Week 7: Dimensionality reduction and regression. II

Željko Ivezić and Mario Jurić, Department of Astronomy, UW

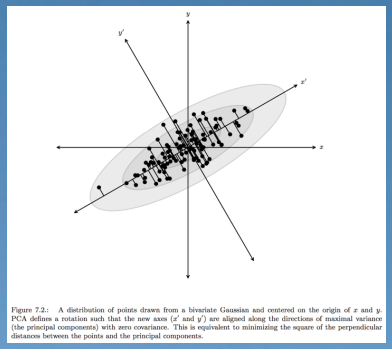
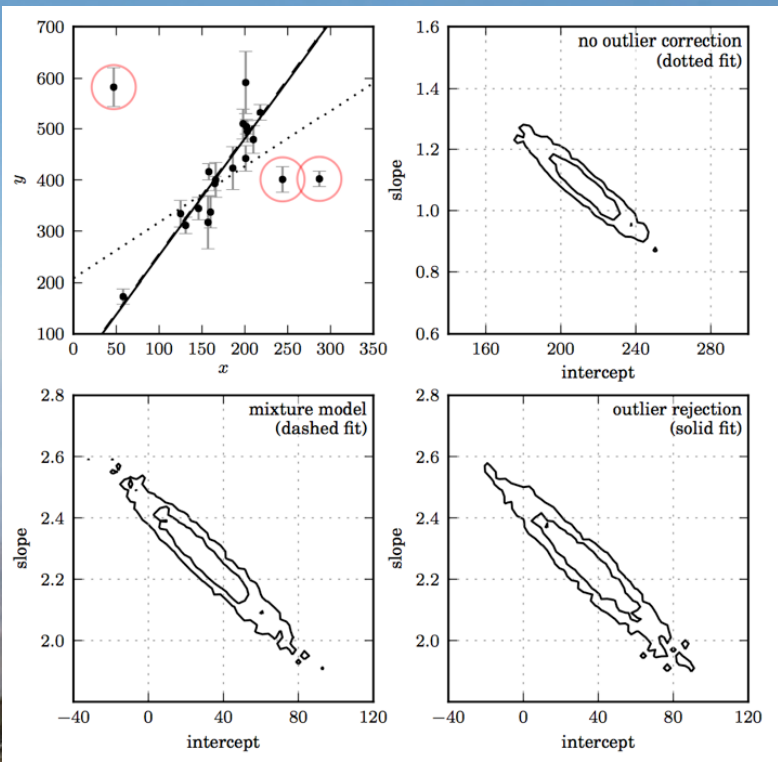


Figure 7.2.: A distribution of points drawn from a bivariate Gaussian and centered on the origin of x and y. PCA defines a rotation such that the new axes (x' and y') are aligned along the directions of maximal variance (the principal components) with zero covariance. This is equivalent to minimizing the square of the perpendicular distances between the points and the principal components.



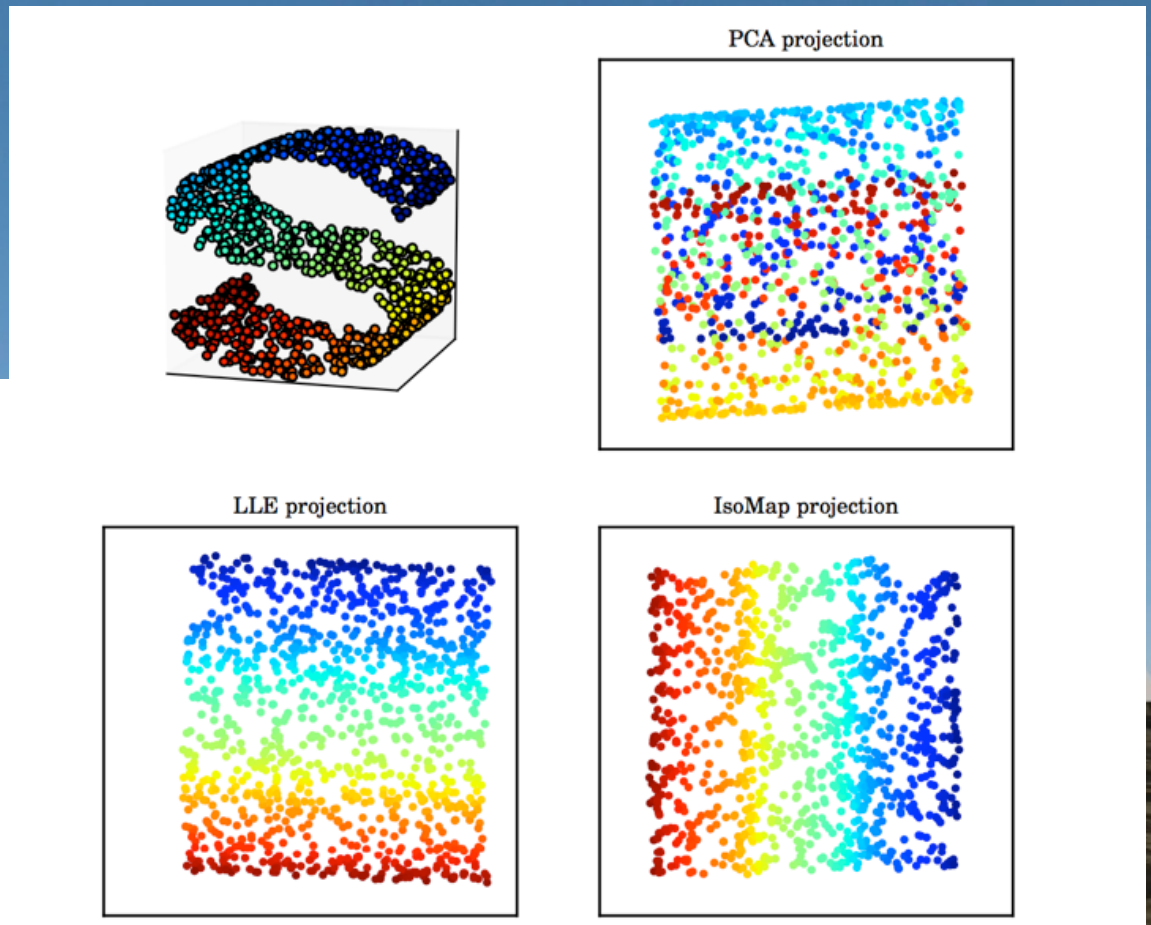


Figure 7.8.: A comparison of PCA and manifold learning. The top-left panel shows an example S-shaped data set (a two-dimensional manifold in a three-dimensional space). PCA identifies three principal components within the data. Projection onto the first two PCA components results in a mixing of the colors along the manifold. Manifold learning (LLE and IsoMap) preserves the local structure when projecting the data, preventing the mixing of the colors.

Outline

- Dimensionality Reduction
  - Covariance and Correlation
  - **Principal Component Analysis**
  - Non-negative Matrix Factorization
  - Independent Component Analysis
  - Manifold learning (Locally Linear Embedding)
- **Regression and a few misc. points**
  - (Gaussian) **errors in both variables**
  - regression with **non-Gaussian errors and/or outliers**
  - (fast matching using KD trees)

LSQ regression and non-LSQ regression

Motivation:

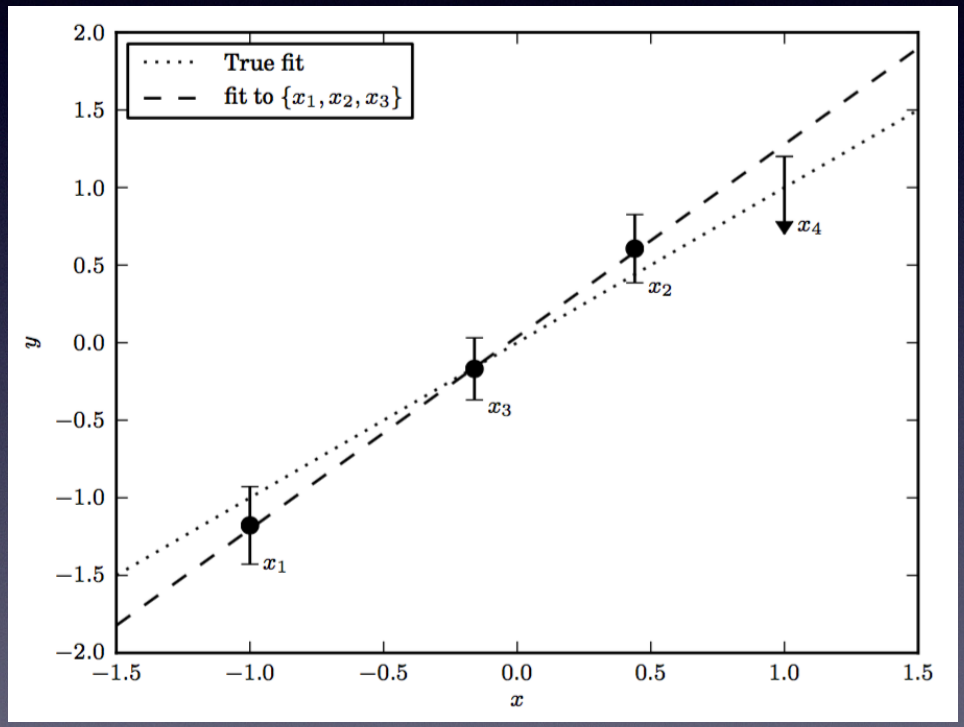
While e.g. least-square regression is often discussed, there are a few additional exceedingly useful and related tools in astroML for regression problems that often appear in practice:

– (Gaussian) errors in both variables
– regression with non-Gaussian errors and/or outliers

Recall that we addressed ordinary Least Squares Method in Week 3 Thursday lecture (on the Maximum Likelihood method)

$$p(\{y_i\}|\{x_i\},\boldsymbol{\theta},I) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(\frac{-(y_i-(\theta_0+\theta_1 x_i))^2}{2\sigma_i^2}\right)$$

The origin of "Least Squares": 2 comes from Gaussian errors



$$\theta_1 = \frac{\sum_i^N x_i y_i - \bar{x}\bar{y}}{\sum_i^N (x_i-\bar{x})^2},$$

$$\theta_0 = \bar{y} - \theta_1 \bar{x},$$

$$\sigma^2 = \sum_{i=1}^{N} (y_i - \theta_0 + \theta_1 x_i)^2,$$

$$\sigma_{\theta_1}^2 = \sigma^2 \frac{1}{\sum_i^N (x_i-\bar{x})^2},$$

$$\sigma_{\theta_0}^2 = \sigma^2 \left(\frac{1}{N} + \frac{\bar{x}^2}{\sum_i^N (x_i-\bar{x})^2}\right)$$

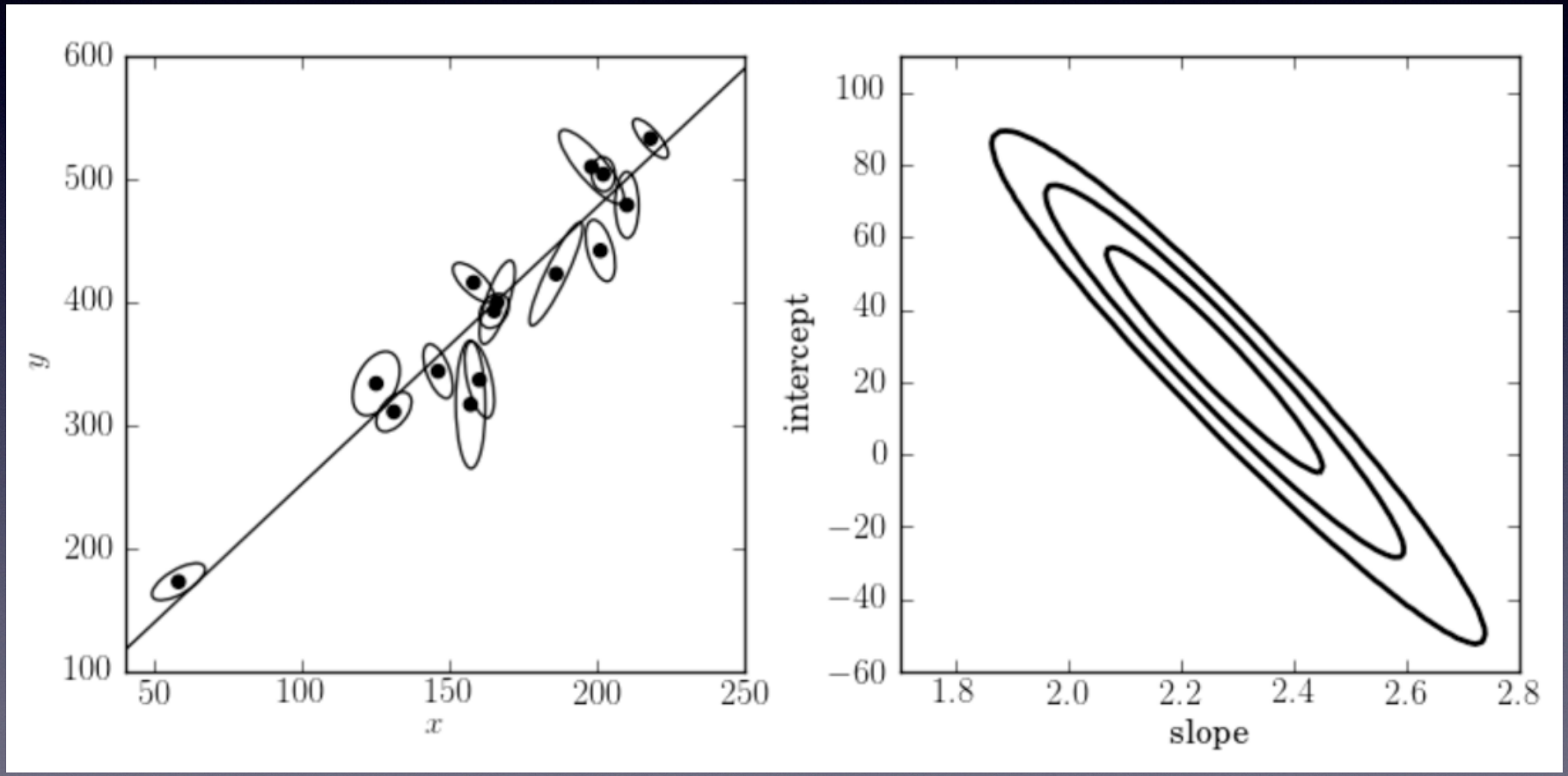
Main assumptions:
1) Gaussian errors for y
2) No errors in x

Gaussian errors in both variables

(see Hogg, Bovy & Lang, 2010, arXiv:1008.4686)

Example code:

http://www.astroml.org/book_figures/chapter8/fig_total_least_squares.html



Execute the above code and then switch to notebook

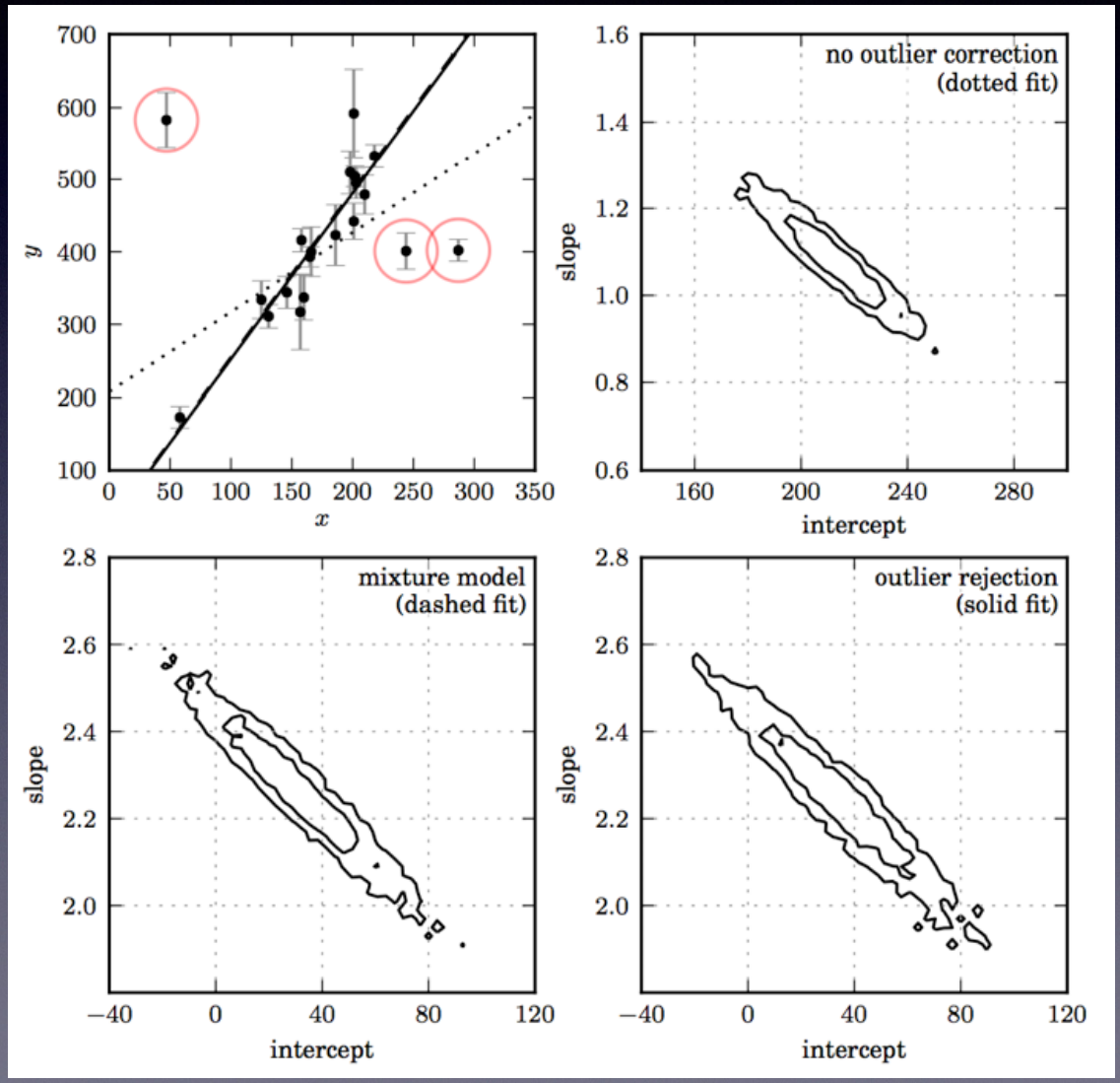
Regression with non-Gaussian errors and outliers (see Hogg, Bovy & Lang, 2010, arXiv:1008.4686)

Example code:

http://www.astroml.org/book_figures/chapter8/fig_outlier_rejection.html

The code uses MCMC

It's easy to change the outlier model (the mixture likelihood)...



Regression with non-Gaussian errors and outliers

M estimators (M stands for "maximum-likelihood-type") approach the problem of outliers by modifying the underlying likelihood estimator to be less sensitive than the classic L_2 norm. M estimators are a class of estimators that include many maximum-likelihood approaches (including least squares). They replace the standard least squares, which minimizes the sum of the squares of the residuals between a data value and the model, with a different function. Ideally the M estimator has the property that it increases less than the square of the residual and has a unique minimum at zero.

Huber loss function

An example of an M estimator that is common in robust regression is that of the Huber loss (or cost) function [9]. The Huber estimator minimizes

$$\sum_{i=1}^{N} e(y_i|y), \tag{8.65}$$

where

$$e(t) = \begin{cases} \frac{1}{2}t^2 & \text{if } |t| \leq c, \\ c|t| - \frac{1}{2}c^2 & \text{if } |t| \geq c, \end{cases} \tag{8.66}$$

Regression with non-Gaussian errors and outliers

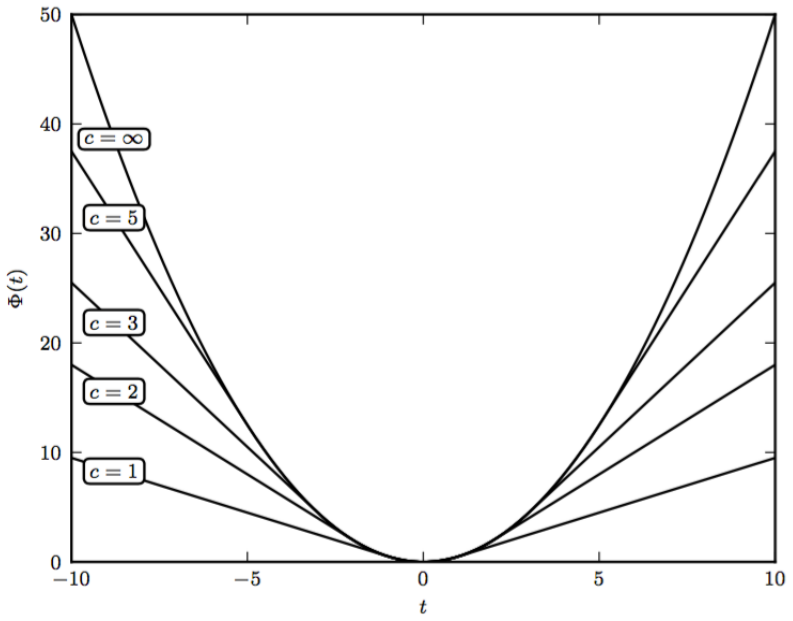


Figure 8.7.: The Huber loss function for various values of c.

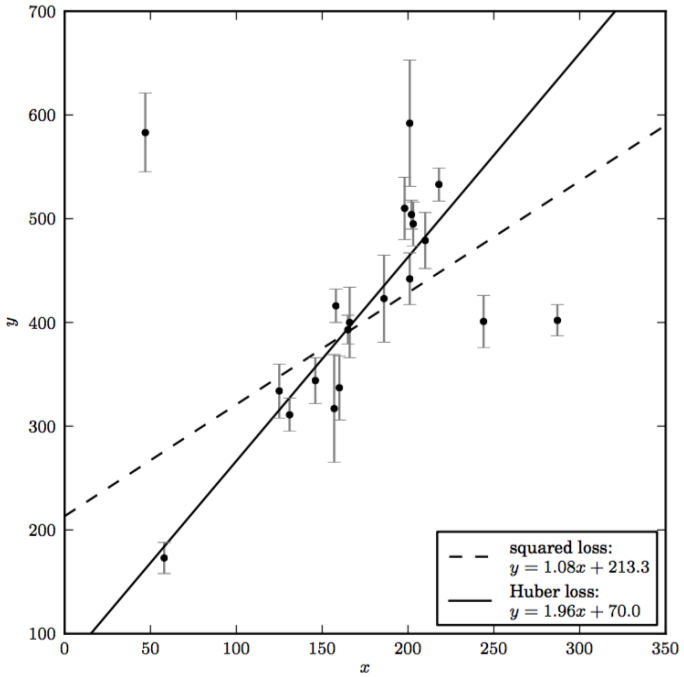


Figure 8.8.: An example of fitting a simple linear model to data which includes outliers (data is from table 1 of [8]). A comparison of linear regression using the squared-loss function (equivalent to ordinary least-squares regression) and the Huber loss function, with $c = 1$ (i.e., beyond 1 standard deviation, the loss becomes linear).

Regression with non-Gaussian errors and outliers

Bayesian mixture model:

distribution. The mixture model includes three additional parameters: μ_b and V_b, the mean and standard deviation of the background, and p_b, the probability that any point is an outlier. With this model, the likelihood becomes (cf. eq. 5.83; see also [8])

$$p(\{y_i\}|\{x_i\},\{\sigma_i\},\theta_0,\theta_1,\mu_b,V_b,p_b) \propto$$
$$\prod_{i=1}^{N}\left[\frac{1-p_b}{\sqrt{2\pi\sigma_i^2}}\exp\left(-\frac{(y_i-\theta_1 x_i-\theta_0)^2}{2\sigma_i^2}\right)+\frac{p_b}{\sqrt{2\pi(V_b+\sigma_i^2)}}\exp\left(-\frac{(y_i-\mu_b)^2}{2(V_b+\sigma_i^2)}\right)\right]. \quad (8.67)$$

Using MCMC sampling and marginalizing over the background parameters yields the dashed-line fit in figure 8.9. The marginalized posterior for this model is shown in the lower-left panel. This fit is much less affected by the outliers than is the simple regression model used above.

Finally, we can go further and perform an analysis analogous to that of §5.6.7, in which we attempt to identify bad points individually. In analogy with eq. 5.94 we can fit for nuisance parameters g_i, such that if $g_i = 1$, the point is a "good" point, and if $g_i = 0$ the point is a "bad" point. With this addition our model becomes

$$p(\{y_i\}|\{x_i\},\{\sigma_i\},\{g_i\},\theta_0,\theta_1,\mu_b,V_b) \propto$$
$$\prod_{i=1}^{N}\left[\frac{g_i}{\sqrt{2\pi\sigma_i^2}}\exp\left(-\frac{(y_i-\theta_1 x_i-\theta_0)^2}{2\sigma_i^2}\right)+\frac{1-g_i}{\sqrt{2\pi(V_b+\sigma_i^2)}}\exp\left(-\frac{(y_i-\mu_b)^2}{2(V_b+\sigma_i^2)}\right)\right]. \quad (8.68)$$

This model is very powerful: by marginalizing over all parameters but a particular g_i, we obtain a posterior estimate of whether point i is an outlier. Using this procedure, the "bad" points have been marked with a circle in the upper-left panel of figure 8.9. If instead we marginalize over the

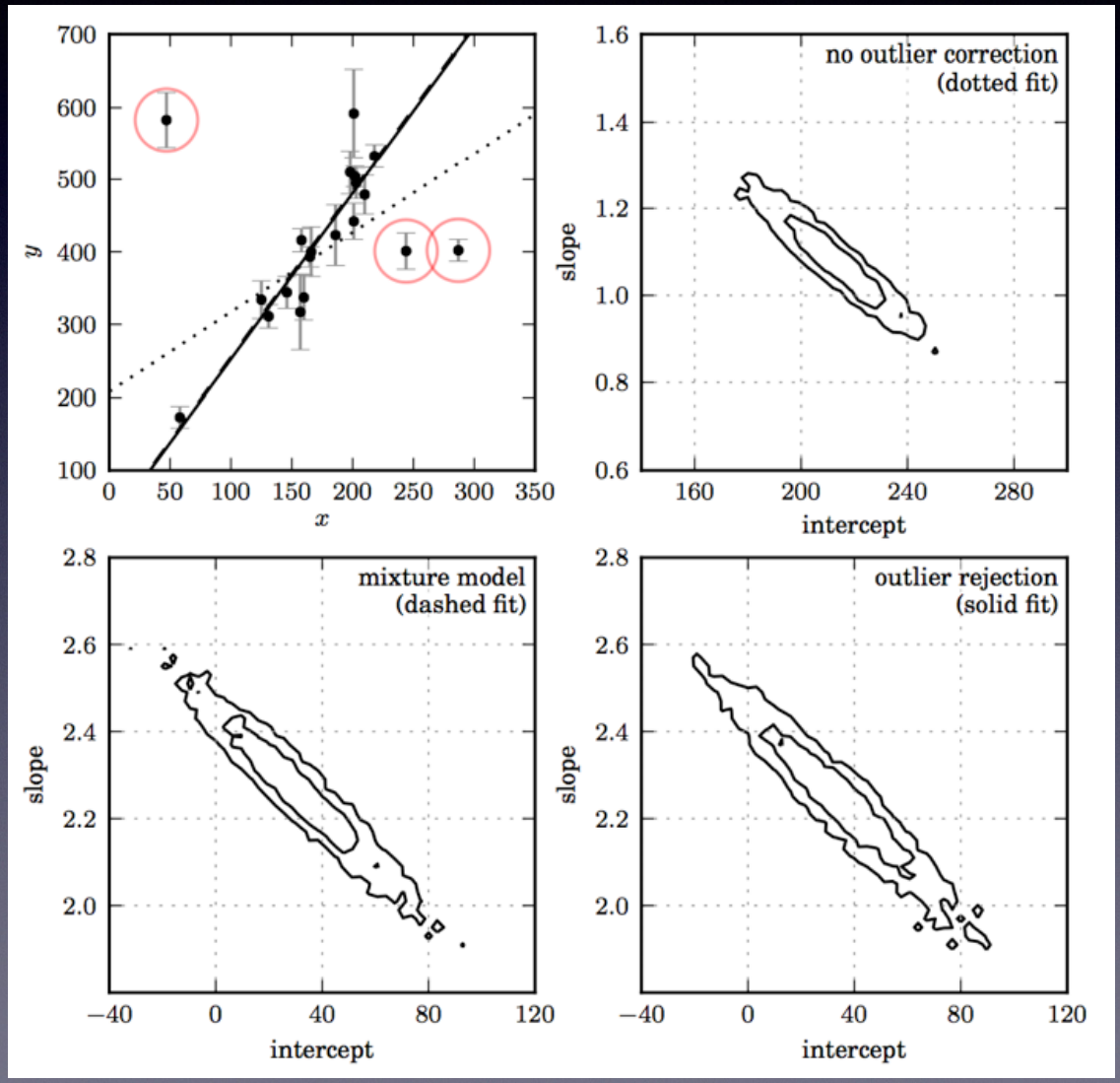
Regression with non-Gaussian errors and outliers (see Hogg, Bovy & Lang, 2010, arXiv:1008.4686)

Example code:

http://www.astroml.org/book_figures/chapter8/fig_outlier_rejection.html

The code uses MCMC

It's easy to change the outlier model (the mixture likelihood)...



Gaussian regression

- we don't have to use polynomials; we can use any set of basis functions
- Gaussian basis functions are often very convenient, e.g. for convolution
- see notebook for an example

The Gaussian distribution has two main properties that make it special. First, it lends itself to analytic treatment in many cases; most notably, a convolution of two Gaussian distributions is also Gaussian (it's hard to believe, but computers haven't existed forever). The convolution of a function $f(x)$ with a function $g(x)$ (both assumed real functions) is defined as

$$(f \star g)(x) = \int_{-\infty}^{\infty} f(x')\, g(x - x')\, dx' = \int_{-\infty}^{\infty} f(x - x')\, g(x')\, dx'. \tag{3.44}$$

In particular, the convolution of a Gaussian distribution $\mathcal{N}(\mu_o, \sigma_o)$ (e.g., an intrinsic distribution we are trying to measure) with a Gaussian distribution $\mathcal{N}(b, \sigma_e)$ (i.e., Gaussian error distribution with bias b and random error σ_e) produces parameters for the resulting Gaussian[4] $\mathcal{N}(\mu_C, \sigma_C)$ given by

$$\mu_C = (\mu_o + b) \quad \text{and} \quad \sigma_C = \left(\sigma_o^2 + \sigma_e^2\right)^{1/2}. \tag{3.45}$$

Similarly, the Fourier transform of a Gaussian is also a Gaussian (see §10.2.2). Another unique feature of the Gaussian distribution is that the sample mean and the sample variance are independent.

Fast matching using KD trees

Example code:

http://www.astroml.org/examples/algorithms/plot_crossmatch.html

Much faster than naive matching

Easy to use