Data Mining in Astronomy & regression

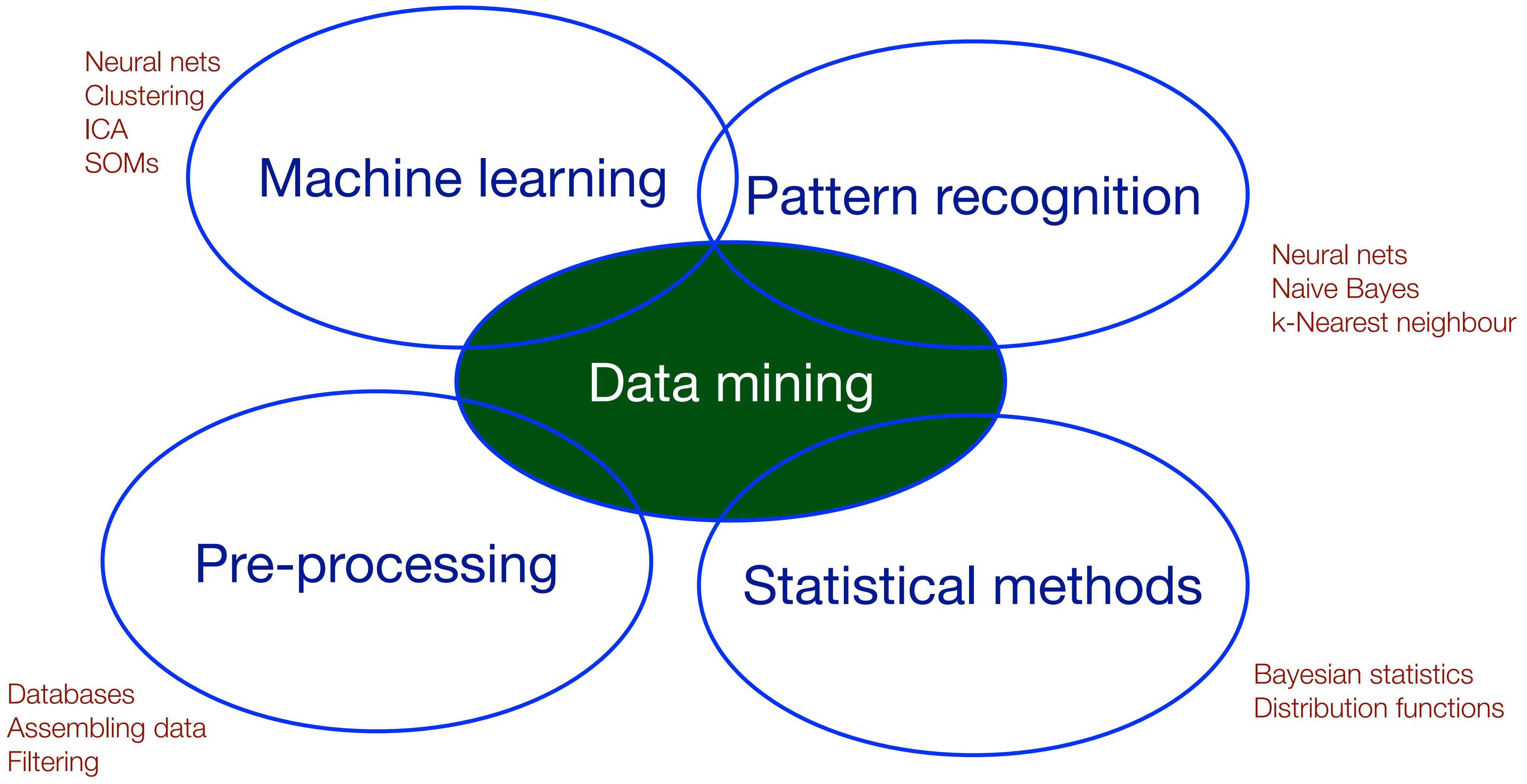
What is data mining/machine learning?

WE HAVE A GIGANTIC DATABASE FULL OF CUSTOMER BEHAVIOR INFORMATION.
EXCELLENT. WE CAN USE NON-LINEAR MATH AND DATA MINING TECHNOLOGY TO OPTIMIZE OUR RETAIL CHANNELS!
IF THAT'S THE SAME THING AS SPAM, WE'RE HAVING A GOOD MEETING HERE.
www.dilbert.com scottadams@aol.com
© 2000 United Feature Syndicate, Inc.
11/13/00

What is data mining/machine learning?

One possible definition:

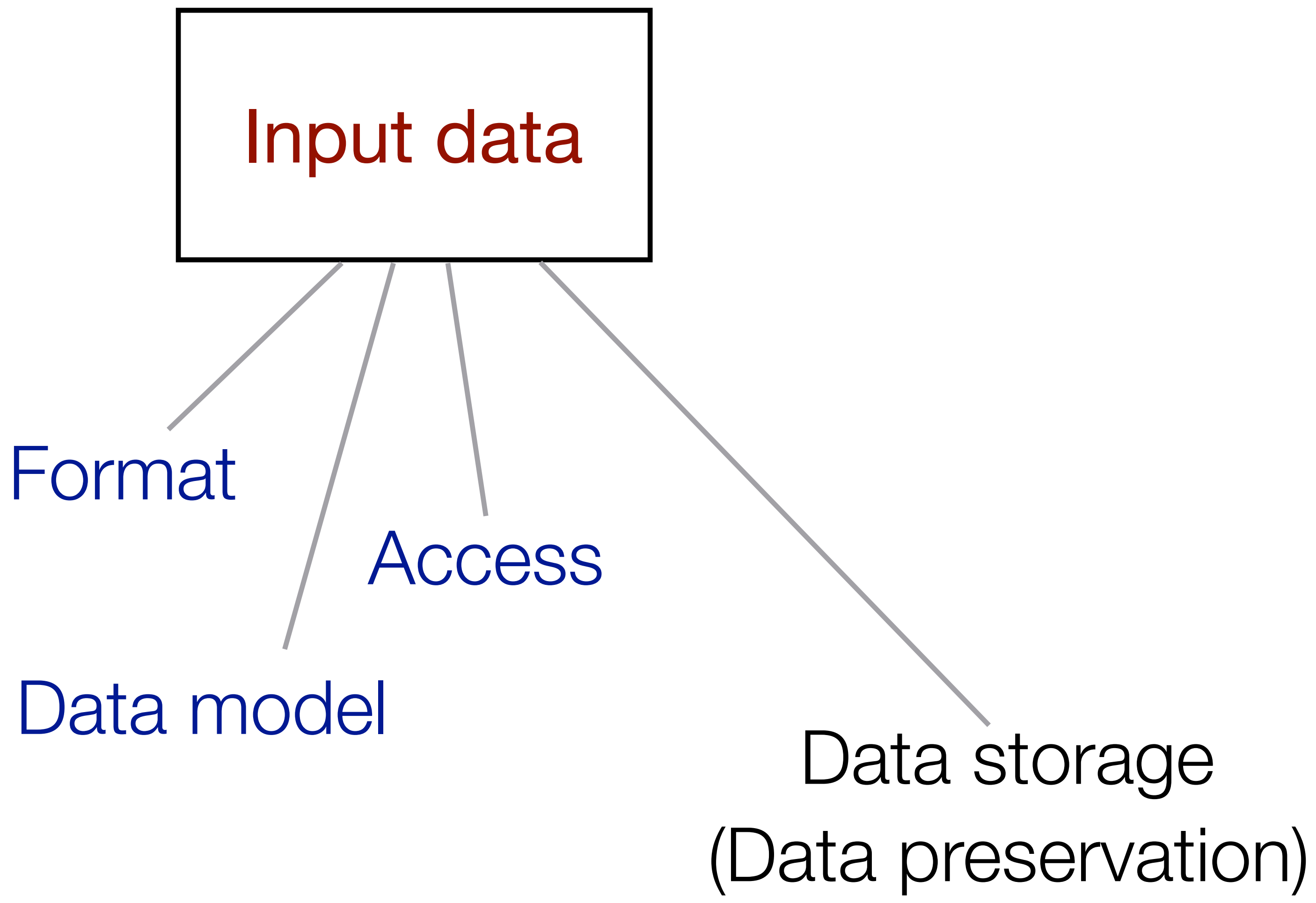
The process of extracting patterns from data



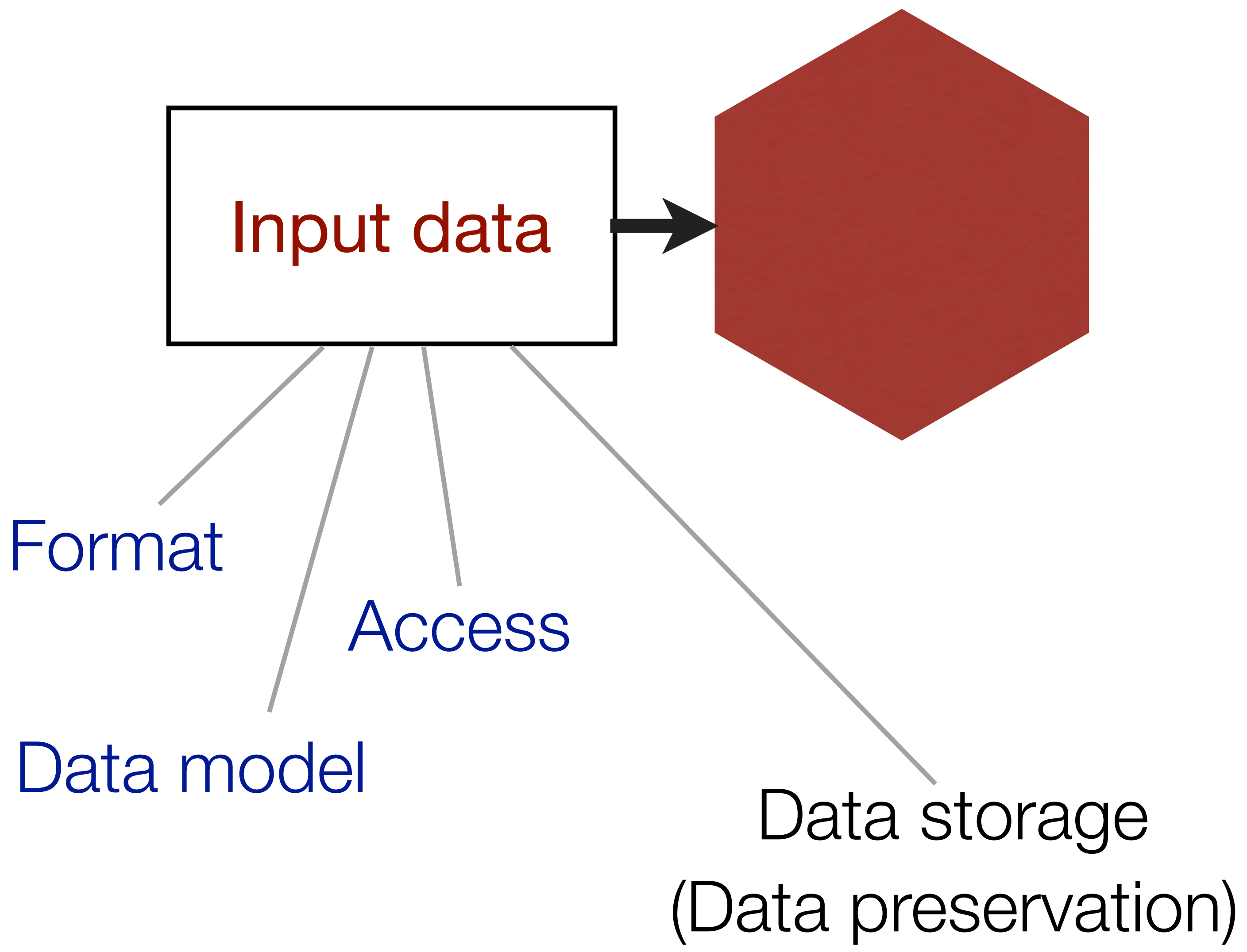
Note that in this course this includes the storing and organisation of data!

I will take data mining, machine learning, and pattern recognition to be synonyms.

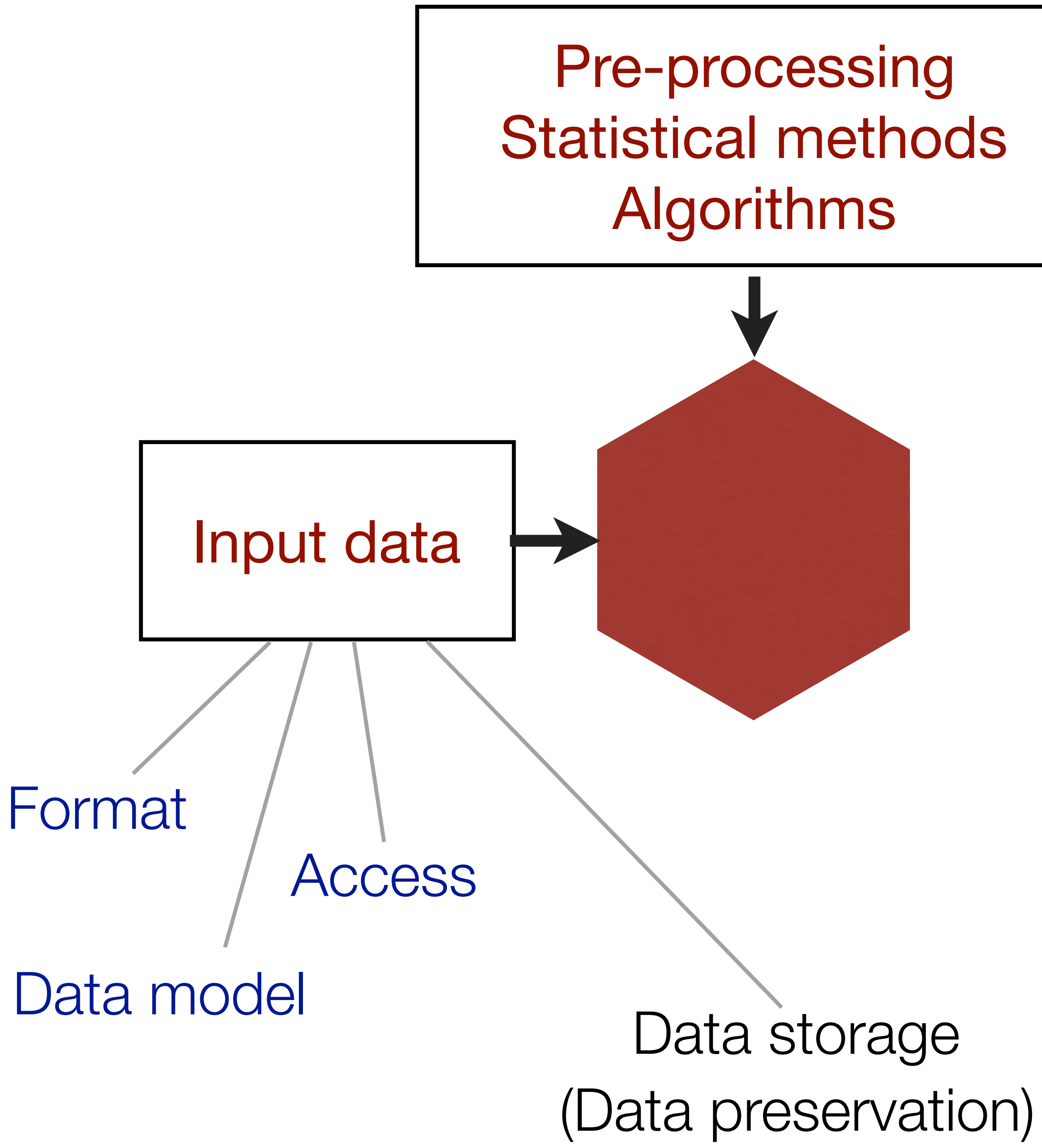
Data Mining Outline



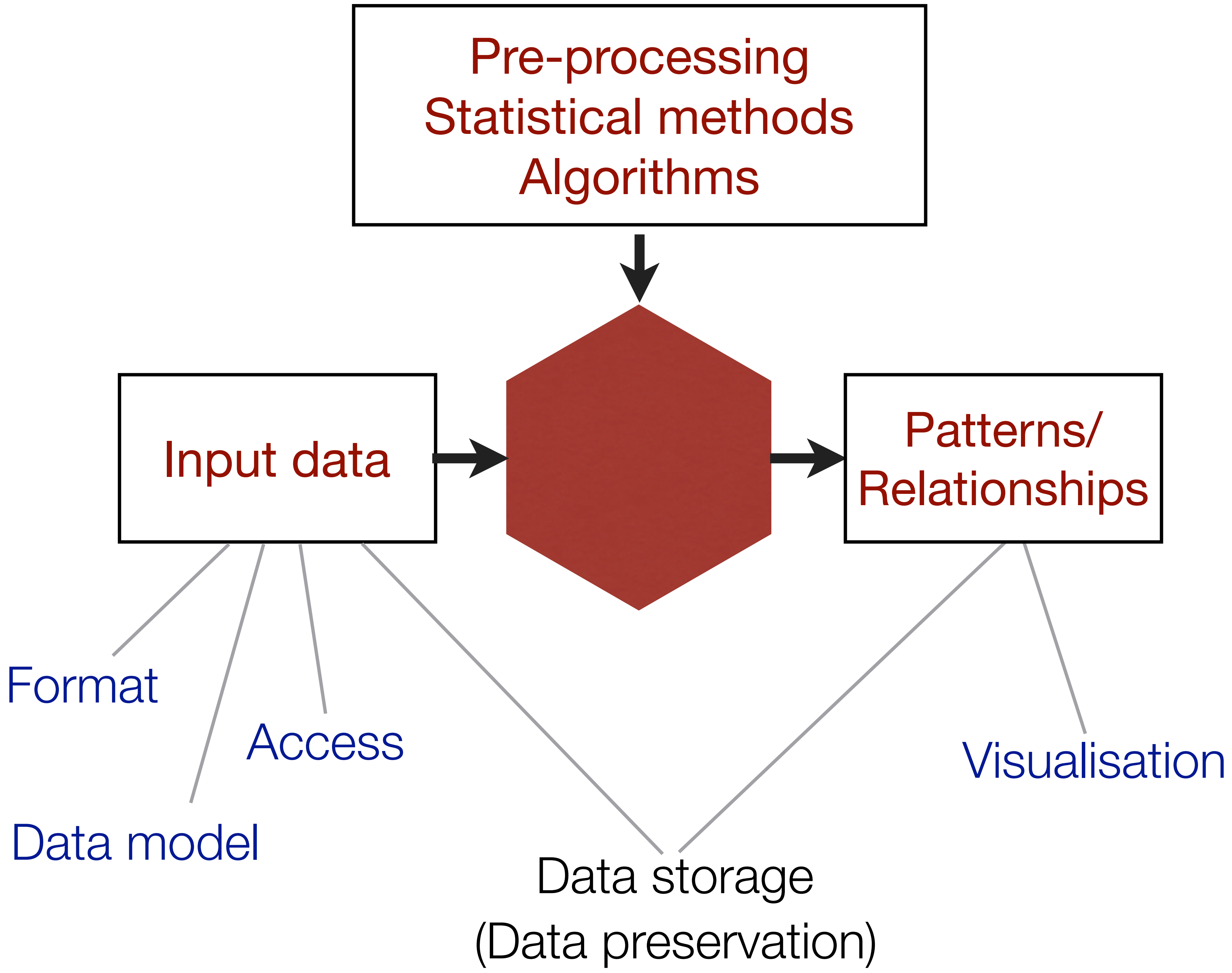
Data Mining Outline



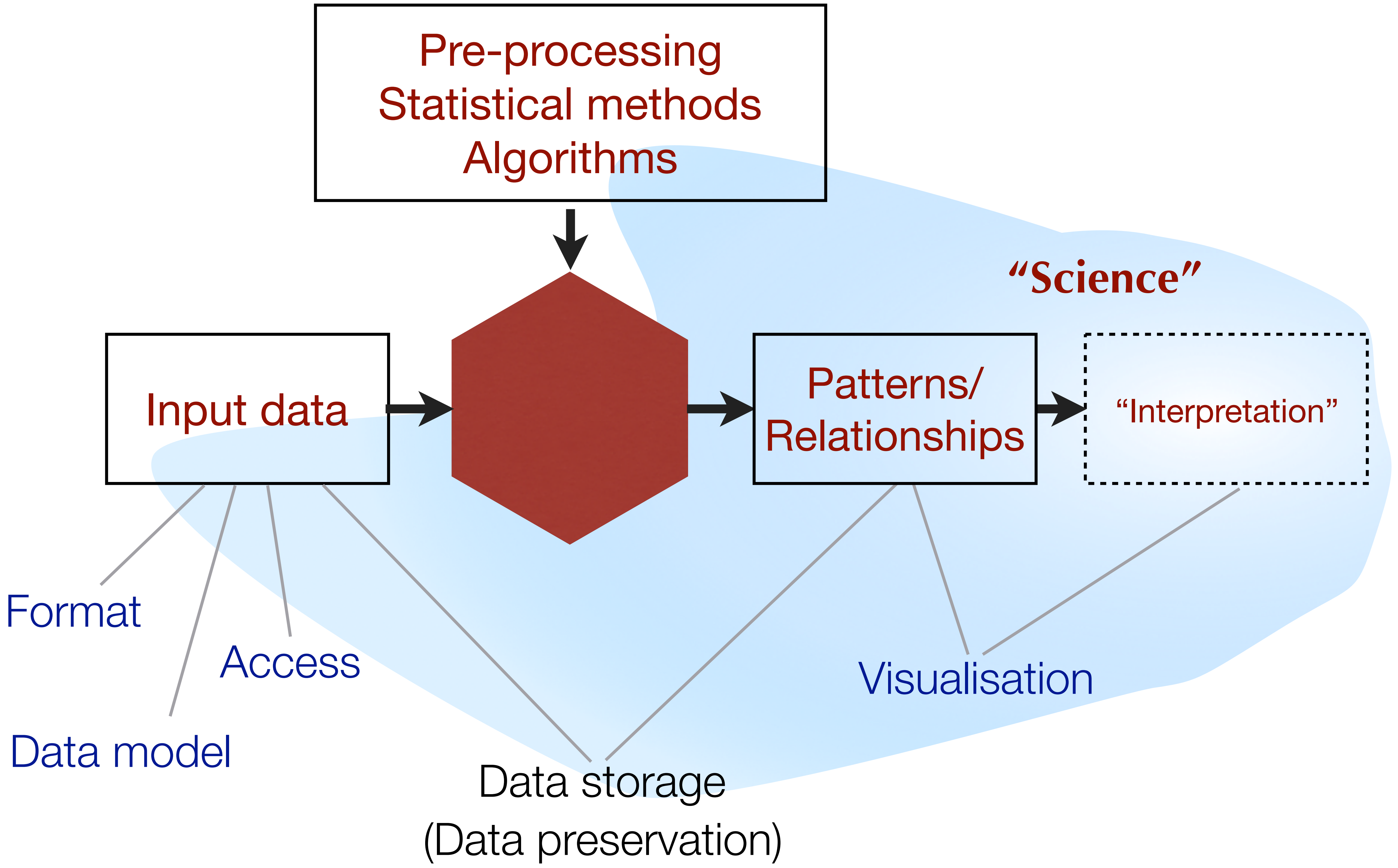
Data Mining Outline



Data Mining Outline



Data Mining Outline



What is machine learning?

From “Deep Learning”, Goodfellow, Bengio, Courville

“Machine learning is essentially a form of applied statistics with increased emphasis on the use of computers to statistically estimated complicated functions and a decreased emphasis on prov[id]ing confidence intervals around these functions”

Examples outside of astronomy:

Amazon recommends .....

Use the buying patterns of all customers to create characteristic directions (books) in a multi-dimensional space. Apply this to individual customers to predict their preferences.

Examples outside of astronomy:

Amazon recommends .....

Use the buying patterns of all customers to create characteristic directions (books) in a multi-dimensional space. Apply this to individual customers to predict their preferences.

Organisation of merchandise in a supermarket

Find clusters of products that are bought together. Ensure you don't discount both at the same time.

Examples outside of astronomy:

Amazon recommends .....

Use the buying patterns of all customers to create characteristic directions (books) in a multi-dimensional space. Apply this to individual customers to predict their preferences.

Organisation of merchandise in a supermarket

Find clusters of products that are bought together. Ensure you don't discount both at the same time.

Associating genome sequences

Identify groups of genomes, cross-correlate, look for structure

Examples outside of astronomy:

Amazon recommends .....

Use the buying patterns of all customers to create characteristic directions (books) in a multi-dimensional space. Apply this to individual customers to predict their preferences.

Organisation of merchandise in a supermarket

Find clusters of products that are bought together. Ensure you don't discount both at the same time.

Associating genome sequences

Identify groups of genomes, cross-correlate, look for structure

Identifying plant diseases

Train machines using the strategy of human experts - maybe they will outperform their teachers in the end (it happened for soy plants).

Examples outside of astronomy:

Amazon recommends .....

Use the buying patterns of all customers to create characteristic directions (books) in a multi-dimensional space. Apply this to individual customers to predict their preferences.

Organisation of merchandise in a supermarket

Find clusters of products that are bought together. Ensure you don't discount both at the same time.

Associating genome sequences

Identify groups of genomes, cross-correlate, look for structure

Identifying plant diseases

Train machines using the strategy of human experts - maybe they will outperform their teachers in the end (it happened for soy plants).

Mail order catalogue

Use past shopping patterns to predict who are the most likely to respond to your new catalogue.

Examples inside astronomy:

Regression: Hubble's law, Tully-Fisher relation ++++

Clustering: Identifying asteroid classes from SDSS photometry

Hierarchical trees: Galaxy morphology classification, photometric redshifts

MCMC & friends: Most fields.

Density estimators: Most fields.

Support Vector Machines: Stellar classification

Gaussian process regression: Time-series, galaxy formation

Random trees: Photometric redshifts for galaxies

Principal Component Analysis: PSF estimation, post-starbursts, data reduction improvements +++

Self-organising maps: Photometric redshifts, Υ-ray source classification.

Neural nets: Photometric redshifts, stellar classification, galaxy morphology

Deep neural nets: galaxy morphology, image generation, gravitational lens finders

What does it involve?

Data preparation

e.g.: Extract measurements from data
Check for quality/accuracy

What does it involve?

Data preparation

e.g.: Extract measurements from data
Check for quality/accuracy

Apply a data mining technique

e.g. from pattern recognition/machine learning

What does it involve?

Data preparation

e.g.: Extract measurements from data
Check for quality/accuracy

Sometimes optional

Inject data into a database

Create the database structure.
Read data using X and insert into database using SQL.

Get data from database

Apply a data mining technique

e.g. from pattern recognition/machine learning

Data mining

Sometimes it is referred to as ‘**knowledge discovery**’, but some people prefer ‘**knowledge creation**’

It can typically be divided into a number of methods:

- Classification
    - Given a set of images, say, that we know the morphology of, we want to be able to classify a large set of other objects. Other examples: Classification of stellar spectra with GAIA, classification of transient sources, variable source detection etc.
- Estimation/prediction
- Grouping/clustering
- Dimension reduction

Data mining

Sometimes it is referred to as ‘**knowledge discovery**’, but some people prefer ‘**knowledge creation**’

It can typically be divided into a number of methods:

- Classification
- Estimation/prediction
    - You have X, Y, Z, ... and you want to predict a value A. An example would be if you are given a set of emission lines and you want to estimate the metal abundance in the gas. Most fitting methods fall into this category. Another area of interest is to estimate distributions from a few observations - the aim of kernel estimation techniques.
- Grouping/clustering
- Dimension reduction

Data mining

Sometimes it is referred to as ‘**knowledge discovery**’, but some people prefer ‘**knowledge creation**’

It can typically be divided into a number of methods:

- Classification
- Estimation/prediction
- Grouping/clustering
    - This is used to find groups objects with similar properties. One example is to find different asteroid families from orbital information. Another might be to find galaxy clusters, or simply to find outliers/weirdoes.
- Dimension reduction

Data mining

Sometimes it is referred to as ‘**knowledge discovery**’, but some people prefer ‘**knowledge creation**’

It can typically be divided into a number of methods:

- Classification
- Estimation/prediction
- Grouping/clustering
- Dimension reduction
    - It is much easier to handle & visualise low-dimensional data. Dimension reduction techniques allow you to go from multi-dimensional data, with 1000s of dimensions, to a manageable set of variables. This can be used for model fitting (e.g. principal components analysis) and for exploratory data visualisation.

Why is this important?

Big surveys — HENCE DATABASES

The last decade has seen a trend towards big surveys and massive theoretical calculations.
This will continue - and it will produce vast amounts of data.

Large amounts of data — HENCE STATISTICS

Large amounts of data offers a lot of potential but can be hard to work with/explore. This means that we will need other tools to make optimal use of the new data.

Sharing of data and resources — HENCE VO/GRID/CLOUD..

In the future it will likely be essential to share your data to get the most out of them and many surveys already plan how to do this. But also computing resources might be shared differently than today - using e.g. grid computing.

This is all true for observational, experimental and theoretical data!

Some specific examples

- Data rates are increasing - astronomers today produce data at about ~few Tb/night integrated over all telescopes. But LSST should produce ~15 Tb *per night*! [3 sq deg each 10-15 seconds]. How do you store/organise the data?

- These large surveys produce large number of objects. (SDSS & 2MASS all contain > 100 million objects with perhaps 1000 attributes each). How do you check that all data are ok?

- How do you analyse these kinds of data? To find the closest match using naïve search on 10^8 objects requires 10^{16} operations so correlation functions can be hard to calculate - and how do you find new relationships & events?

Some specific examples

- Data rates are increasing - astronomers today produce data at about ~few Tb/night integrated over all telescopes. But LSST should produce ~15 Tb *per night*! [3 sq deg each 10-15 seconds]. How do you store/organise the data?

  Pre-processing & data bases!

- These large surveys produce large number of objects. (SDSS & 2MASS all contain > 100 million objects with perhaps 1000 attributes each). How do you check that all data are ok?

- How do you analyse these kinds of data? To find the closest match using naïve search on 10^8 objects requires 10^{16} operations so correlation functions can be hard to calculate - and how do you find new relationships & events?

Some specific examples

- Data rates are increasing - astronomers today produce data at about ~few Tb/night integrated over all telescopes. But LSST should produce ~15 Tb *per night*! [3 sq deg each 10-15 seconds]. How do you store/organise the data?

Pre-processing & data bases!

- These large surveys produce large number of objects. (SDSS & 2MASS all contain > 100 million objects with perhaps 1000 attributes each). How do you check that all data are ok?

You don't! - robust analysis

- How do you analyse these kinds of data? To find the closest match using naïve search on 10^8 objects requires 10^{16} operations so correlation functions can be hard to calculate - and how do you find new relationships & events?

Some specific examples

- Data rates are increasing - astronomers today produce data at about ~few Tb/night integrated over all telescopes. But LSST should produce ~15 Tb *per night*! [3 sq deg each 10-15 seconds]. How do you store/ organise the data?

  Pre-processing & data bases!

- These large surveys produce large number of objects. (SDSS & 2MASS all contain > 100 million objects with perhaps 1000 attributes each). How do you check that all data are ok?

  You don't! - robust analysis

- How do you analyse these kinds of data? To find the closest match using naïve search on 10^8 objects requires 10^{16} operations so correlation functions can be hard to calculate - and how do you find new relationships & events?

  Data mining/statistical methods/ clever algorithms

Some numbers...

#Galaxies: 10^5 galaxies per square degree when going to m_I=25.5
(for every two magnitudes deeper you detect an order of magnitude more objects)

#Galaxies: 20,000 deg^2 “extragalactic sky” ⇒ ~10^9 sources

#Stars: $10^{5\text{-}6}$ stars per square degree per magnitude at low galactic latitudes

#Imaging data:

$$N_{\rm pix} = 1.44 \times 10^8 \left(\frac{\theta}{0.3''}\right)^2 \left(\frac{A}{1\deg^2}\right) \text{ pixels}$$

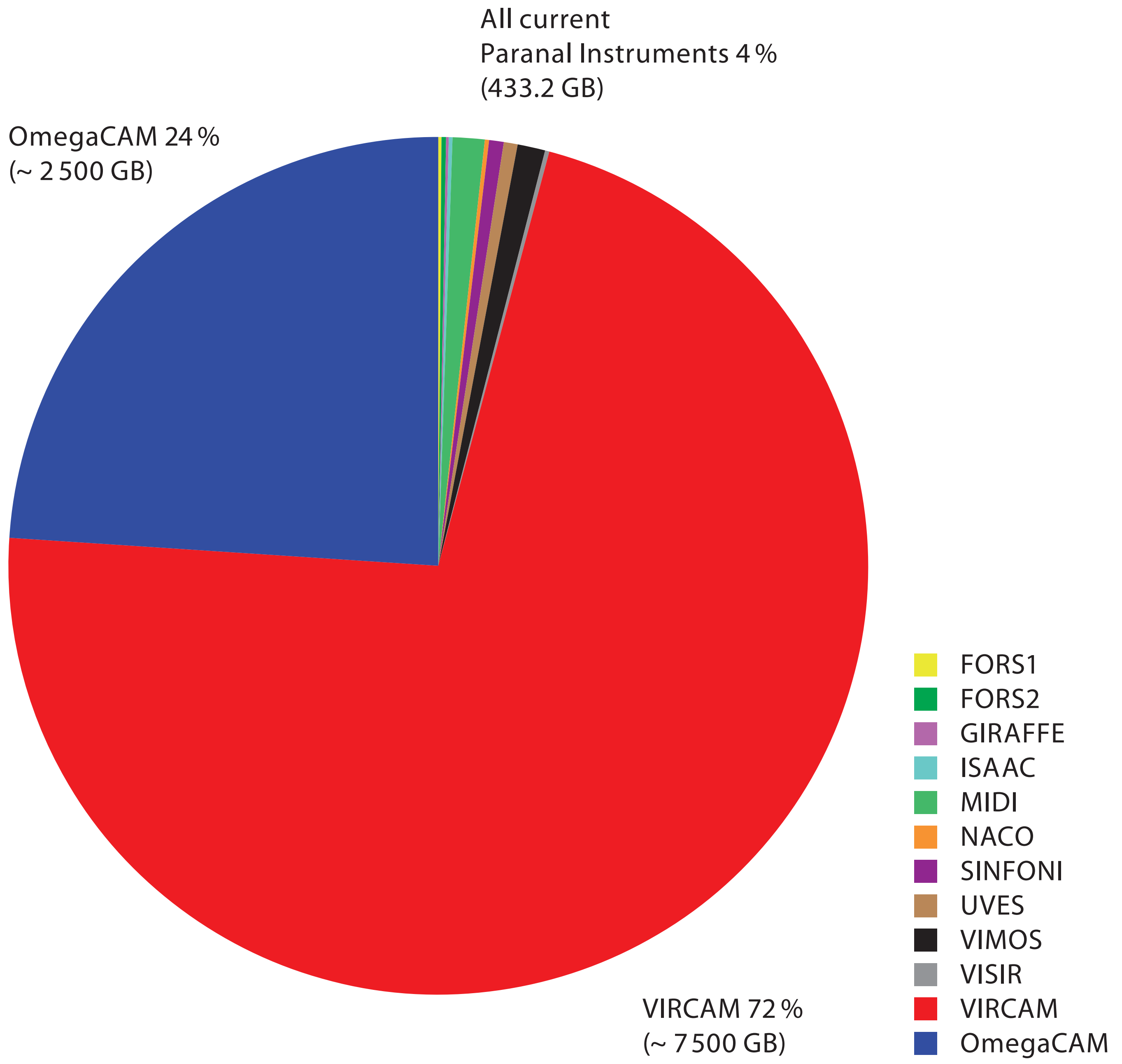
CFHT MegaCam: 36 CCDs with 2048 x 4612 pixels (340 megapixels)
~720 Mb per file for ~ 18000 pixels on the side (expanding to 1.4 Gb when converted to float...)

That is for one filter and one exposure - we normally need 4-5 filters and multiple exposures.

Recall:
#seconds per year ~ $3\text{x}10^7$
(top computers ~ 10^{15} Flops)

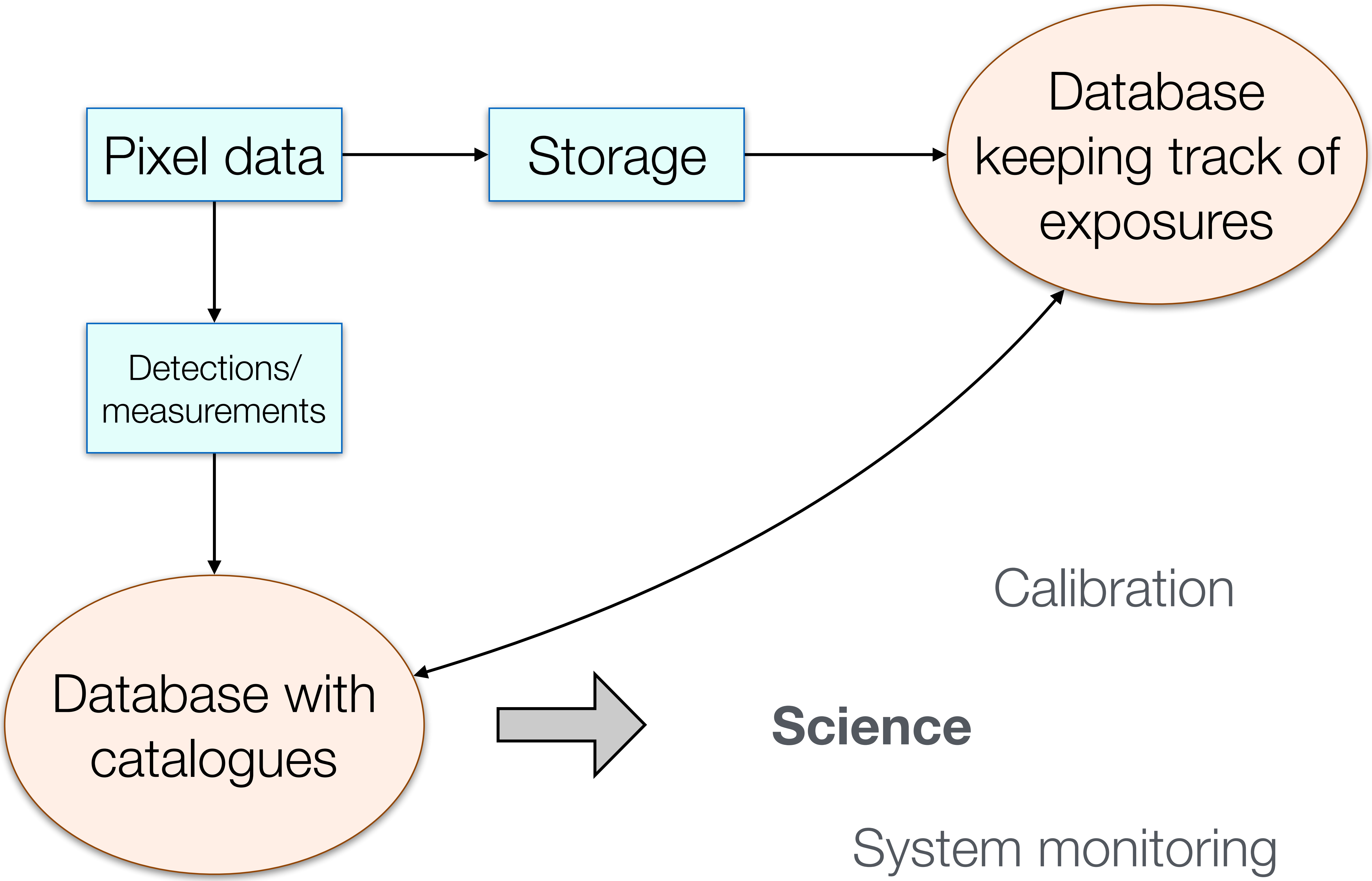
The ESO pie - or the changing world

The way the science changed over the last decade - and will continue to do so in the foreseeable future.



Arnaboldi et al (2007)

The role of databases - a modern view



Calibration

Science

System monitoring

Key ingredients of machine learning

Examples:

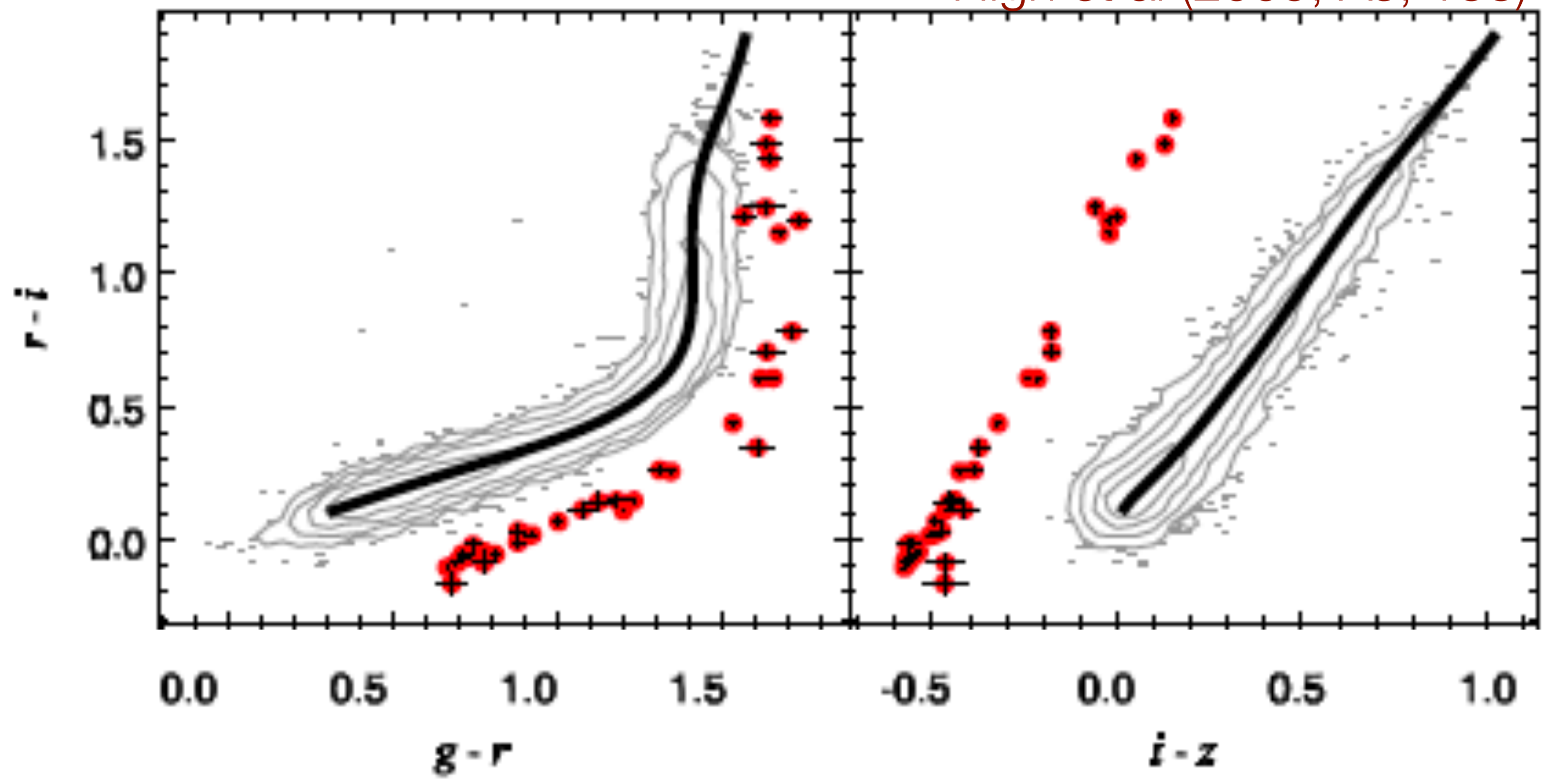
A machine learning method cannot “learn” unless it is provided with examples. These are composed of **features**.

Key ingredients of machine learning

Examples:

A machine learning method cannot “learn” unless it is provided with examples. These are composed of **features**.

High et al (2009, AJ, 138)

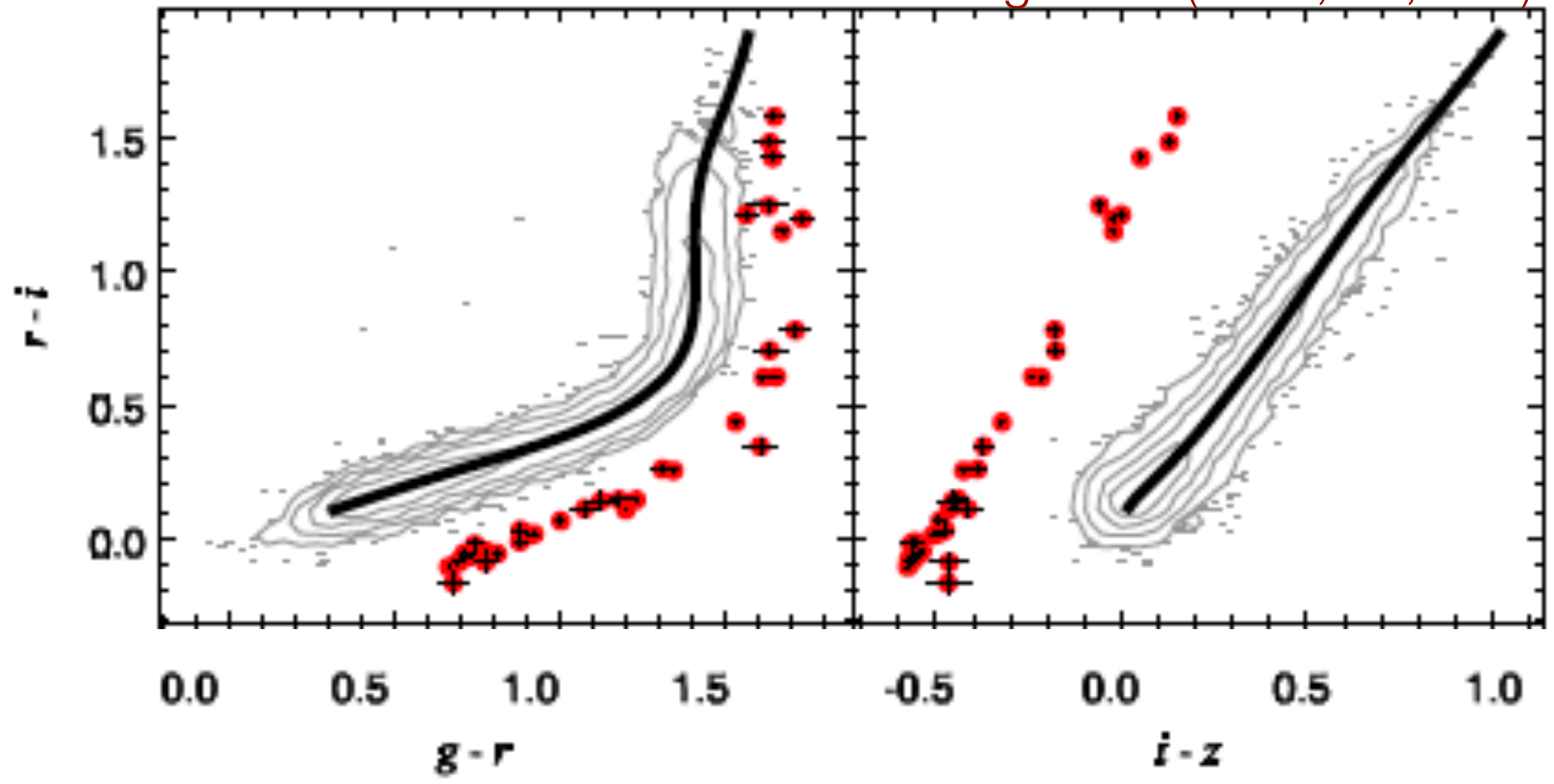


Key ingredients of machine learning

Examples:

A machine learning method cannot “learn” unless it is provided with examples. These are composed of **features**.

High et al (2009, AJ, 138)



Features: g-r, r-i, i-z

Example: [g-r, r-i, i-z]

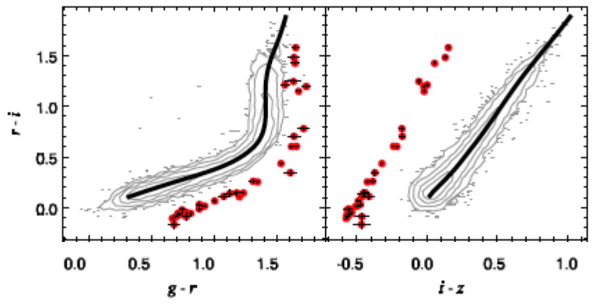
Key ingredients of machine learning

Examples:

A machine learning method cannot “learn” unless it is provided with examples. These are composed of **features**.

High et al (2009, AJ, 138)

Features: g-r, r-i, i-z
Example: [g-r, r-i, i-z]



Often summarised in a **design matrix**. Here each example makes up one row - e.g. for 4 points:

$$\begin{pmatrix} g_1 - r_1 & r_1 - i_1 & i_1 - z_1 \\ g_2 - r_2 & r_2 - i_2 & i_2 - z_2 \\ g_3 - r_3 & r_3 - i_3 & i_3 - z_3 \\ g_4 - r_4 & r_4 - i_4 & i_4 - z_4 \end{pmatrix}$$

Key ingredients of machine learning

Accuracy/error-rate

How well is our algorithm doing? A common measure - the mean squared error:

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y - y_{\mathrm{pred}})^2$$

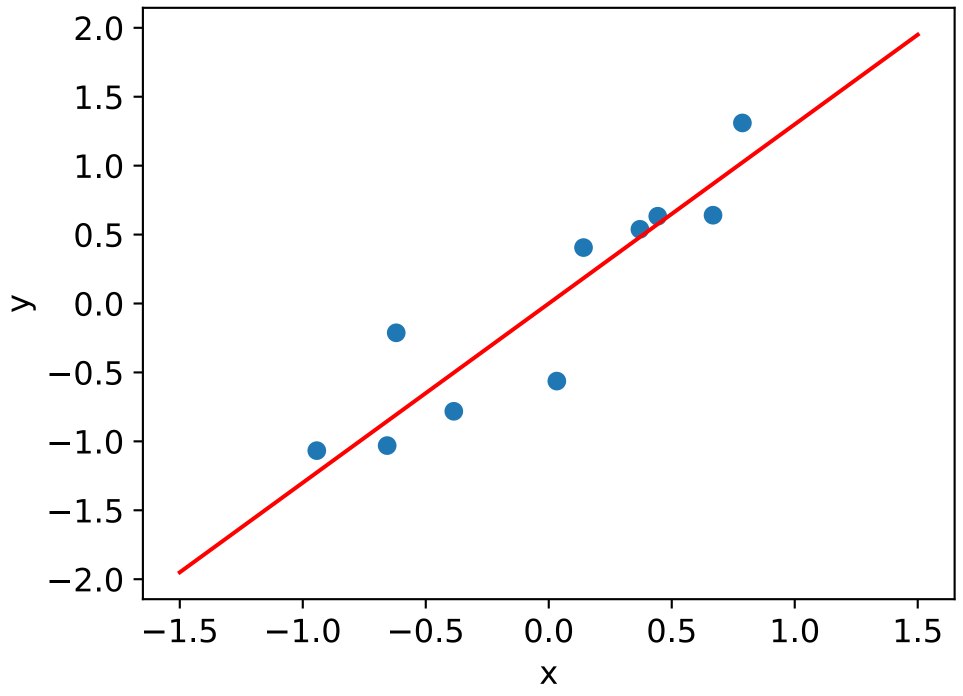
Key ingredients of machine learning

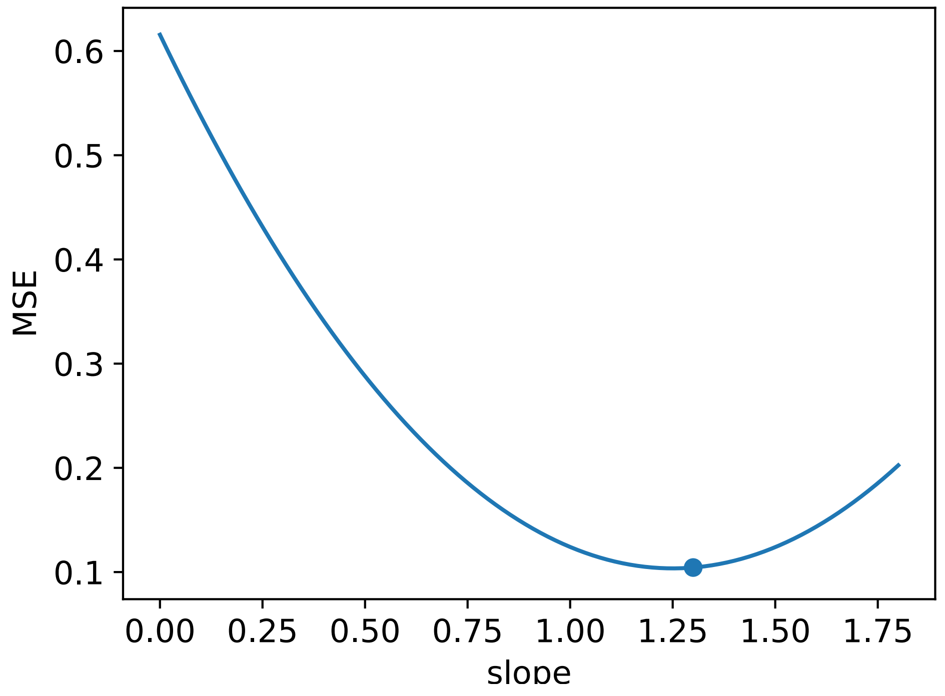
Accuracy/error-rate

How well is our algorithm doing? A common measure - the mean squared error:

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y - y_{\mathrm{pred}})^2$$

Try this on linear regression $y = \mathrm{slope} \times x$

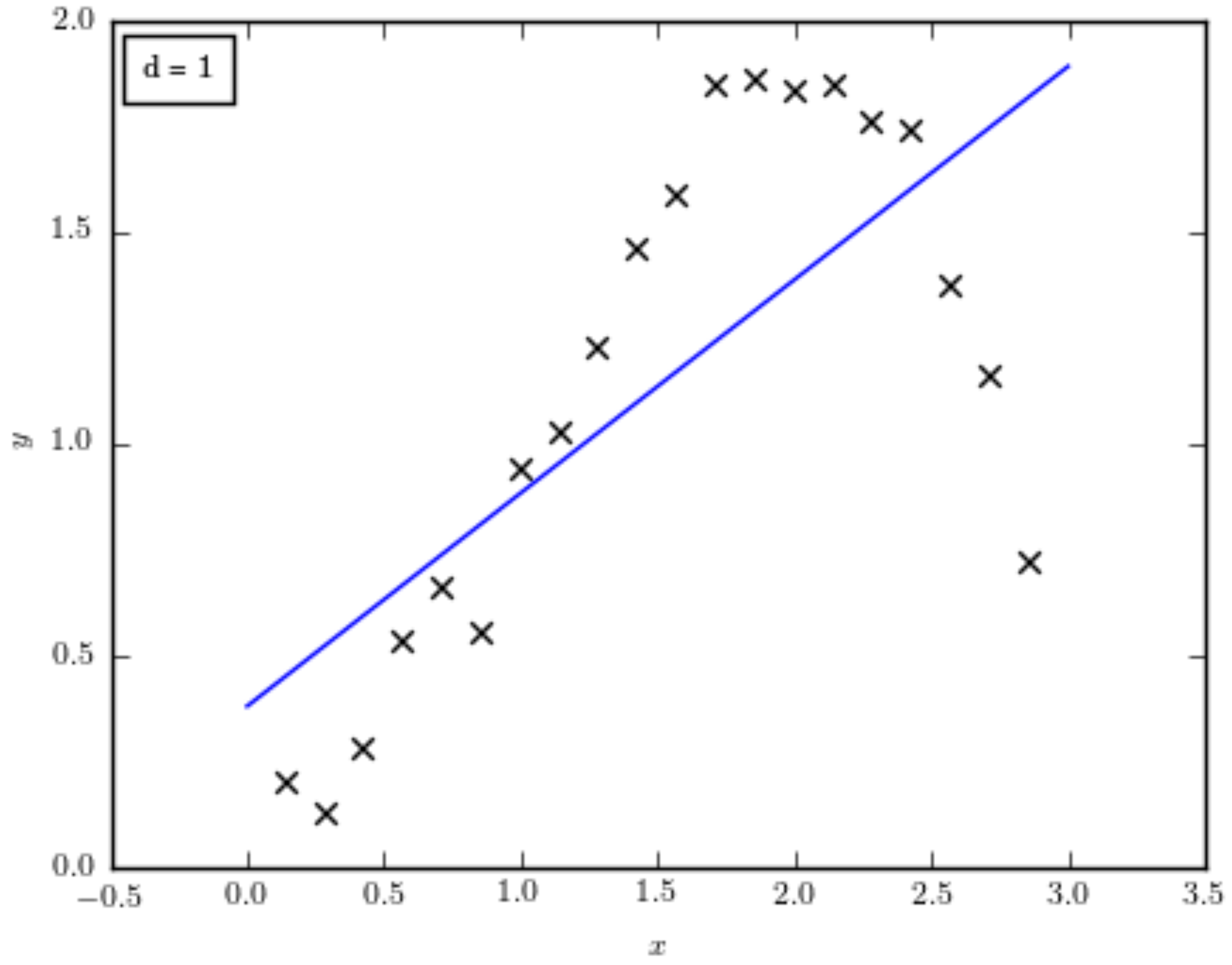




A simple example - fitting a polynomial

Overfitting, underfitting

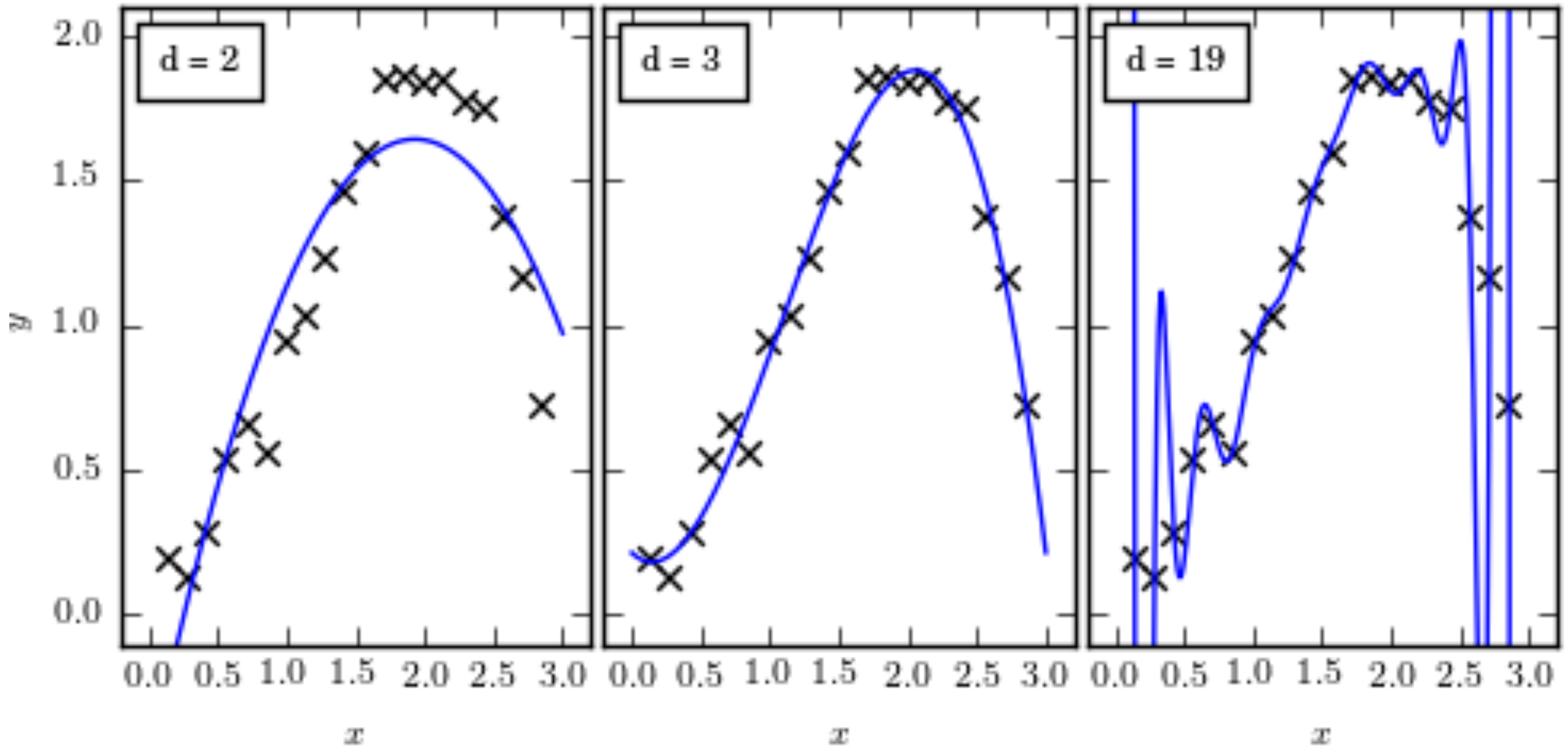
We want our method to **generalise** well



High bias!

From astroML.org

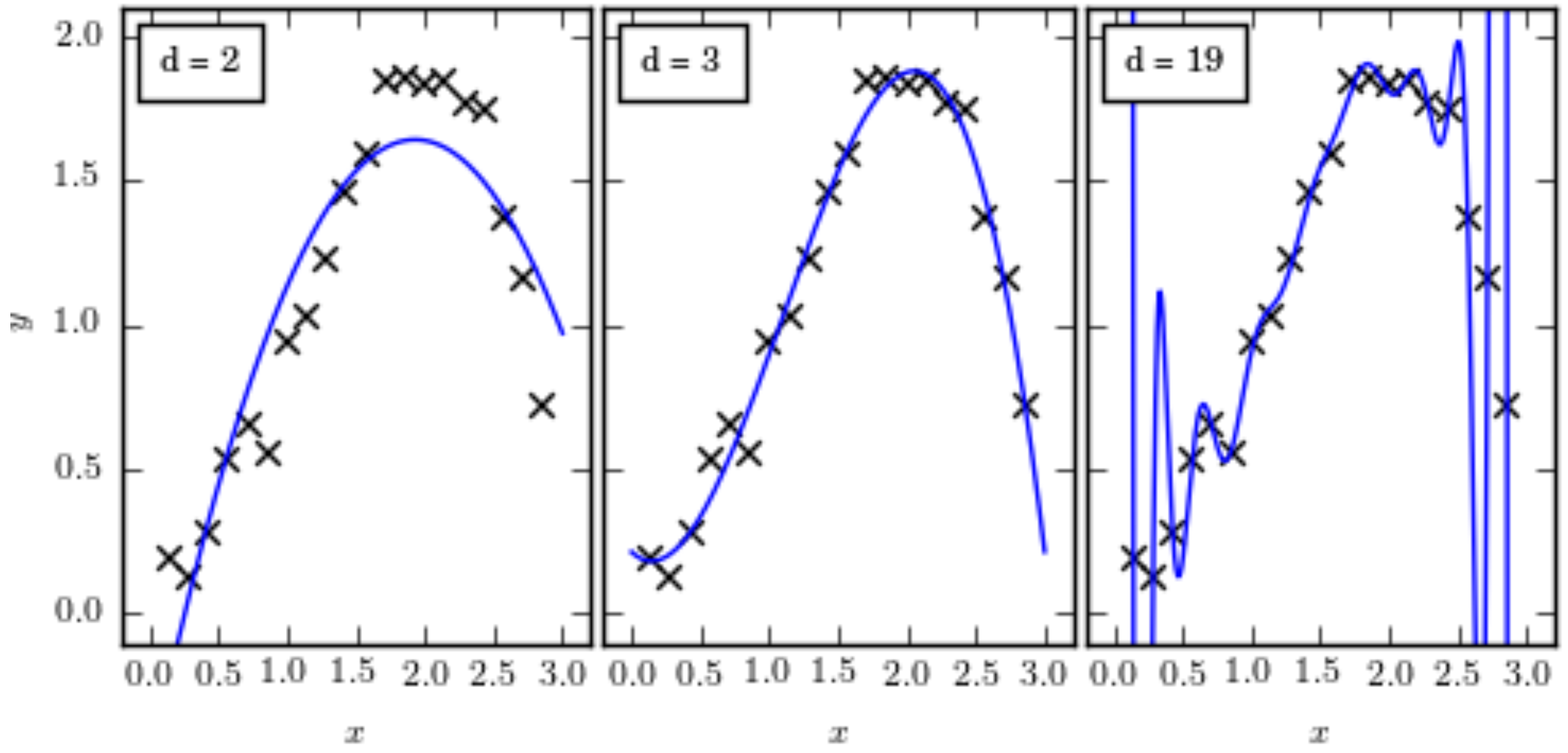
Increasing the flexibility:



Our eye would probably say a 3rd degree polynomial is ok.

From astroML.org

Increasing the flexibility:



Underfitting

Overfitting

Key ingredients of machine learning

Parameters, hyper-parameters

Consider a polynomial:

$$y = \sum_{i=0}^{M} a_i x^i$$

Here a_i are the parameters of the model and what our machine learning algorithm will give us.

But M - the maximum polynomial power is *also* a parameter. It is a different kind of parameter and we call this a **hyperparameter**.

How do you decide

We will return to this but we should think of dividing our data in three:

The training set

This is what we use to find the best-fitting parameters and we minimise the **training error** (e.g. MSE) on this.

The test set

We evaluate the generalisation error of our algorithm on this sample. These data are *not* used for the fitting. We refer to this as **test error**.

The validation set

We use this to determine the optimal settings of the hyper-parameters.

Regression

Standard linear regression

In this case we typically have p observables at each of N points (predictors) and want to predict a response variable, y_i at each x_i

A standard way to fit this is to minimise the residual sum of squares (RSS) or N times MSE:

$$\mathrm{RSS} = \sum_i (y_i - \hat{y}_i)^2$$

where $\hat{y}_i$ is the estimate of y_i. Which for linear regression is:

$$\hat{y}_i = \theta_0 + \sum_{j=1}^{p} \theta_j x_{ij}$$

Common formulation - the design matrix

The problem to solve is then often written:

$$Y = \mathrm{M}\boldsymbol{\theta}$$

Where Y is $(y_1, y_2, \ldots, y_N)$ and $\boldsymbol{\theta}=(\theta_1, \theta_2, \ldots, \theta_p)$

M is known as the design matrix as mentioned earlier and is

$$\mathrm{M} = \begin{pmatrix} 1 & x_{1,1} & x_{1,2} & \cdots & x_{1,p} \\ 1 & x_{2,1} & x_{2,2} & \cdots & x_{2,p} \\ \hdotsfor{5} \\ 1 & x_{N,1} & x_{N,2} & \cdots & x_{N,p} \end{pmatrix}$$

Common formulation - the design matrix

$$Y = \mathsf{M}\boldsymbol{\theta}$$

If we also introduce the covariance matrix of uncertainties on Y:

$$\mathsf{C} = \begin{pmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \sigma_N^2 \end{pmatrix}$$

the general solution of the linear regression is given by

$$\boldsymbol{\theta} = \left(\mathsf{M}^T\mathsf{C}^{-1}\mathsf{M}\right)^{-1}\left(\mathsf{M}^T\mathsf{C}^{-1}Y\right)$$

with uncertainties on the parameters given by

$$\Sigma_{\boldsymbol{\theta}} = \left(\mathsf{M}^T\mathsf{C}^{-1}\mathsf{M}\right)^{-1}$$

Basis functions

Note that a regression would still be linear if we transformed all the predictors with a function:

$$y_i = \theta_0 + \sum_{j=1}^{N} \theta_j \phi_j(x_i)$$

This is known as basis function regression and can for instance be done using BasisFunctionRegression in astroML.linear_model.

Linear regression - one way to do it in Python:

```
M, T = pickle_from_file('T-vs-colour-regression.pkl')
```

```
from astroML.linear_model import LinearRegression

model = LinearRegression(fit_intercept=True)
result = model.fit(M, T/1e4)
Tpred = model.predict(M)

result.coef_   # Coefficients of the fit.
```

Note that the intercept is the first element of the coefficient array. So if M is N_{obj}x4, `coef_` will be 5 elements long.

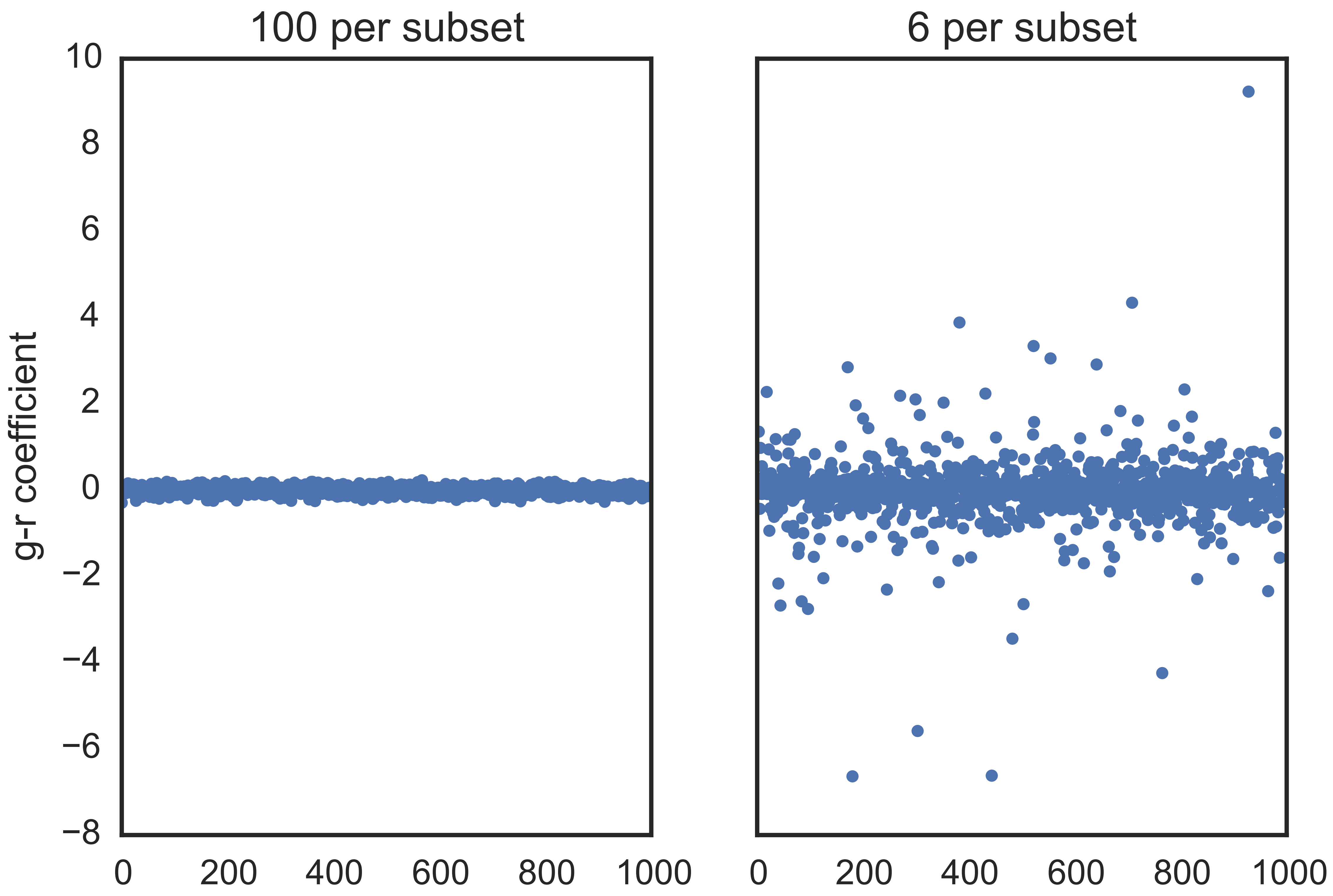
Regularising linear regression - ridge regression

When N is comparable to p, linear regression typically has large variance. To combat this we might want to trade a bit of bias for a lower variance.

As an example we can fit this:

$$T = \theta_0 + \theta_1(u-g) + \theta_2(g-r) + \theta_3(r-i) + \theta_5(i-z)$$

to the sspp dataset in astroML, where T is the effective temperature of the stars and the colours should be obvious.



Fits of 1000 random subsets with a linear regressor and showing just one coefficient.

Regularising linear regression - ridge regression

When N is comparable to p, linear regression typically has large variance. To combat this we might want to trade a bit of bias for a lower variance.

We do this by **regularising** the solution and minimise:

$$\mathrm{RSS} + \lambda \sum_{j=1}^{p} \beta_j^2$$

Here we limit the *size* of the parameter vector $\boldsymbol{\beta}$.

This does introduce a **regularisation parameter**: λ

Next time we will look at systematic ways to determine the regularisation parameter.

Regularising linear regression - ridge regression

We do this by **regularising** the solution and minimise:

$$\mathrm{RSS} + \lambda \sum_{j=1}^{p} \beta_j^2$$

Here we limit the *size* of the parameter vector $\boldsymbol{\beta}$.

When applied to linear regression, this leads to **ridge regression**.

Ridge regression - how to

```
from sklearn.linear_model import Ridge
model = Ridge(alpha=0.05, normalize=True)

result = model.fit(M, T/1e4)
Tpred = model.predict(M)

res.coef_   # Coefficients of the fit.
```

Note: alpha = λ in my (and others') notation.

Very similar to LinearRegression - with one exception:

The normalize keyword.

Ridge regression - normalisation

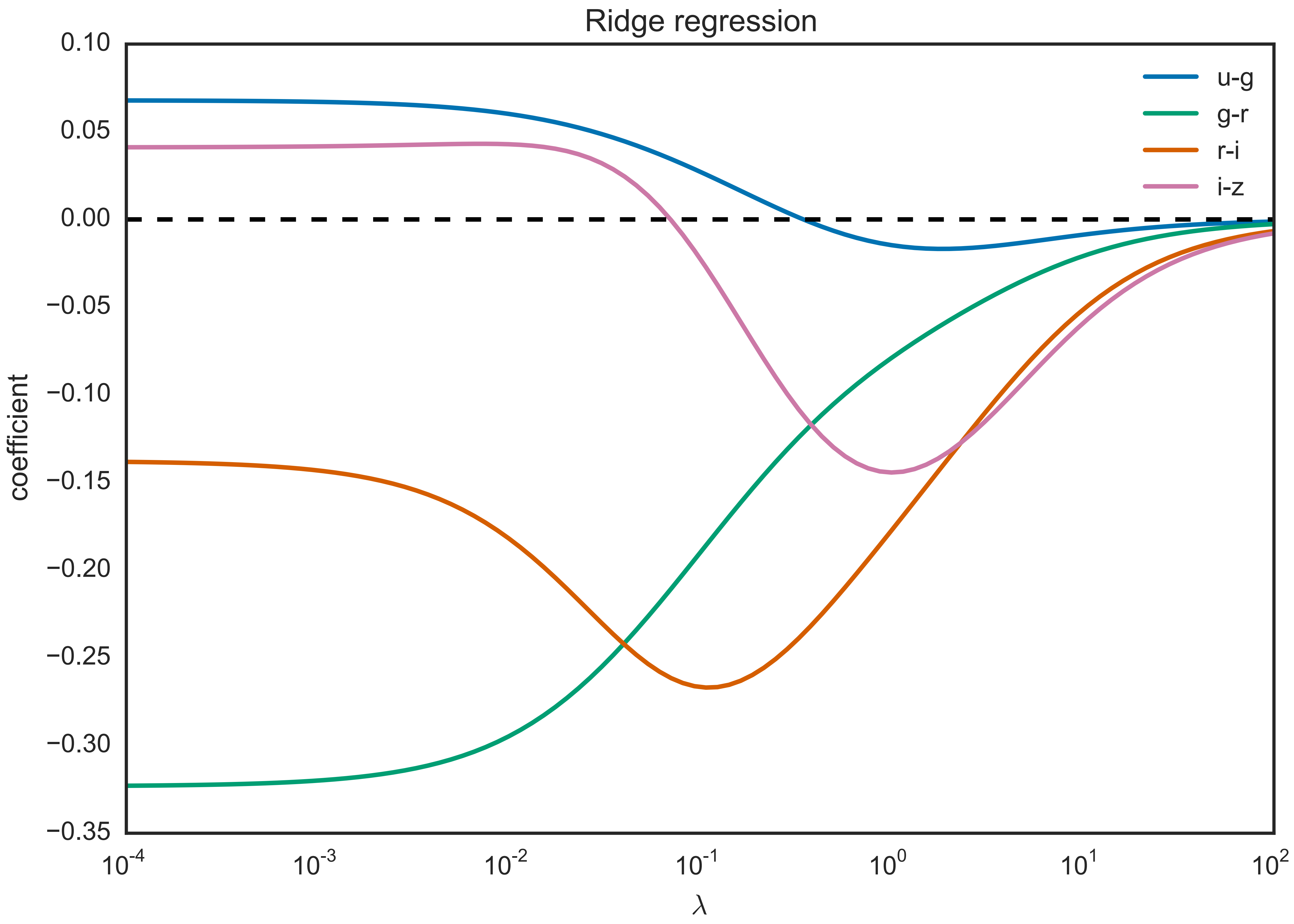
Ridge regression can also be seen to want to minimise

$$\sum_{i=1}^{N} \left(y_i - \theta_0 - \sum_{j=1}^{p} \theta_j x_{ij} \right)^2$$

subject to

$$\sum_{j=1}^{p} \theta_j^2 \leq s$$

Obviously that length will depend on the **units** of x_{ij}. It is therefore common to "**whiten**" or **standardize** x by dividing by its standard deviation (ideally robustly).


Ridge regression
coefficient
λ
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
−0.25
−0.30
−0.35
10^-4
10^-3
10^-2
10^-1
10^0
10^1
10^2
u-g
g-r
r-i
i-z

Ridge regression - degrees of freedom

There are of course p parameters, but because of the constraints in ridge regression, the effective number of degrees of freedom is not p - rather it is a smaller number depending on λ.

As far as I know this is not available through sklearn, but you can calculate it from the SVD of X:

$$\mathsf{X} = \mathsf{UDV}^T$$

If the diagonal entries in D are d_j, the d.o.f. is:

$$\mathrm{df} = \sum_{j=1}^{p} \frac{d_j^2}{d_j^2 + \lambda}$$

Regularising linear regression - LASSO

Ridge regression minimises the l_2 norm of the coefficients. The Lasso (Least Absolute Shrinkage and Selection Operator) minimises the l_1 norm.

The minimisation is now of

$$\mathrm{RSS} + \lambda \sum_{j=1}^{p} |\theta_j|$$

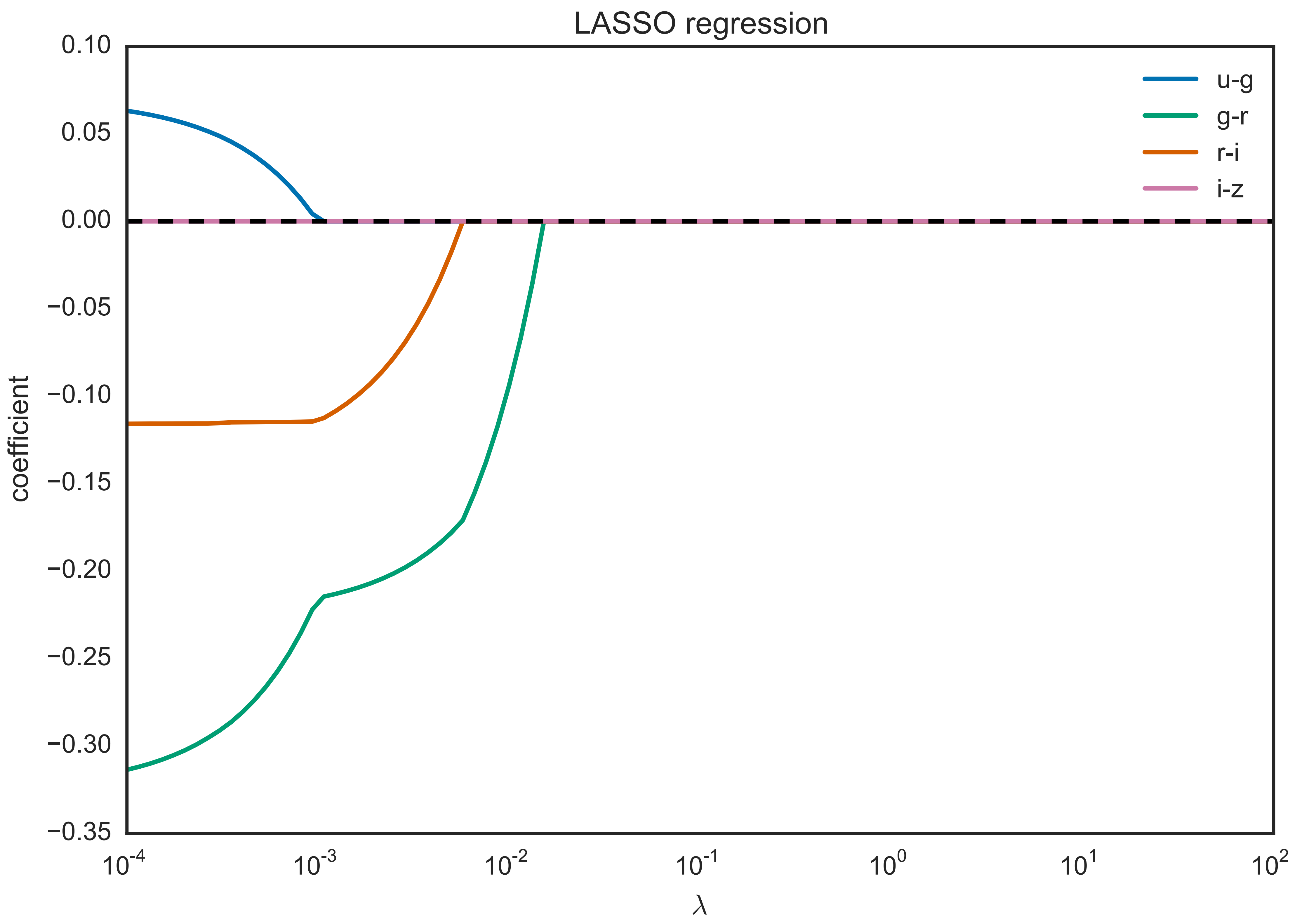
Lasso regression - how to

```
from sklearn.linear_model import Lasso
model = Lasso(alpha=alpha, normalize=True)

result = model.fit(M, T/1e4)
Tpred = model.predict(M)

res.coef_   # Coefficients of the fit.
```

So just like ridge regression - but the result is somewhat different


LASSO regression
coefficient
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
−0.25
−0.30
−0.35
10^{-4}
10^{-3}
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
λ
u-g
g-r
r-i
i-z

Variable selection with the Lasso

So some coefficients end up being set to zero!

This fact means that the lasso performs ***variable selection***.

This feature of the lasso can be phrased to say
that it returns **sparse models**.

Basically it can be interpreted to say which variables
(predictors) are most important for predicting the output.

Subset selection

The most rigorous selection of variables is what is called ***subset selection***. This basically considers all possible combinations of parameters:

```
1. Set $M_0$ to be null model with no predictors
2. for k=1,2…p
    • Fit all $\binom{p}{k}$ models that contain exactly k predictors
    • Pick the model among these that has the lowest RSS and call this $M_k$.
3. Select the best model among these M by CV or similar.
```

No ready made routine for this - but similar tools are available in sklearn.feature_selection.

So what do I do now?

Register on Blackboard

Check out the course git repository

Read the math reminder document

Go through the Python & topcat “problem set”
(try to solve it before looking at the solution suggestion)

Try out the creation of sqlite databases