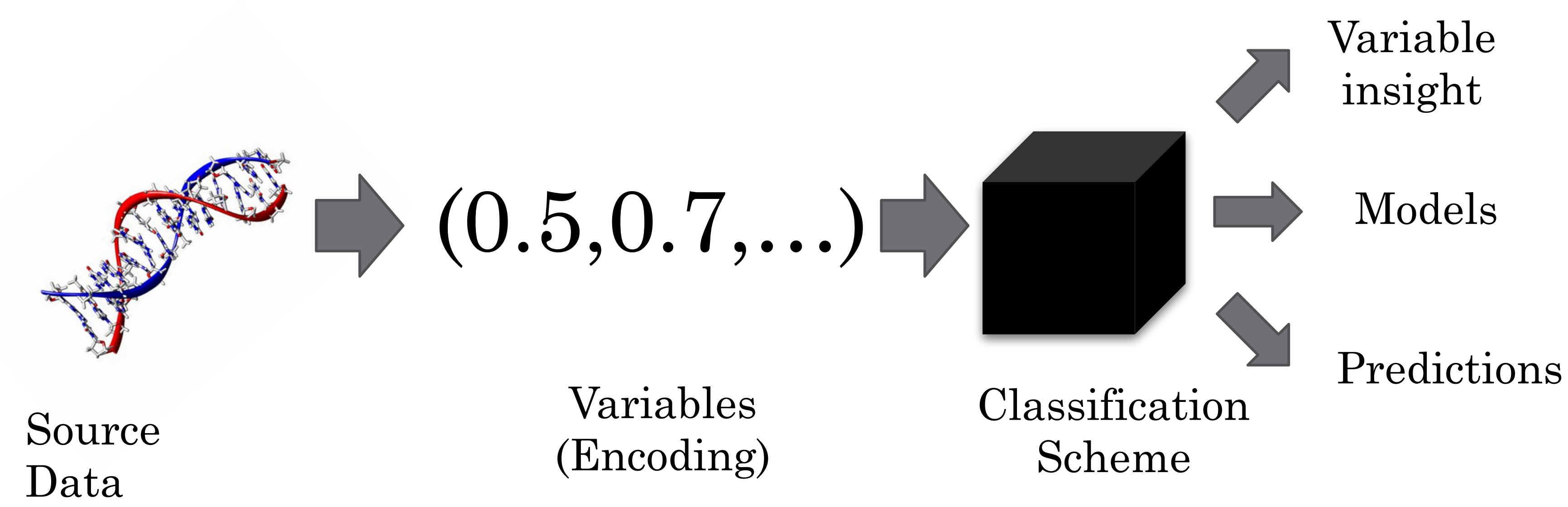
Feature Selection and Related Approaches

CSCI 4181 / 6802 Module 1-FEAT

The big picture



$(0.5,0.7,\ldots)$

Variable insight

Models

Predictions

Source Data

Variables (Encoding)

Classification Scheme

Module 1 Lecture 1
+ today

Overview

1. So many features!!!
2. Feature selection – choosing the best subset of variables
3. Feature extraction – merging components of variables

```
gagggaggcacaggcgcttgctttgcaggagtcagctctgccttcctcggctggagtgtgggtggcttggtgagccggtggtcaggaattctctctcctccttgcaattttcctt
ctgggagcacgccaagatgtcccttgtgactgtccccttctaccagaagagacataggcacttcgaccagtcctaccgtaatattcaaacacggtacctgctggacgaatatgcg
aaagcgagcttccacccaggcatcttcccagaagtccttgagtcagcggtcgtcttcacagagagcctccagccagacgtccctgggaggaaccatctgcagggtctgtgcgaag
gagcacgcaggaagatgaggagcaggagaacagaagcaggtaccagtccctggtggccgcctatggtgaggccaagcgacagcgcttcctcagcgagctggcccacttggaggag
ccacctggcacgctcccaggcccgcgacaagctggacaaatacgccattcagcagatgatggaggacaagctggcctgggagagacacacatttgaagagcggataagcagggct
gatcctggtgcggctgcgatcccacaccgtctgggagaggatgtctgtgaaactctgcttcaccgtgcaaggatttcccacgcccgtggtgcagtggtacaaagatggcagtctg
ccaggcggctgaaccgggaaagtacaggattgagagcaactatggcgtacacacactggagatcaacagggcagactttgacgacactgcgacatactcagcagtggccaccaat
cggacaagtgtccaccaacgcggcggtggtggtgagaaggttccggggagacgaggaaccattccgttcggtgggactcccgattggattgcccctgtcatcgatgattccgtac
cttcgacgtccagtttttggagaagtttggggtcaccttcaggagggaaggcgagacggtcactctcaagtgcaccatgctggtgacgccggacctgaagcgggtgcagccgcgc
gtggtaccgcgatgacgtgctgttgaaagagtccaagtggacgaagatgttctttggagaaggccaggcctccctgtccttcagccacctgcacaaggacgacgagggcctgtac
gcgcatcgtgtctcggggcggcgtcagcgaccacagcgccttcctgtttgtcagagatgctgacccgctggtcacagggggcccccggtgcacccatggacttgcagtgccacgac
ccgggactacgtcatcgtgacctggaagccgcccaacaccaccactgagagccccgtcatgggctattttgtggaccgatgtgaagtaggaacgaataattgggtgcagtgcaat
accggtgaaaatctgcaaatacccggtcacaggctttttgaaggaaggtcttacatattccgagtgagggcagtgaacagtgcgggcatcagccgaccctccagggtctctgat
ggctgcacttgaccccttggacctcagaaggttacaagccgttcatttggagggagagaaggagattgccatttatcaggatgaccttgaaggtgacgcccaggttccagggcct
cggtgtgcacgcttccgagatcagcagaaactatgtcgtcctcagctgggagccacccactccccgtggcaaggacccgctcatgtacttcattgagaagtcggtggtggggagc
ctggcagagagtcaacgcccagacggctgtgagatccccgagatatgccgtgtttgacctcatggaagggaagtcttatgtgttccgagtgctgtcagcaaaccggcatggcctg
accttcggagataacgtccccattcaggcccaggatgtgaccgttgtcccttctgctccgggtcgggttcttgcttcccgaaacaccaagacgtcggtggtggtgcagtgggac
caagcatgaggaggacctgctgggctactacgtggactgctgtgtggccggaaccaacctctgggagccctgcaaccacaagcccattggatacaacaggttcgtggtgcacggc
cacgggagagcagtacatcttccgagtcaaggcggtcaatgctgtggggatgagtgaaaattcccaggaatcagacgtcataaaagtgcaggccgcactcaccgtcccgtcccat
tgggattacgctcctcaactgtgacggccactccatgaccctcggctggaaggtcccgaaattcagtggtggctcgcccatcctgggctactacctggacaagcgtgaagttcac
aaactggcacgaggtcaattcctcacccagcaaaccgacaatcctaacggtggacggcttgacggaaggctcactctacgagttcaaaatcgccgccgtcaacctggccggcatc
gccctcagatcccagtgagcacttcaagtgtgaggcctggaccatgccggagcccggtcctgcctacgacttgacgttctgtgaggtcagggacacgtccttggtcatgctgtgg
ccctgtgtactccggcagcagccctgtttctggatatttcgtggacttcagggaggaggatgctggagagtggatcactgtaaatcagacgacaacagccaaccgttatttaaag
tgacctgcagcaaggtaagacctatgtcttcagggtccgggcagtcaatgcaaatggcgtggggaagccctcagacacgtcggagcctgtgctggtagaggcgagaccaggcacc
aatcagtgctggtgtcgatgaacaaggcaacatctatctgggcttcgactgccaggaaatgacagacgcgtctcagttcacctggtgtaaatcctacgaggagatttcagatgat
gtttaaaatcgaaaccgtggggatcactccaagctgtacttaaagaatccggataaggaggatttagggacttactccgtgtctgtaagtgatacagacggagtgtcctccagt
tctggacccagaagagctcgagcgtttgatggcattgagcaatgaaataaagaaccccacaattcctctgaaatcggaattagcttatgagatttttgataaggggcgggttcgc
gctccaggctgagcacttatcaccagatgccagctaccgatttattattaacgacagagaagtctctgacagcgagatacacagaattaaatgtgacaaagctactggcattatt
ggtgatggatcgatttagtattgaaaatgaggggacctacactgtgcagattcatgatgggaaagccaaaagtcagtcttctctagttcttattggagatgcattcaagactgtg
gaggctgagtttcaaaggaaagaatttctcaggaaacaaggccctcatttgctgagtacttgcactgggatgtcacggaagaatgtgaagttcgacttgtttgcaaggttgca
caagaaagaaaccgttttcaaatggctcaaggatgatgttctgtatgaaacggagacactgcctaacctggagaggggaatctgtgagctcctcatcccaaagttgtcaaagaag
cggtgaatacaaggcaaccttgaaagatgacagaggccaagatgtgtccatccttgaaatagctggcaaagtgtatgatgatatgattttggcaatgagtagagtctgtgggaaa
ttcgccactgaaggtactctgcaccccagaaggaatacgacttcagtgtttcatgaagtattttacagacgaaatgaaagtgaactggtgtcacaaagatgctaagatctcatcc
gcatatgagaatcgggggagtgaagagatggcttggctgcagatatgtgagccgactgagaaggataaaggaaaatacacttttgagattttcgatggcaaagacaaccatcaa
ccttgacctgtccggacaagcttttgatgaagcatttgcagaattccagcaattcaaagctgctgcttttgcagagaagaatcgtggcaggttgatcggcggcttgcctgacgtg
catcatggaagggaagaccttgaatctgacctgcacggtgtttggaaaccctgaccccgaagtgatttggttcaagaacgaccaggacatccagctcagcgagcacttctcggtg
gagcaggccaagtacgtcagcatgaccatcaaaggcgtgacctccgaggactcgggcaagtacagcatcaacatcaagaataagtatggcgggagaagatcgacgtgacagtg
gtacaaaca [illegible] tttcctagcctggag
aaatatgc [illegible] gtgcttttgatttt
ggtgatga [illegible] agagagtgggttggc
cacactagcattttcacgggtgtgggcacatgggtgtggcacctggacgtgtgcagcatgtggcggtctgtgtgaagccaccgtgcttctctttgggggccgcgagatctagc
gaaatcctggctgtcgaggctttgaagcatgtgttacctggttaagcttgttttctcttgctttaggcaaataaaagtttaaaatcaaaaaaaagcattgagcaatgaaataa
```

So, let's represent some DNA!

Reminder

All possible degenerate characters of length 1 to (say) 10

{ A, B, C, …, V }

{ AA, AB, …, VV }

…

{ AAAAAAAAAA, AAAAAAAAAC, …, VVVVVVVVVV }

So…

$$15^1 + 15^2 + 15^3 + 15^4 + 15^5 + 15^6 + 15^7 + 15^8 + 15^9 + 15^{10}$$

$$\cong 15^{10}$$

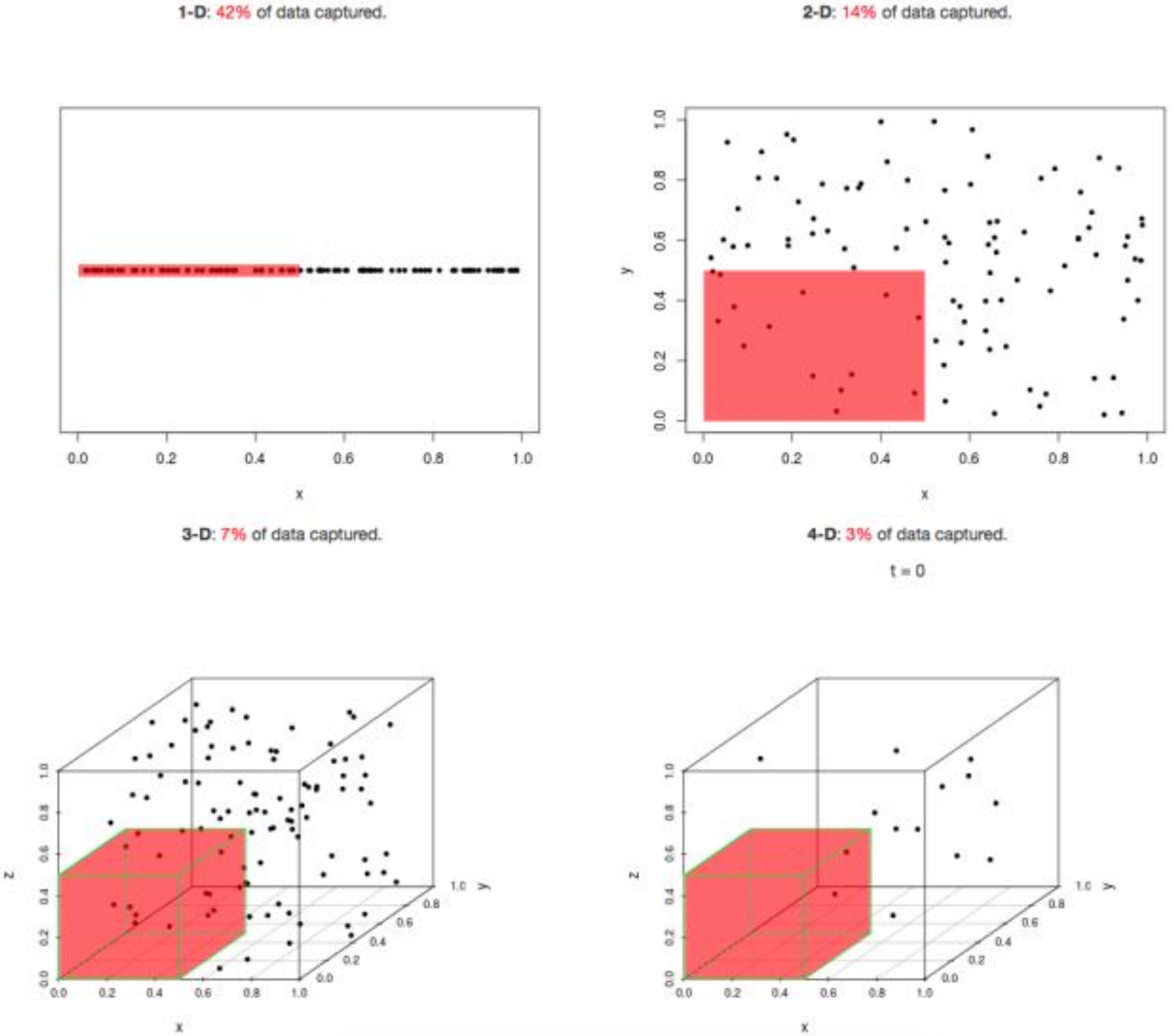
$$\cong 5.8 \times 10^{11}$$

Hmmm.

Problem?

- An excessively high-dimensional set of features / parameters is:
  - Computationally intractable
  - Fertile ground for overfitting
  - Hard to understand!

Curse of Dimensionality

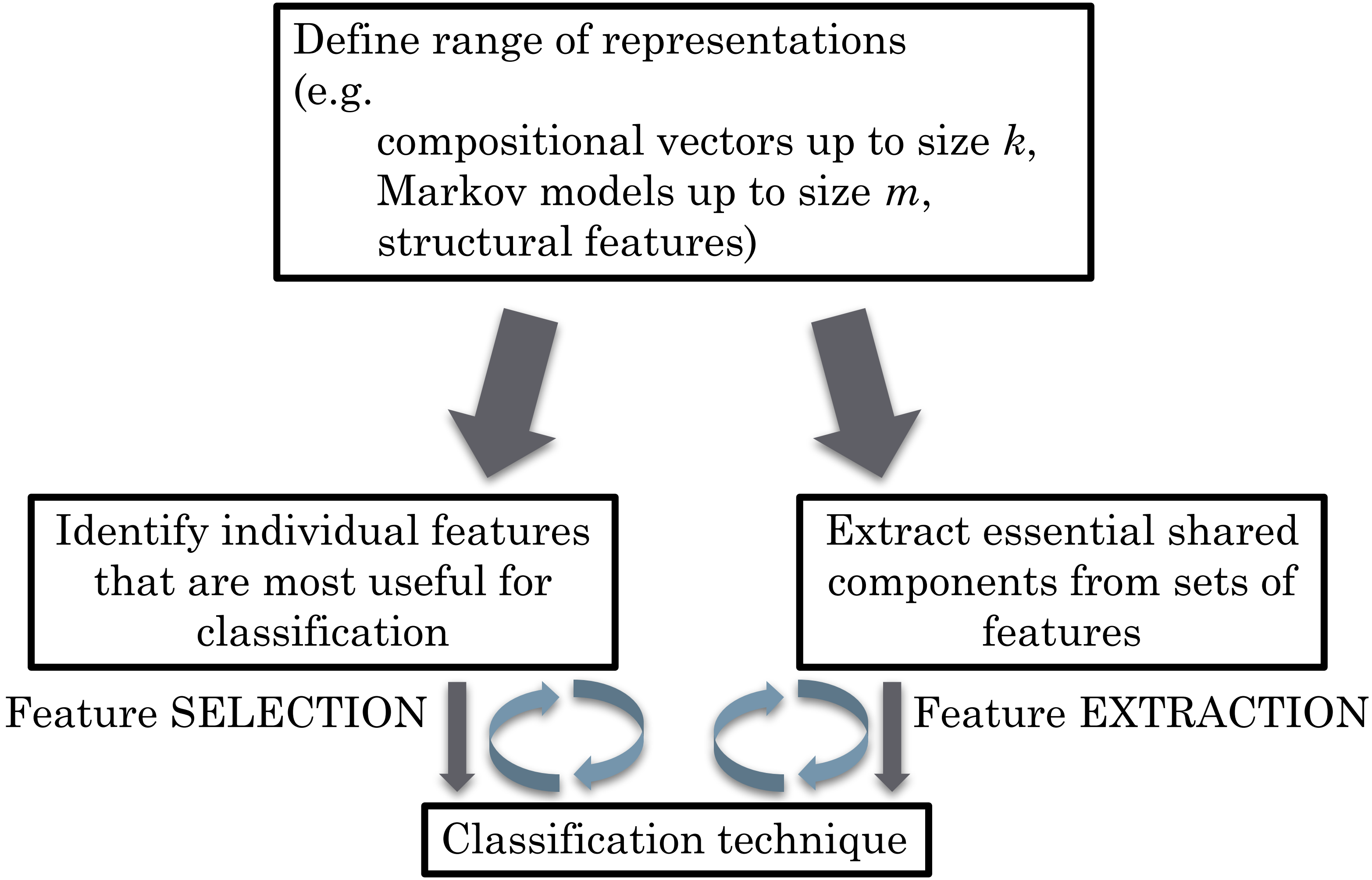


REGULARIZATION

- In basic terms, one or more procedures that puts "pressure" on a model to be simple
- Need to BALANCE accuracy vs. complexity
- Super-super general form:

$$Score = \boxed{Accuracy} - \boxed{\lambda \times Complexity}$$

Dimensionality Reduction: One ticket to model simplification



Define range of representations
(e.g.
compositional vectors up to size k,
Markov models up to size m,
structural features)

Identify individual features that are most useful for classification

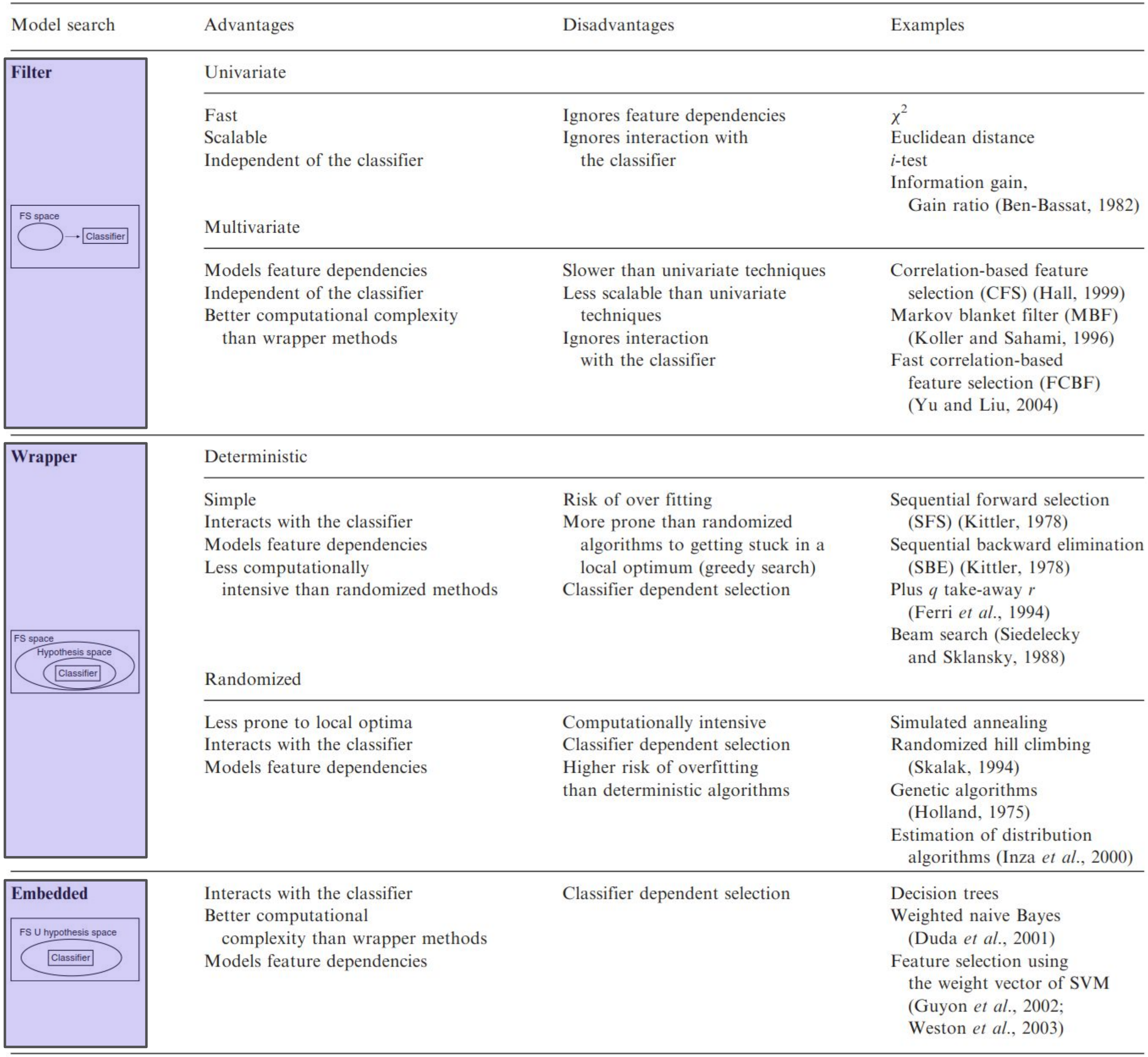
Extract essential shared components from sets of features

Feature SELECTION

Feature EXTRACTION

Classification technique

Feature Selection



Model search	Advantages	Disadvantages	Examples
Filter	Univariate		
	Fast Scalable Independent of the classifier	Ignores feature dependencies Ignores interaction with the classifier	χ^2 Euclidean distance *i*-test Information gain, Gain ratio (Ben-Bassat, 1982)
FS space Classifier	Multivariate		
	Models feature dependencies Independent of the classifier Better computational complexity than wrapper methods	Slower than univariate techniques Less scalable than univariate techniques Ignores interaction with the classifier	Correlation-based feature selection (CFS) (Hall, 1999) Markov blanket filter (MBF) (Koller and Sahami, 1996) Fast correlation-based feature selection (FCBF) (Yu and Liu, 2004)
Wrapper	Deterministic		
	Simple Interacts with the classifier Models feature dependencies Less computationally intensive than randomized methods	Risk of over fitting More prone than randomized algorithms to getting stuck in a local optimum (greedy search) Classifier dependent selection	Sequential forward selection (SFS) (Kittler, 1978) Sequential backward elimination (SBE) (Kittler, 1978) Plus *q* take-away *r* (Ferri *et al*., 1994) Beam search (Siedelecky and Sklansky, 1988)
FS space Hypothesis space Classifier	Randomized		
	Less prone to local optima Interacts with the classifier Models feature dependencies	Computationally intensive Classifier dependent selection Higher risk of overfitting than deterministic algorithms	Simulated annealing Randomized hill climbing (Skalak, 1994) Genetic algorithms (Holland, 1975) Estimation of distribution algorithms (Inza *et al*., 2000)
Embedded FS U hypothesis space Classifier	Interacts with the classifier Better computational complexity than wrapper methods Models feature dependencies	Classifier dependent selection	Decision trees Weighted naive Bayes (Duda *et al*., 2001) Feature selection using the weight vector of SVM (Guyon *et al*., 2002; Weston *et al*., 2003)

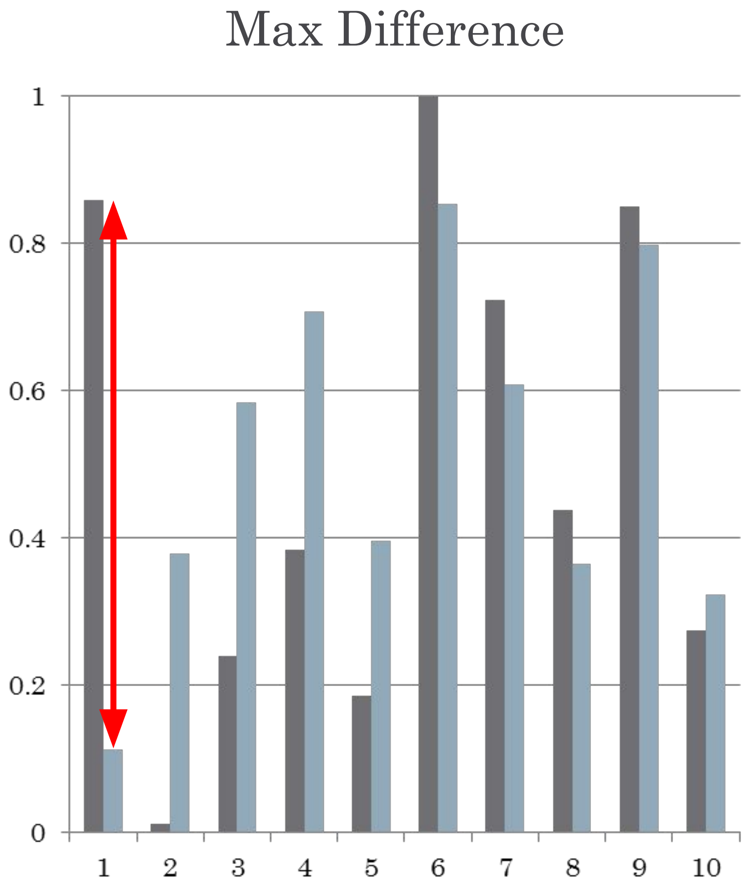
Saeys et al (2007) *Bioinformatics*

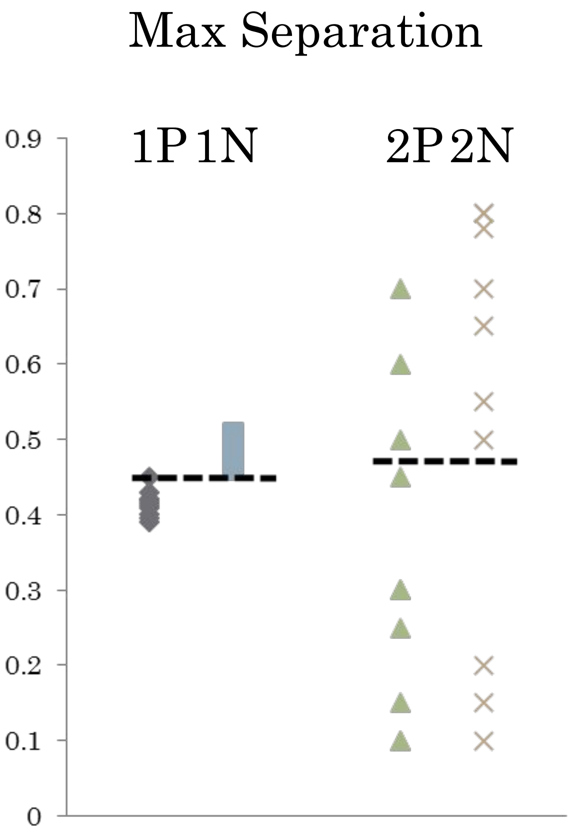
‘Filter’ Methods

Consider the individual impact of variables *before* using the classifier (typically using a simple screening criterion)

- Variable RELEVANCE
- Variable REDUNDANCY

RELEVANCE


Max Difference
1
0.8
0.6
0.4
0.2
0
1
2
3
4
5
6
7
8
9
10


Max Separation
1P1N
2P2N
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0

Mutual Information – an expression of redundancy

For two categorical variables X and Y:

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log\left(\frac{p(x,y)}{p(x)\,p(y)}\right)$$

Probability that two classes are seen together in this dataset

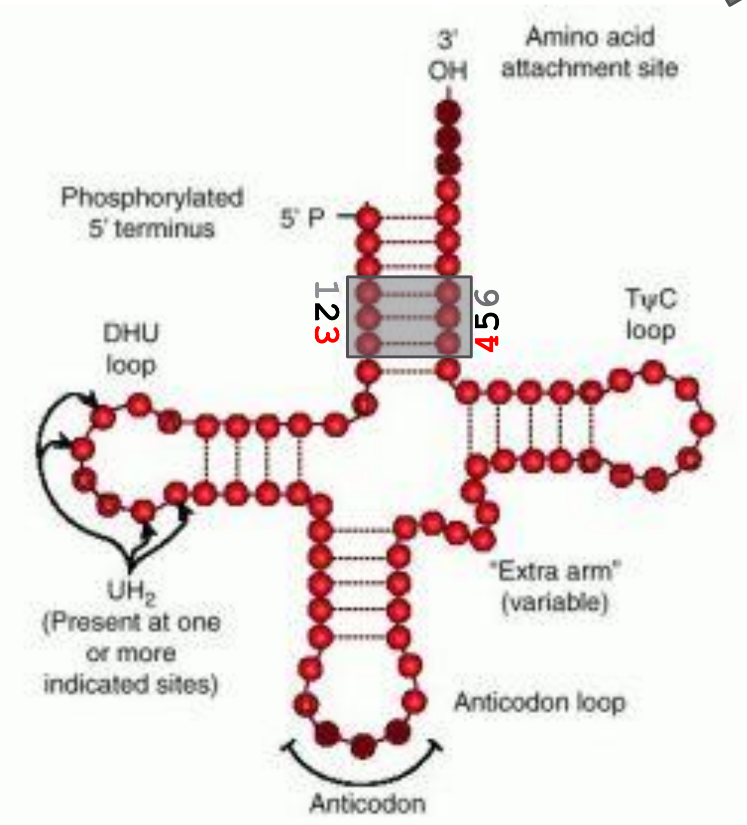
Independent probabilities of each class

How much does knowing y tell us about the value of x (or vice versa?)

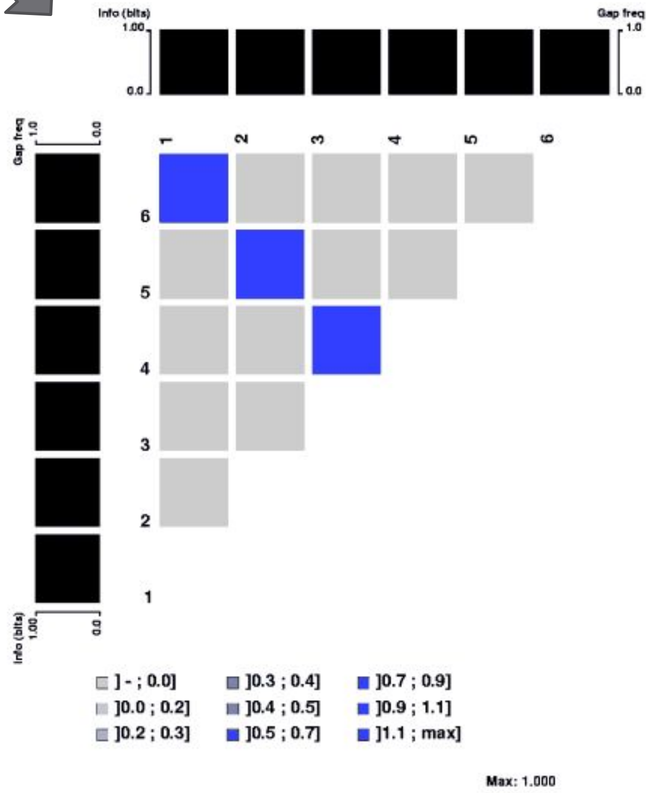
Also applicable to continuous variables (integrals)

(Equation cribbed from Wikipedia)

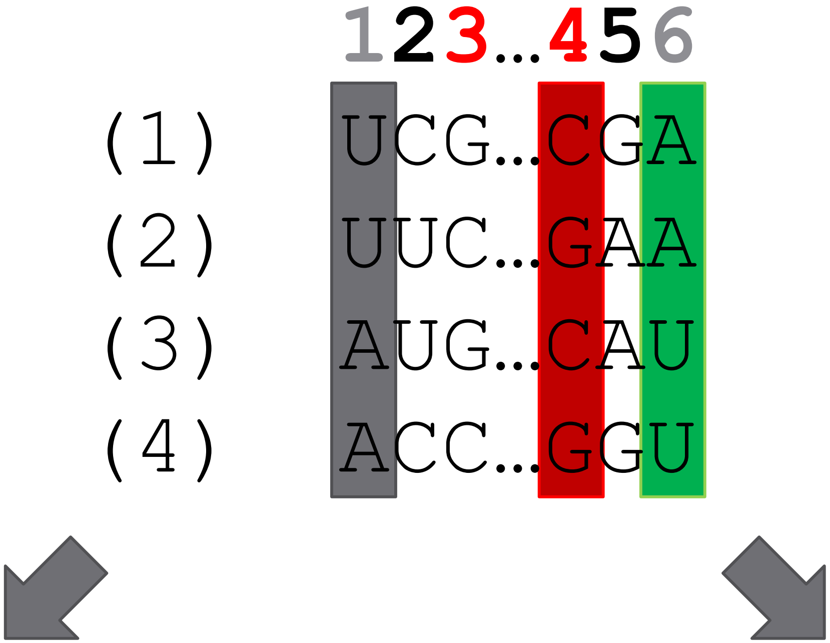
Example: tRNA Sequences


3' OH
Amino acid attachment site
Phosphorylated 5' terminus
5' P
123
456
DHU loop
TψC loop
UH$_2$ (Present at one or more indicated sites)
"Extra arm" (variable)
Anticodon loop
Anticodon

```
       123…456
(1)    UCG…CGA
(2)    UUC…GAA
(3)    AUG…CAU
(4)    ACC…GGU
```



$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log\left(\frac{p(x,y)}{p(x)\,p(y)}\right)$$



```
         123...456
(1)      UCG...CGA
(2)      UUC...GAA
(3)      AUG...CAU
(4)      ACC...GGU
```

Col 1 vs col 4

$I = 4[0.25 \times \log_2(0.25 / 0.25)]$

$= 0$

(complete independence)

Col 1 vs col 6

$I = 2[0.5 \times \log_2(0.5 / 0.25)]$

$= 1$

(complete redundancy)

Minimum Redundancy – Maximum Relevance (MRMR)

- Minimum redundancy: select variables that are largely independent, as assessed by
  - Low mutual information
  - Minimal correlation
  - Maximal Euclidean distance

- Maximum relevance: select variables that are good classifiers!

Peng H, Long F, Ding C (2005) Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1226-1238.

MRMR aims to maximize either

(relevance – redundancy)

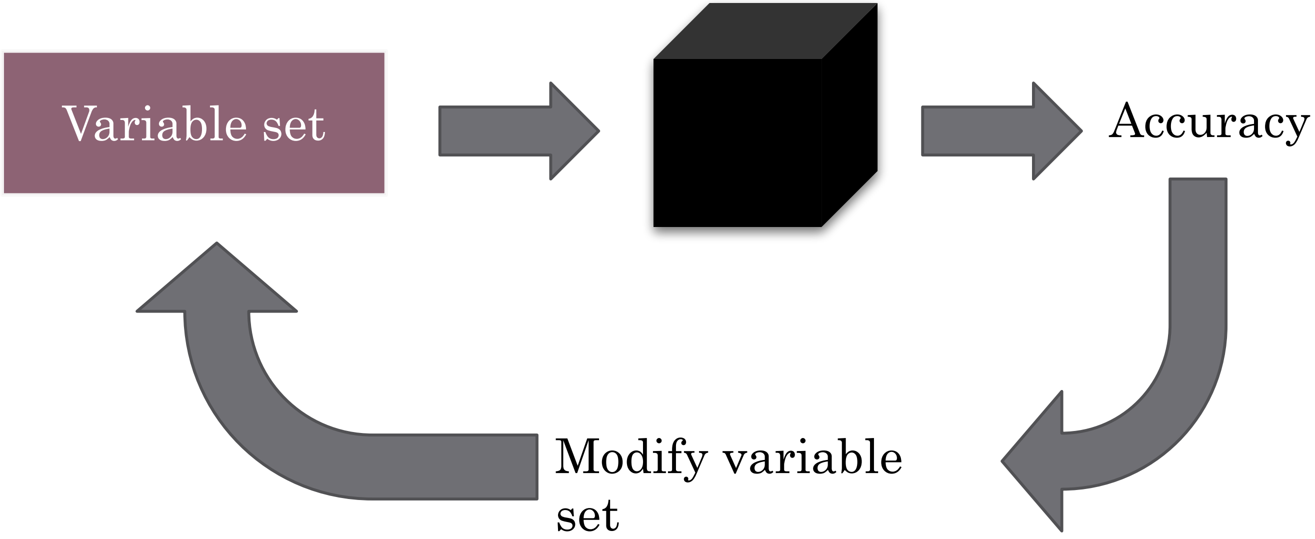
OR

(relevance / redundancy)

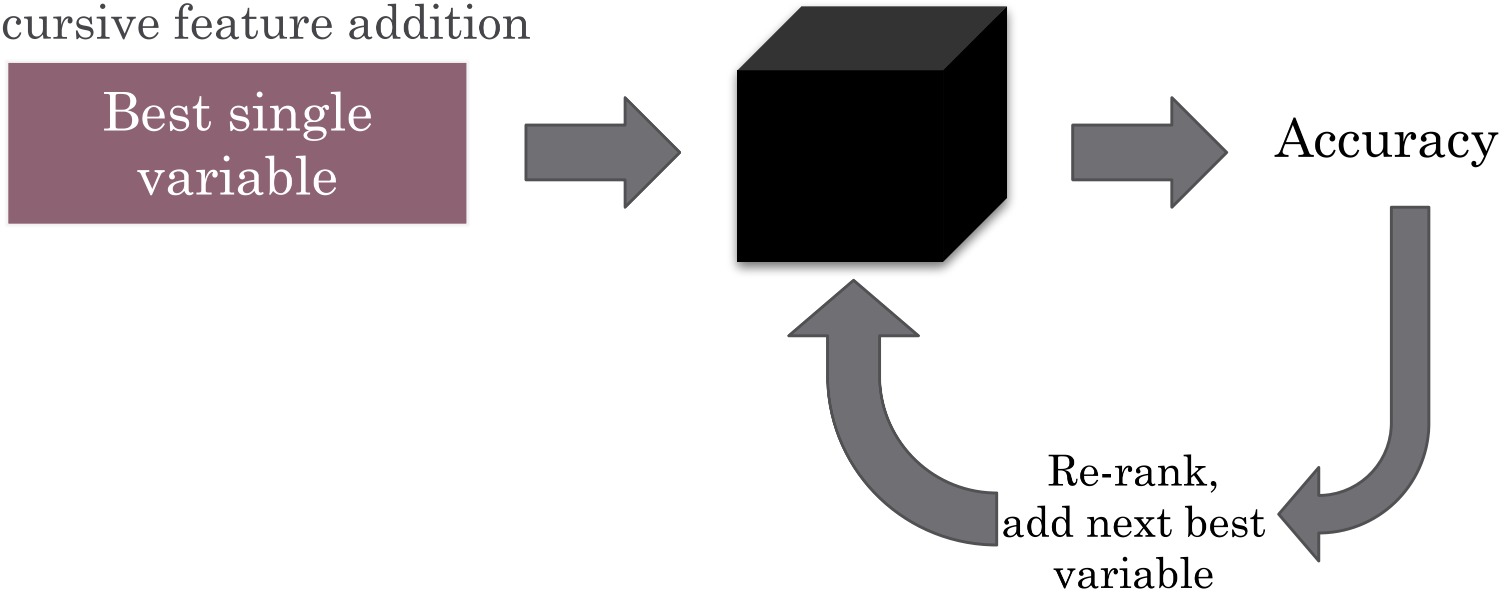
Using a greedy approach.

Wrapper Methods

Same idea as filter methods, but instead of having a quick screening process, feedback from the full classifier is used to select variables



Recursive feature addition

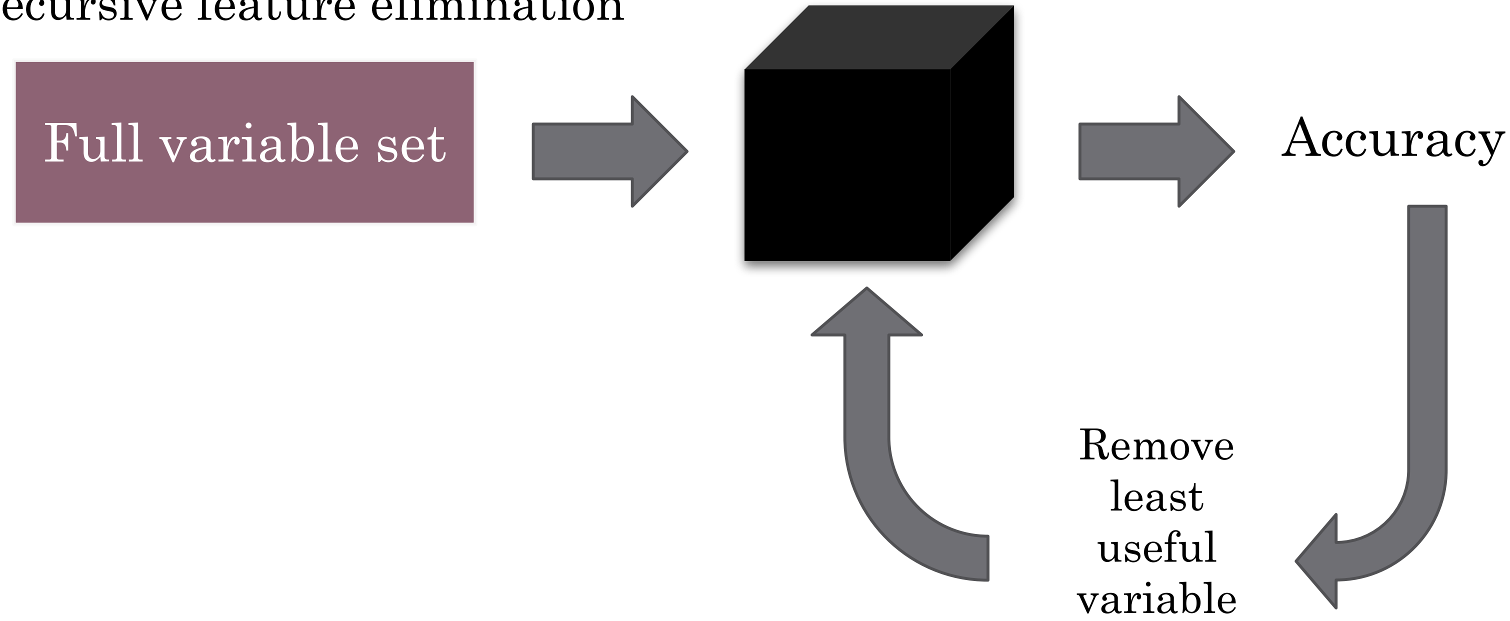


Best single variable

Accuracy

Re-rank, add next best variable

Recursive feature elimination



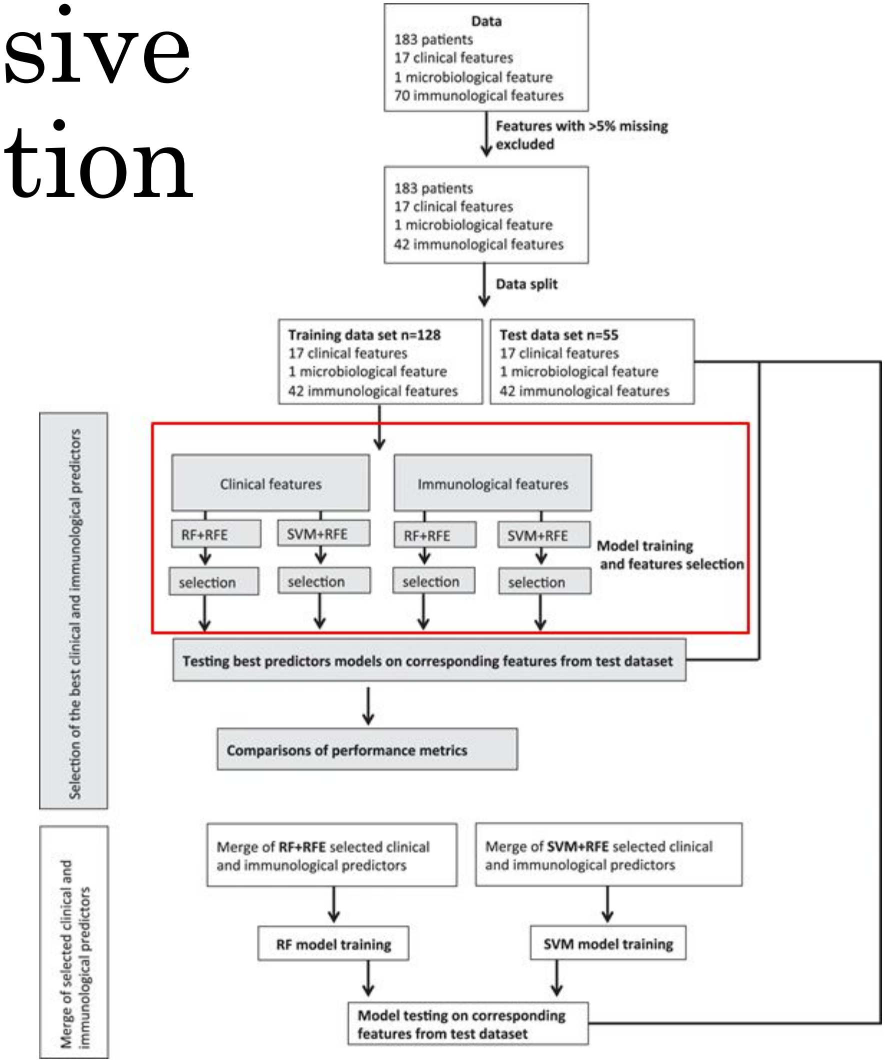
Full variable set

Accuracy

Remove least useful variable

Example: recursive feature elimination

- What factors are the best predictors of UTI?
- Try recursive feature elimination



Data
183 patients
17 clinical features
1 microbiological feature
70 immunological features

Features with >5% missing excluded

183 patients
17 clinical features
1 microbiological feature
42 immunological features

Data split

Training data set n=128
17 clinical features
1 microbiological feature
42 immunological features

Test data set n=55
17 clinical features
1 microbiological feature
42 immunological features

Selection of the best clinical and immunological predictors

Clinical features

Immunological features

RF+RFE

SVM+RFE

RF+RFE

SVM+RFE

selection

selection

selection

selection

Model training and features selection

Testing best predictors models on corresponding features from test dataset

Comparisons of performance metrics

Merge of selected clinical and immunological predictors

Merge of **RF+RFE** selected clinical and immunological predictors

Merge of **SVM+RFE** selected clinical and immunological predictors

RF model training

SVM model training

Model testing on corresponding features from test dataset

Gadalla et al. (2019) *Sci Rep*

Classifier

"Accuracy"

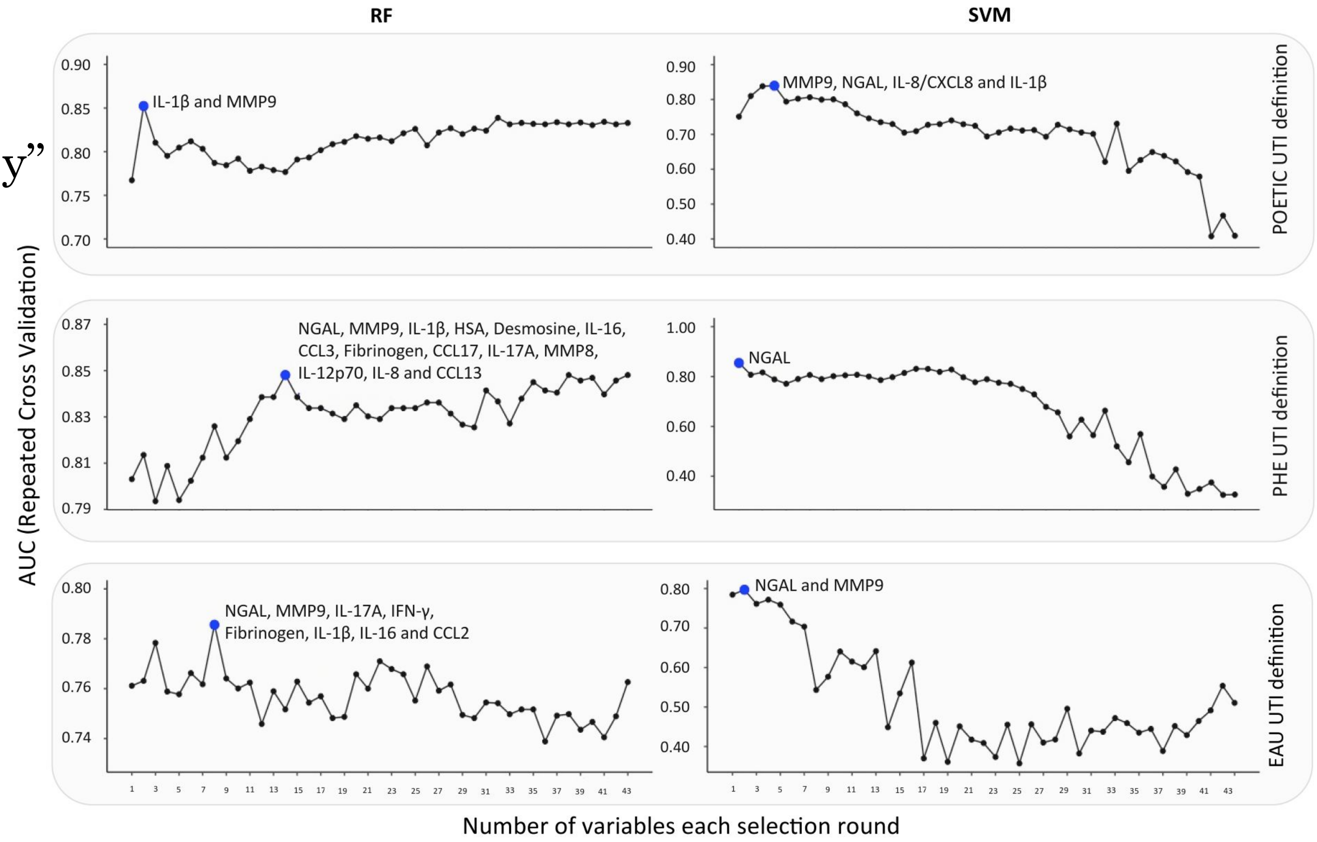


Figure S2: Feature selection among immunological markers using different UTI classification guidelines. POETIC: Point of care testing for urinary tract infection in primary care, PHE: Public Health England, EAU: European Association of Urology, AUC: Area under the ROC curve, RF: Random forest and SVM: Support vector machine

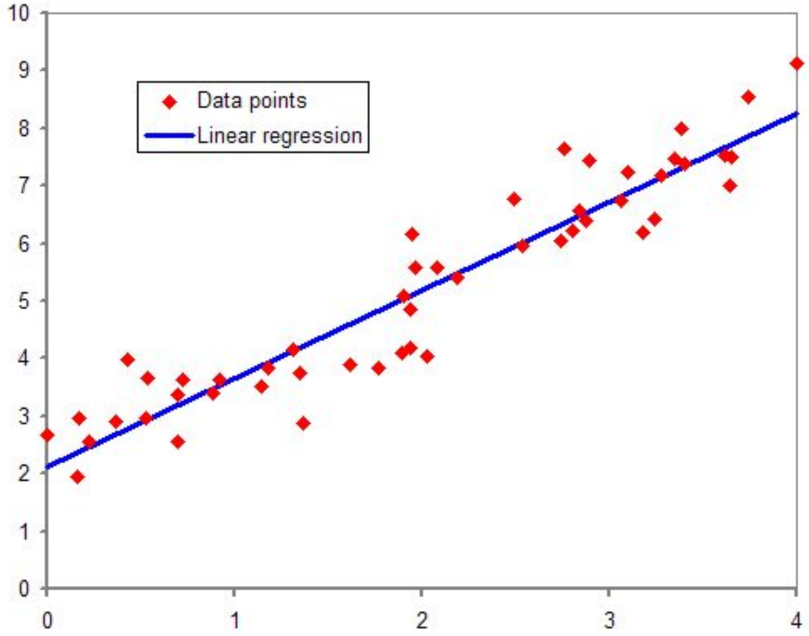
Messages:

- Small, interpretable sets – yay!
- The choice of classifier can make a BIG difference

Gadalla et al. (2019) *Sci Rep*

Embedded methods

- Optimize variable set during model training
- Tend to be faster than wrappers
- Let’s think about univariate regression:



(yes, from Wikipedia)

y = mx + b

y = mx + b + ε

Optimization:

choose m, b such that the sum of squared errors is minimized

Multiple regression

- General form:

$$\underline{\mathbf{y}} = \underline{X\boldsymbol{\beta}} + \underline{\boldsymbol{\varepsilon}},$$

Predicted value

For each predictor,
Coefficient (m) times value

For each predictor,
Error term

- So many possible predictors!
  - Plenty of opportunities to overfit
  - Spurious relationships make it hard to interpret coefficients

(yes, from Wikipedia)

Two ways to deal with this

- **LASSO:** aggressively prune variables

$$= \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \underbrace{\|y - X\beta\|_2^2}_{\text{Loss}} + \lambda \underbrace{\|\beta\|_1}_{\text{Penalty}}$$

- Linear penalty aggressively sets many coefficients to zero (equivalent to removing variables)
- Large λ: big penalty, fewer variables

https://www.analyticsvidhya.com/blog/2015/08/comprehensive-guide-regression/

Two ways to deal with this

- **Ridge regression:** penalize coefficients less aggressively

$$= \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \underbrace{\|y - X\beta\|_2^2}_{\text{Loss}} + \lambda \underbrace{\|\beta\|_2^2}_{\text{Penalty}}$$

- Squared penalty aggressively sets many coefficients to zero (equivalent to removing variables)
- Large λ: big penalty, smaller coefficients (but more non-zero variables)

https://www.analyticsvidhya.com/blog/2015/08/comprehensive-guide-regression/

So

- Do you want to keep fewer variables (hard decisions) or more variables (weak decisions?)

- You can have it all with Elastic Net!

$$\hat{\beta} \equiv \underset{\beta}{\mathbf{argmin}}(\underbrace{\|y - X\beta\|^2}_{\textbf{Regression}} + \underbrace{\lambda_2\|\beta\|^2}_{\textbf{Ridge}} + \underbrace{\lambda_1\|\beta\|_1}_{\textbf{LASSO}}).$$

- $\lambda_1 + \lambda_2 = 1$
- Large λ_1: stronger LASSO
- Large λ_2: stronger ridge

Example: predictive value of gene expression in cancer

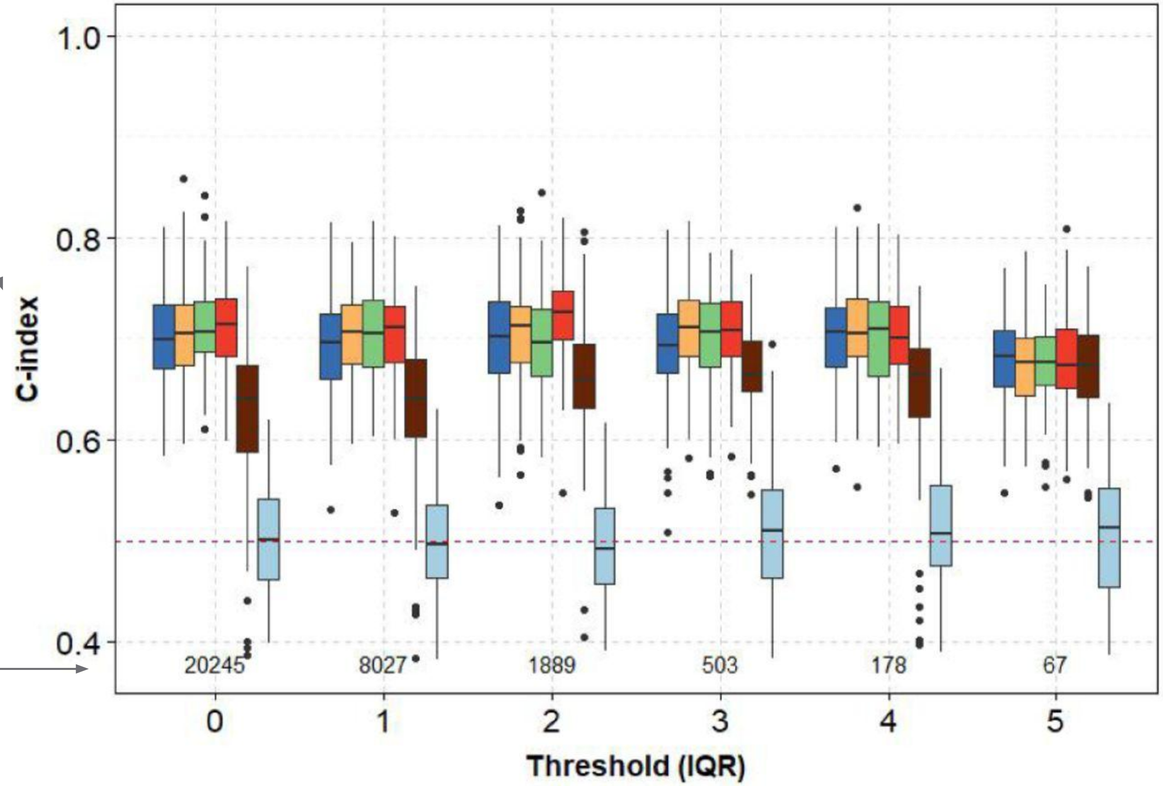
- mRNA-seq: sequence a random sample of the RNA expressed inside a set of cells
- Compare expression levels between two categories of subjects (e.g., cancer vs. control)
- Try out various combinations of LASSO + ridge

- How well do the trained models perform?
- How many genes are retained?

Jardiller et al (2020) *BioRXiv*

C-index: a measure of concordance between predictions
(more correlation that straight accuracy)

🟦 Lasso
🟧 ElasticNet
🟩 Adaptive ElasticNet
🟥 Ridge
🟫 Plain regression

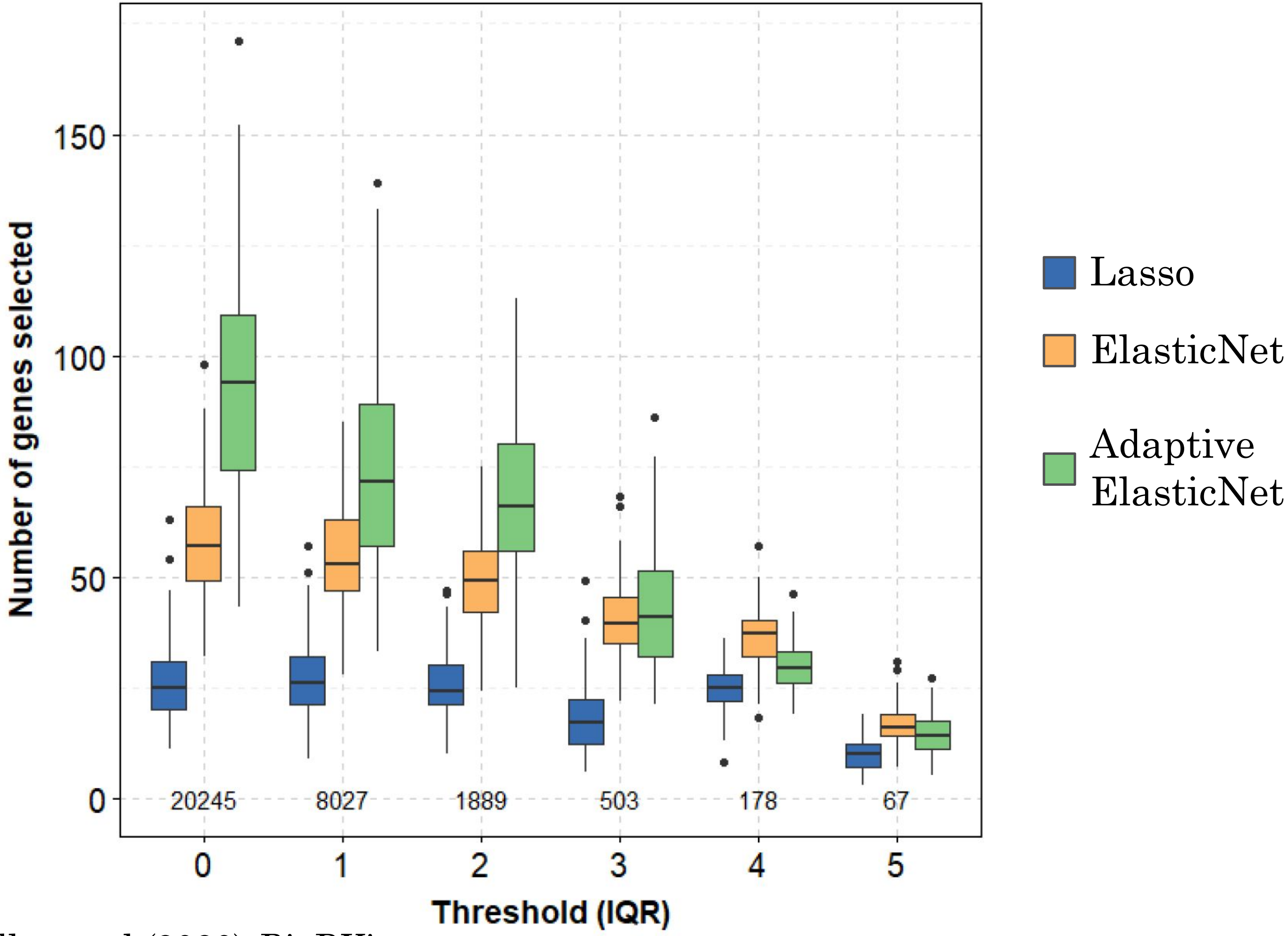


# of genes

Simple filtering method!

Jardiller et al (2020) *BioRXiv*

Renal carcinoma



Jardiller et al (2020) *BioRXiv*

Feature Extraction

Try to condense n variables into $< n$ derived variables or ‘metavariables’

Simple example: remove 1 of 2 identical variables from a data set

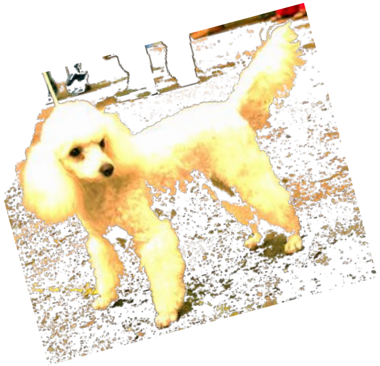
Principal Components Analysis (Pearson, 1901)

- Assume that there is (not necessarily complete) redundancy among variables in the data set

- We want to create *metavariables* that capture this redundancy

Poodle vector 1

Poodle vector 2



Metapoodle

http://commons.wikimedia.org/wiki/File:Pudel_miniatura_342.jpg

The covariance matrix

$$Cov(x,y) = \frac{\sum_{i=1}^{n}(x_i - \mu_x)(y_i - \mu_y)}{n-1}$$

C =

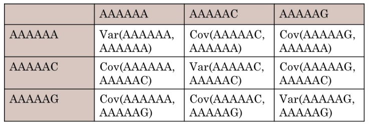
	AAAAAA	AAAAAC	AAAAAG
AAAAAA	Var(AAAAAA,AAAAAA)	Cov(AAAAAC,AAAAAA)	Cov(AAAAAG, AAAAAA)
AAAAAC	Cov(AAAAAA,AAAAAC)	Var(AAAAAC,AAAAAC)	Cov(AAAAAG, AAAAAC)
AAAAAG	Cov(AAAAAA,AAAAAG)	Cov(AAAAAC,AAAAAG)	Var(AAAAAG,AAAAAG)

Eigenvectors and eigenvalues

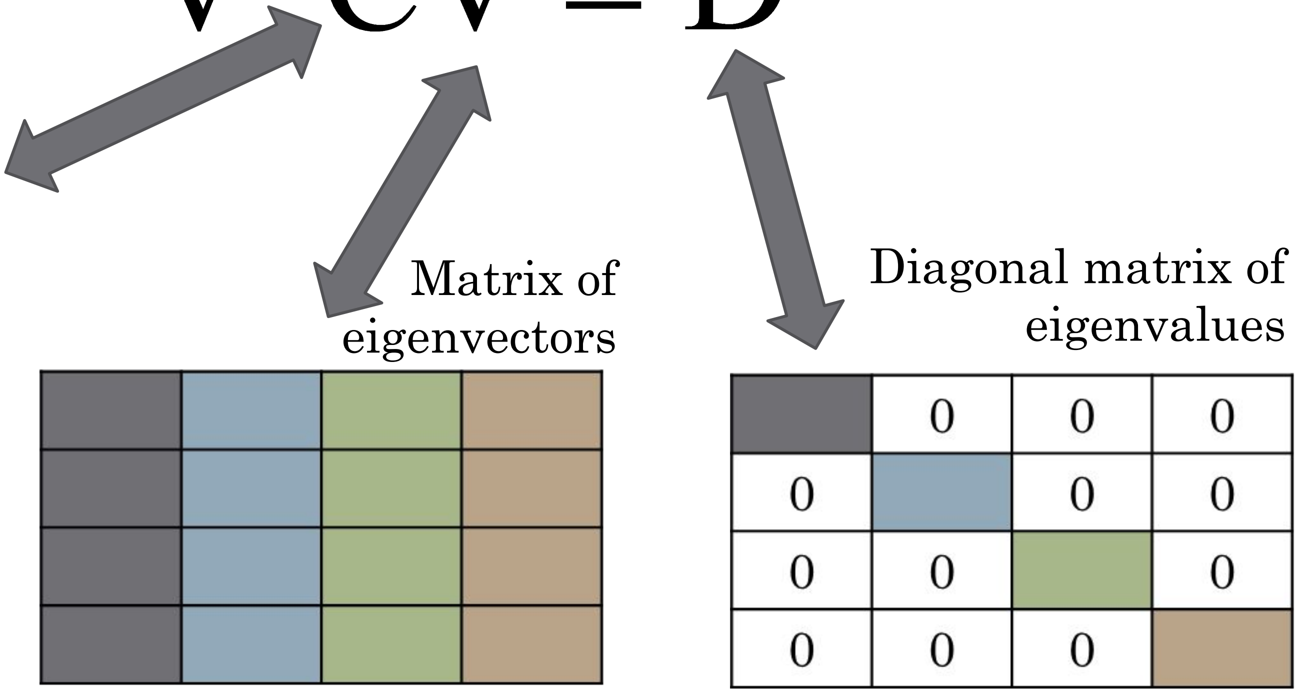
Diagonalize C using the matrix V

$$V^{-1}CV = D$$

Covariance matrix



	AAAAAA	AAAAAC	AAAAAG
AAAAAA	Var(AAAAAA, AAAAAA)	Cov(AAAAAC, AAAAAA)	Cov(AAAAAG, AAAAAA)
AAAAAC	Cov(AAAAAA, AAAAAC)	Var(AAAAAC, AAAAAC)	Cov(AAAAAG, AAAAAC)
AAAAAG	Cov(AAAAAA, AAAAAG)	Cov(AAAAAC, AAAAAG)	Var(AAAAAG, AAAAAG)



Matrix of eigenvectors

⬛	🟦	🟩	🟫
⬛	🟦	🟩	🟫
⬛	🟦	🟩	🟫
⬛	🟦	🟩	🟫

Diagonal matrix of eigenvalues

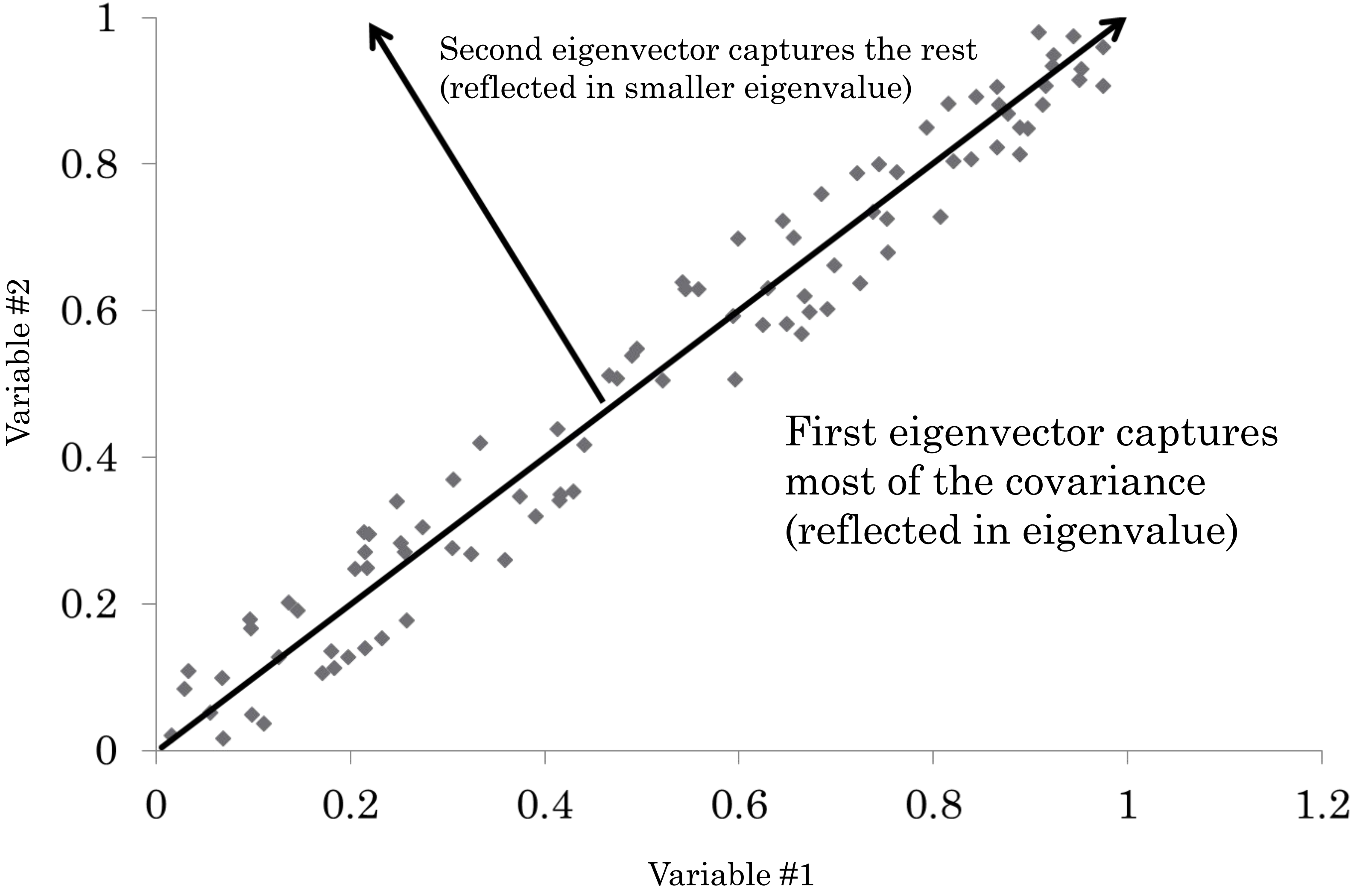
⬛	0	0	0
0	🟦	0	0
0	0	🟩	0
0	0	0	🟫

The eigenvectors capture **shared elements of covariance** from the original variables

The eigenvectors are mutually **orthogonal**

Graphical depiction of eigenvectors

for great understanding

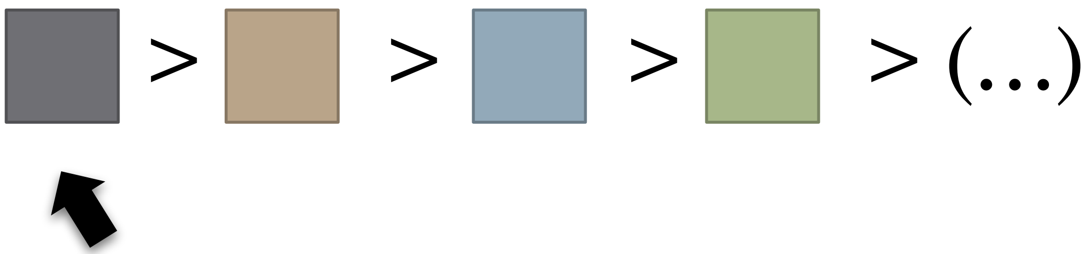


Second eigenvector captures the rest
(reflected in smaller eigenvalue)

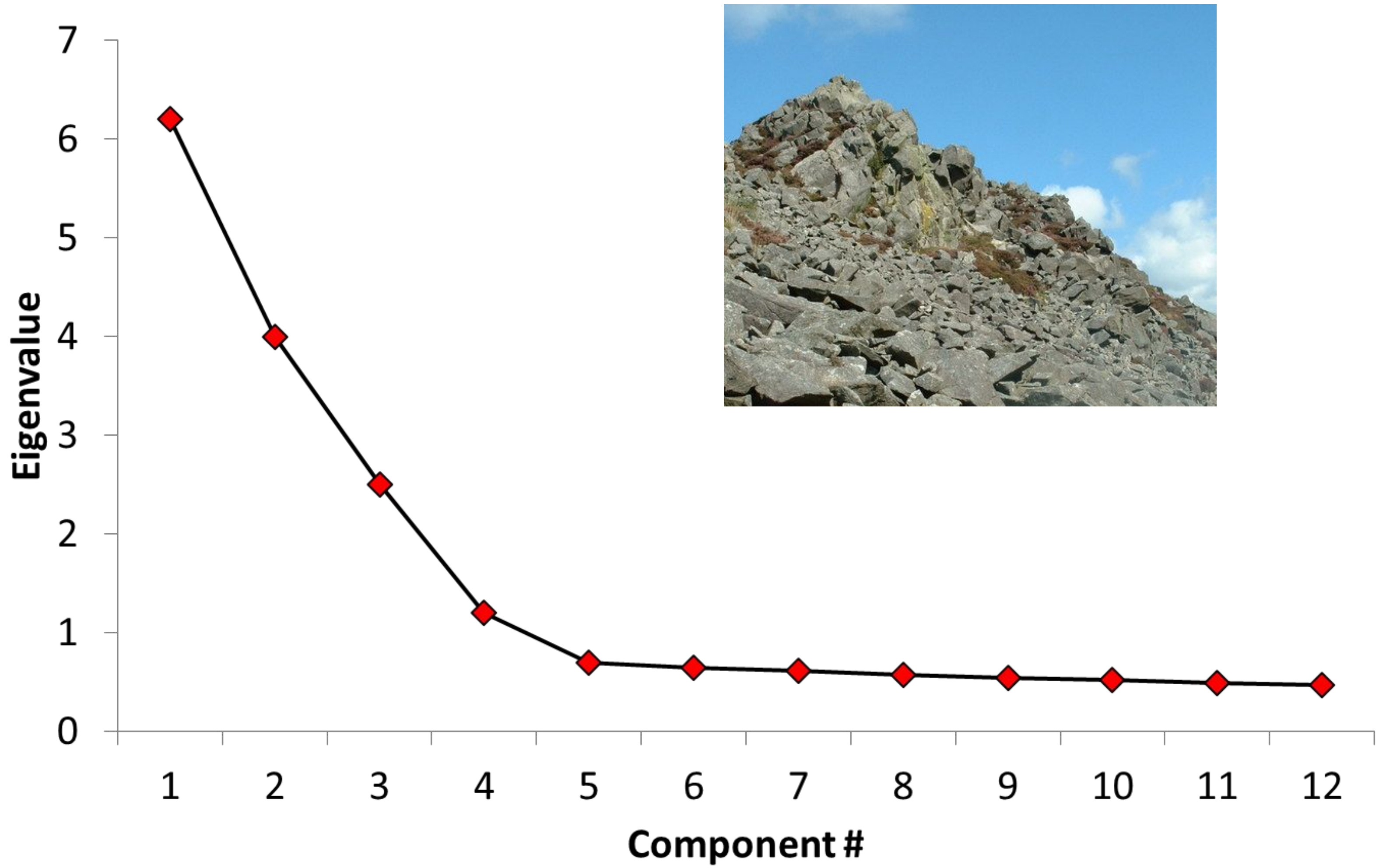
First eigenvector captures
most of the covariance
(reflected in eigenvalue)

Choosing components

Sort by eigenvalue



Component 1 captures the greatest amount of shared covariance from the original data (proportional to the corresponding eigenvalue)



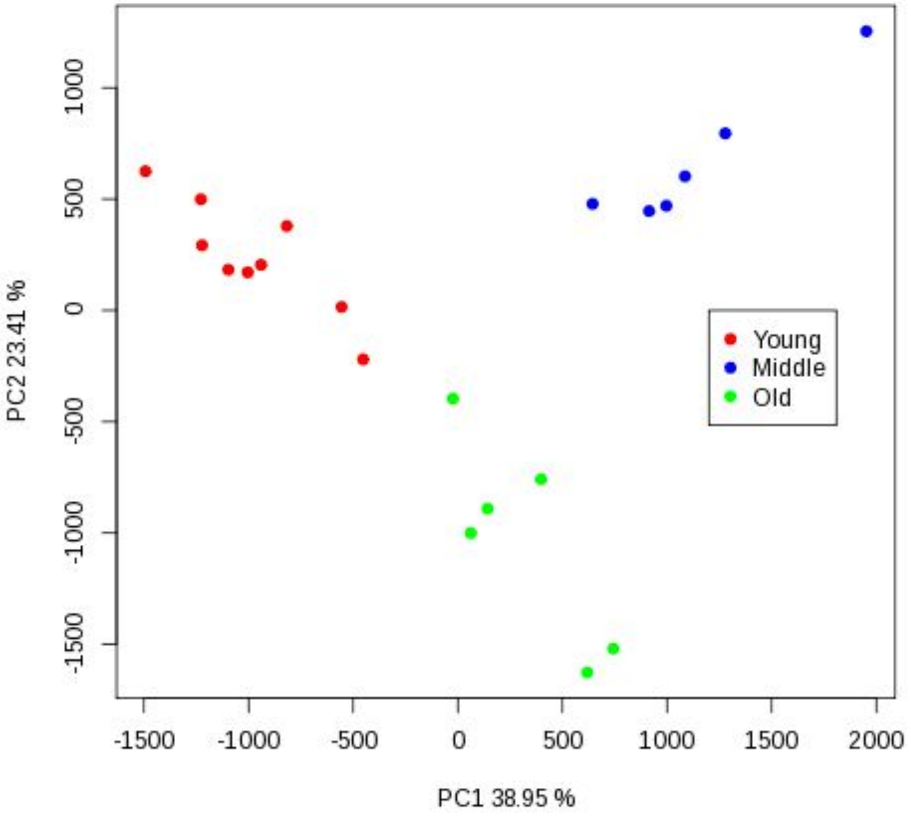
Scree plot

http://commons.wikimedia.org/wiki/File:Carningli_scree_slope_-_geograph.org.uk_-_394198.jpg

Graphical view

Input: estimated bacterial species frequencies for 21 mouse fecal samples

Plot of first two principal components (with % of variance explained)



Langille MG, Meehan CJ, Koenig JE, Dhanani AS, Rose RA, Howlett SE, Beiko RG (2014) Microbial shifts in the aging mouse gut. *Microbiome* 2:50.

Loadings OR *What am I Looking At?*

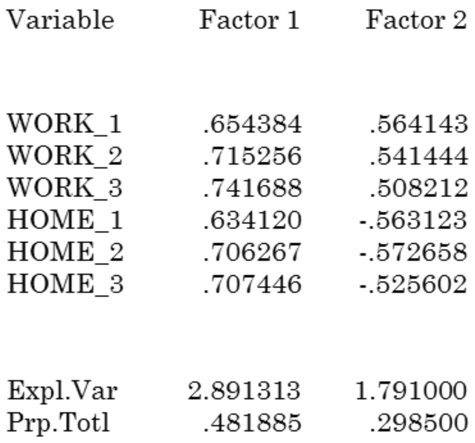
The contribution of each variable to each component

	Component 1	Component 2
AAAAAA	0.9	0.03
AAAAAC	0.8	-0.007
AAAAAG	0.84	0.01
...	…	…
GGACCT	0.02	0.15
GGACGA	-0.01	-0.14

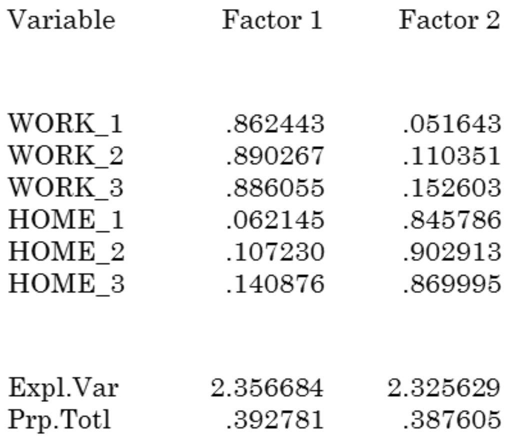
Limitation of feature extraction methods:
what exactly is signified by component *i*?

Interpretation-friendly techniques

The component vectors can be collectively *rotated* to simplify the loadings



Variable	Factor 1	Factor 2
WORK_1	.654384	.564143
WORK_2	.715256	.541444
WORK_3	.741688	.508212
HOME_1	.634120	-.563123
HOME_2	.706267	-.572658
HOME_3	.707446	-.525602
Expl.Var	2.891313	1.791000
Prp.Totl	.481885	.298500



Variable	Factor 1	Factor 2
WORK_1	.862443	.051643
WORK_2	.890267	.110351
WORK_3	.886055	.152603
HOME_1	.062145	.845786
HOME_2	.107230	.902913
HOME_3	.140876	.869995
Expl.Var	2.356684	2.325629
Prp.Totl	.392781	.387605

http://www.statsoft.com/textbook/stfacan.html

Summary

1. We can generate as many features as we want from DNA and protein sequences

2. Not all of these will be USEFUL or INDEPENDENT predictors

3. We should therefore reduce the complexity of the problem using good design and reduction methods