Regression

Machine Learning and Pattern Recognition

Chris Williams

School of Informatics, University of Edinburgh

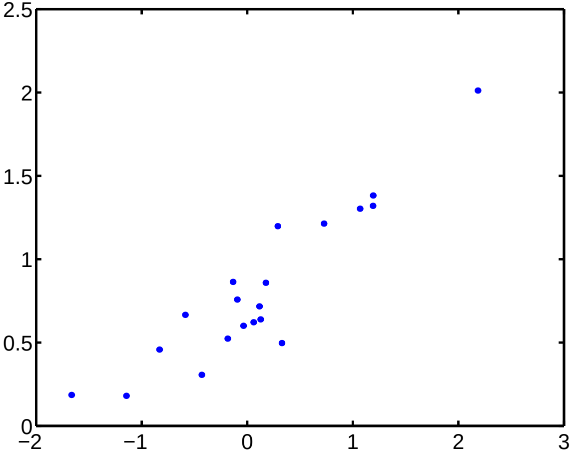
September 2014

(All of the slides in this course have been adapted from
previous versions by Charles Sutton, Amos Storkey, David Barber.)

Classification or Regression?

- Classification: want to learn a discrete target variable
- Regression: want to learn a continuous target variable
- Linear regression, linear-in-the-parameters models
  - Linear regression is a conditional Gaussian model
  - Maximum likelihood solution - ordinary least squares
  - Can use nonlinear basis functions
  - Ridge regression
  - Full Bayesian treatment
- Reading: Murphy chapter 7 (not all sections needed), Barber (17.1, 17.2, 18.1.1)

One Dimensional Data



Linear Regression

- Simple example: one-dimensional linear regression.
- Suppose we have data of the form (x, y), and we believe the data should fol low a straight line: the data should have a straight line fit of the form $y = w_0 + w_1 x$.
- However we also believe the target values y are subject to measurement error, which we will assume to be Gaussian. So $y = w_0 + w_1 x + \eta$ where η is a Gaussian noise term, mean 0, variance σ_η^2.

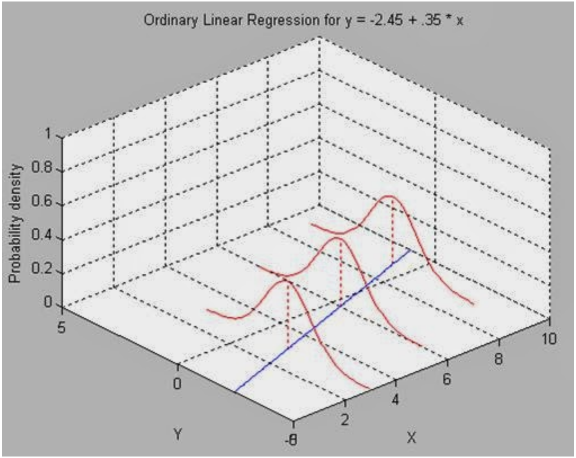
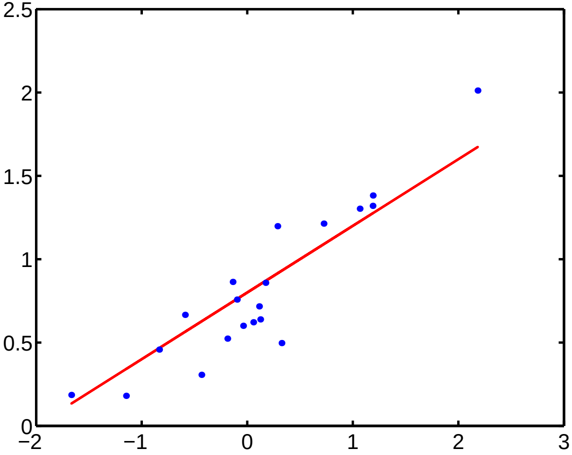


Figure credit: http://jedismedicine.blogspot.co.uk/2014/01/

- Linear regression is just a *conditional* version of estimating a Gaussian (conditional on the input x)

Generated Data



Multivariate Case

- Consider the case where we are interested in $y = f(\mathbf{x})$ for D dimensional $\mathbf{x}$: $y = w_0 + w_1 x_1 + \ldots w_D x_D + \eta$, where $\eta \sim \text{Gaussian}(0, \sigma_\eta^2)$.
- Examples? Final grade depends on time spent on work for each tutorial.
- We set $\mathbf{w} = (w_0, w_1, \ldots w_D)^T$ and introduce $\boldsymbol{\phi} = (1, \mathbf{x}^T)^T$, then we can write $y = \mathbf{w}^T \boldsymbol{\phi} + \eta$ instead
- This implies $p(y|\boldsymbol{\phi}, \mathbf{w}) = N(y; \mathbf{w}^T \boldsymbol{\phi}, \sigma_\eta^2)$
- Assume that training data is iid, i.e., $p(y^1, \ldots y^N | \mathbf{x}^1, \ldots, \mathbf{x}^N, \mathbf{w}) = \prod_{n=1}^N p(y^n | \mathbf{x}^n, \mathbf{w})$
- Given data $\{(\mathbf{x}^n, y^n), n = 1, 2, \ldots, N\}$, the log likelihood is

$$
\begin{aligned}
L(\mathbf{w}) &= \log P(y^1 \ldots y^N | \mathbf{x}^1 \ldots \mathbf{x}^N, \mathbf{w}) \\
&= -\frac{1}{2\sigma_\eta^2} \sum_{n=1}^N (y^n - \mathbf{w}^T \boldsymbol{\phi}^n)^2 - \frac{N}{2} \log(2\pi\sigma_\eta^2)
\end{aligned}
$$

Minimizing Squared Error

$$
\begin{aligned}
\mathcal{L}(\mathbf{w}) &= -\frac{1}{2\sigma_\eta^2}\sum_{n=1}^{N}(y^n - \mathbf{w}^T\boldsymbol{\phi}^n)^2 - \frac{N}{2}\log(2\pi\sigma_\eta^2) \\
&= -C_1\sum_{n=1}^{N}(y^n - \mathbf{w}^T\boldsymbol{\phi}^n)^2 - C_2
\end{aligned}
$$

where $C_1 > 0$ and C_2 don't depend on $\mathbf{w}$. Now

- Multiplying by a positive constant doesn't change the maximum
- Adding a constant doesn't change the maximum.
- $\sum_{n=1}^{N}(y^n - \mathbf{w}^T\boldsymbol{\phi}^n)^2$ is the sum of squared errors made if you use $\mathbf{w}$

So *maximizing* the likelihood is the same as *minimizing* the total squared error of the linear predictor.

So you don't have to believe the Gaussian assumption. You can simply believe that you want to minimize the squared error.

Maximum Likelihood Solution I

- Write $\Phi = (\boldsymbol{\phi}^1, \boldsymbol{\phi}^2, \ldots, \boldsymbol{\phi}^N)^T$, and $\mathbf{y} = (y^1, y^2, \ldots, y^N)^T$
- Φ is called the *design matrix*, has N rows, one for each example

$$L(\mathbf{w}) = -\frac{1}{2\sigma_\eta^2}(\mathbf{y} - \Phi\mathbf{w})^T(\mathbf{y} - \Phi\mathbf{w}) - C_2$$

- Take derivatives of the log likelihood:

$$\nabla_{\mathbf{w}} L(\mathbf{w}) = -\frac{1}{\sigma_\eta^2}\Phi^T(\Phi\mathbf{w} - \mathbf{y})$$

Maximum Likelihood Solution II

- Setting the derivatives to zero to find the minimum gives

$$\Phi^T \Phi \hat{\mathbf{w}} = \Phi^T \mathbf{y}$$

- This means the maximum likelihood $\hat{\mathbf{w}}$ is given by

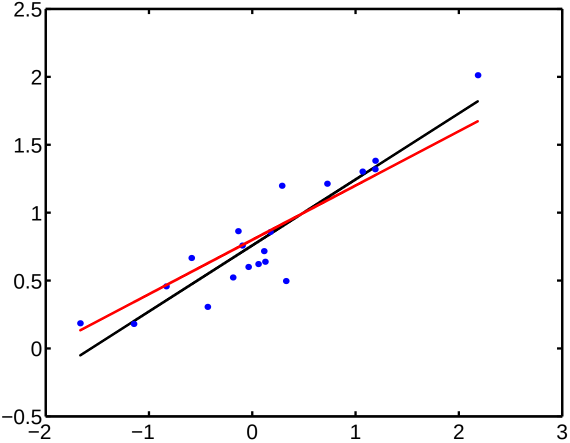
$$\hat{\mathbf{w}} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{y}$$

  The matrix $(\Phi^T \Phi)^{-1} \Phi^T$ is called the *pseudo-inverse*.
- Ordinary least squares (OLS) solution for $\mathbf{w}$
- MLE for the variance

$$\hat{\sigma}_\eta^2 = \frac{1}{N} \sum_{n=1}^{N} (y^n - \mathbf{w}^T \boldsymbol{\phi}^n)^2$$

  i.e. the average of the squared *residuals*

Generated Data



The black line is the maximum likelihood fit to the data.

Nonlinear regression

- All this just used ϕ.
- We chose to put the $\mathbf{x}$ values in ϕ, but we could have put anything in there, including nonlinear transformations of the $\mathbf{x}$ values.
- In fact we can choose any useful form for ϕ so long as the final derivatives are linear wrt $\mathbf{w}$. We can even change the size.
- We already have the maximum likelihood solution in the case of Gaussian noise: the pseudo-inverse solution.
- Models of this form are called general linear models or linear-in-the-parameters models.

Example:polynomial fitting

- Model $y = w_1 + w_2 x + w_2 x^2 + w_4 x^3$.
- Set $\boldsymbol{\phi} = (1, x, x^2, x^3)^T$ and $\mathbf{w} = (w_1, w_2, w_3, w_4)$.
- Can immediately write down the ML solution: $\mathbf{w} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{y}$, where Φ and $\mathbf{y}$ are defined as before.
- Could use any features we want: e.g. features that are only active in certain local regions (radial basis functions, RBFs).

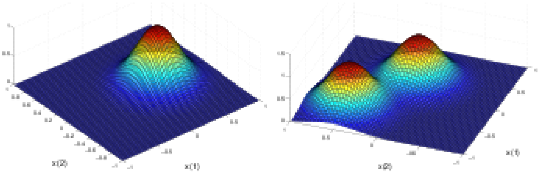


Figure credit: David Barber, BRML Fig 17.6

Dimensionality issues

- How many radial basis functions do we need?
- Suppose we need only three per dimension
- Then we would need 3^D for a D-dimensional problem
- This becomes large very fast: this is commonly called the *curse of dimensionality*
- Gaussian processes (see later) can help with these issues

Higher dimensional outputs

- Suppose the target values are vectors.
- Then we introduce different $\mathbf{w}_i$ for each y_i.
- Then we can do regression independently in each of those cases.

Adding a Prior

- Put prior over parameters, e.g.,

$$p(y|\boldsymbol{\phi}, \mathbf{w}) = N(y; \mathbf{w}^T\boldsymbol{\phi}, \sigma_\eta^2)$$
$$p(\mathbf{w}) = N(\mathbf{w}; 0, \tau^2 I)$$

- I is the identity matrix
- The log posterior is

$$\log p(\mathbf{w}|\mathcal{D}) = \text{const} - \frac{1}{2\sigma_\eta^2}\sum_{n=1}^{N}(y^n - \mathbf{w}^T\boldsymbol{\phi}^n)^2 - \frac{N}{2}\log(2\pi\sigma^2) - \underbrace{\frac{1}{2\tau^2}\mathbf{w}^T\mathbf{w}}_{\text{penalty on large weights}} - \frac{D}{2}\log(2\pi\tau^2)$$

- MAP solution can be computed analytically. Derivation almost the same as with MLE (where $\lambda = \sigma_\eta^2/\tau^2$)

$$\mathbf{w}_{MAP} = (\Phi^T\Phi + \lambda I)^{-1}\Phi^T\mathbf{y}$$

This is called *ridge regression*

Effect of Ridge Regression

- Collecting constant terms from log posterior on last slide

$$\log p(\mathbf{w}|\mathcal{D}) = \text{const} - \frac{1}{2\sigma_\eta^2}\sum_{n=1}^{N}(y^n - \mathbf{w}^T\boldsymbol{\phi}^n)^2 - \underbrace{\frac{1}{2\tau^2}\mathbf{w}^T\mathbf{w}}_{||\mathbf{w}||_2^2.\ \text{penalty term}}$$

- This is called ℓ_2 *regularization* or *weight decay*. The second term is the squared Euclidean (also called ℓ_2) norm of $\mathbf{w}$.
- The idea is to reduce overfitting by forcing the function to be simple. The simplest possible function is constant $\mathbf{w} = 0$, so encourage $\hat{\mathbf{w}}$ to be closer to that.
- τ is a parameter of the method. Trades off between how well you fit the training data and how simple the method is. Most commonly set via cross validation.
- Regularization is a general term for adding a "second term" to an objective function to encourage simple models.

Effect of Ridge Regression (Graphic)

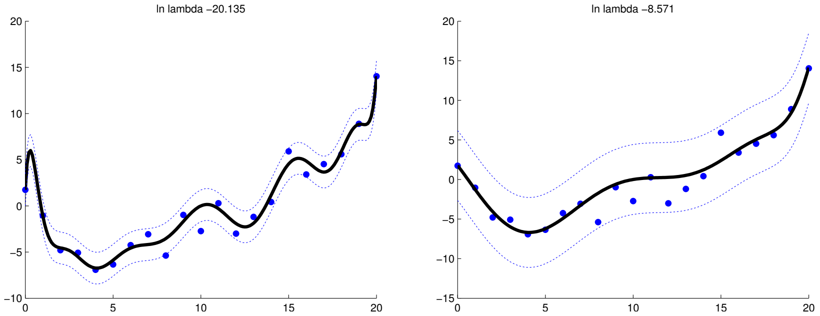


Figure credit: Murphy Fig 7.7

Degree 14 polynomial fit with and without regularization

Why Ridge Regression Works (Graphic)

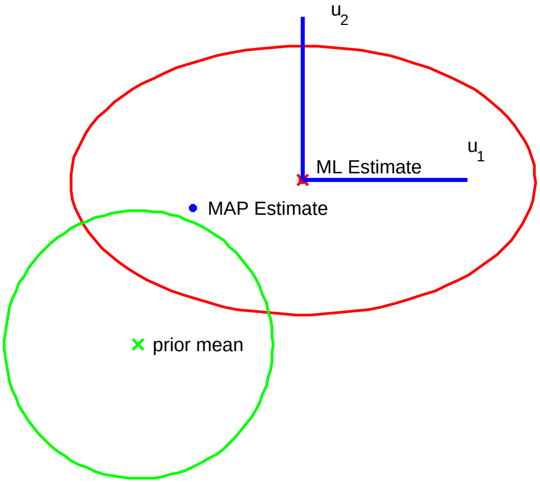


Figure credit: Murphy Fig 7.9

Bayesian Regression

- Bayesian regression model

$$p(y|\boldsymbol{\phi}, \mathbf{w}) = N(y; \mathbf{w}^T \boldsymbol{\phi}, \sigma_\eta^2)$$
$$p(\mathbf{w}) = N(\mathbf{w}; 0, \tau^2 I)$$

- Possible to compute the posterior distribution analytically, because linear Gaussian models are jointly Gaussian (see Murphy §7.6.1 for details)

$$p(\mathbf{w}|\Phi, \mathbf{y}, \sigma_\eta^2) \propto p(\mathbf{w})p(\mathbf{y}|\Phi, \sigma_\eta^2) = N(\mathbf{w}|\mathbf{w}_N, V_N)$$
$$\mathbf{w}_N = \frac{1}{\sigma_\eta^2} V_N \Phi^T \mathbf{y}$$
$$V_N = \sigma_\eta^2 (\sigma_\eta^2 / \tau^2 I + \Phi^T \Phi)^{-1}$$

Making predictions

- For a new test point $\mathbf{x}^*$ with corresponding feature vector $\boldsymbol{\phi}^*$, we have that

$$f(\mathbf{x}^*) = \mathbf{w}^T \boldsymbol{\phi}^* + \eta$$

  where $\mathbf{w} \sim N(\mathbf{w}_N, V_N)$.

- Hence

$$p(y^* | \mathbf{x}^*, \mathcal{D}) \sim N(\mathbf{w}_N^T \boldsymbol{\phi}^*, (\boldsymbol{\phi}^*)^T V_N \boldsymbol{\phi}^* + \sigma_\eta^2)$$

Example of Bayesian Regression

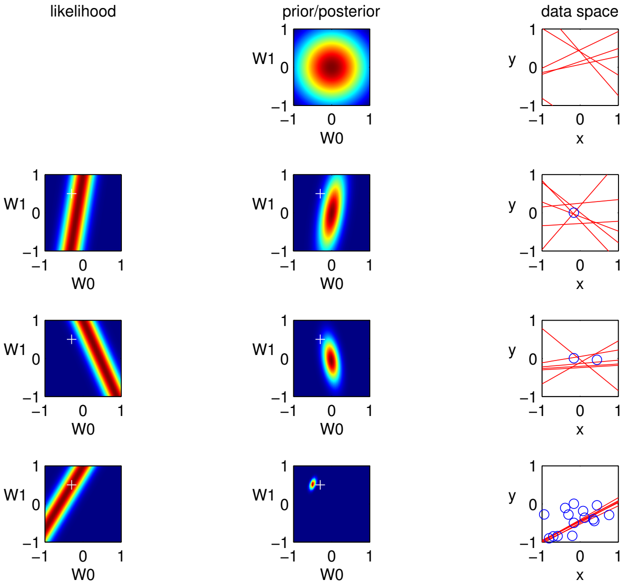


Figure credit: Murphy Fig 7.11

Another Example

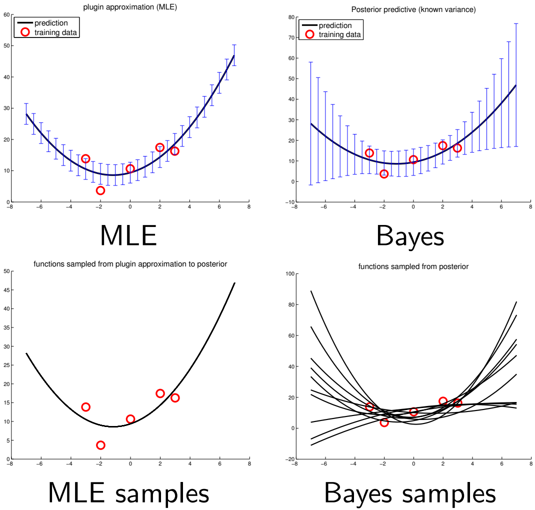


Figure credit: Murphy Fig 7.12

Fitting a quadratic. Notice how the error bars get larger further away from training data

Summary

- Linear regression is a conditional Gaussian model
- Maximum likelihood solution - ordinary least squares
- Can use nonlinear basis functions
- Ridge regression
- Full Bayesian treatment