Linear regression from a machine learning perspective

Refs: Nando de Freitas (Youtube)
"Machine learning. A probabilistic perspective"; Kevin P. Murphy.

Outline:
- Intro to supervised learning
- Linear prediction
- A probabilistic interpretation
- Regularization
- Beyond linear models
- Bayesian linear regression

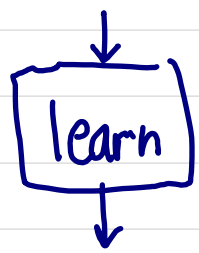
Supervised learning

We are given a training dataset with n instances of input-output pairs

$$\{(\bar{x}_1, \bar{y}_1), (\bar{x}_2, \bar{y}_2), (\bar{x}_3, \bar{y}_3), \ldots, (\bar{x}_n, \bar{y}_n)\}$$

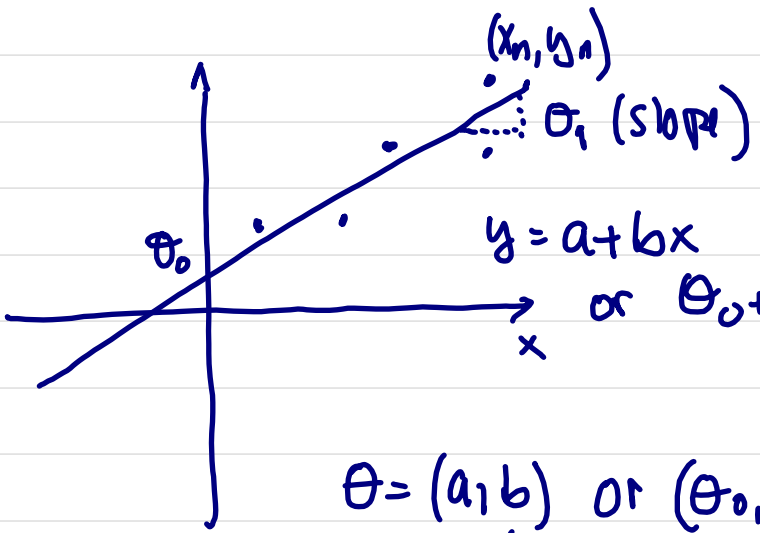
where $\bar{x}_i$ are d-vectors $\bar{x}_i = [x_{i1}, x_{i2}, \ldots x_{id}]$
The inputs are also referred-to as "predictors" or "covariates", the output are the "targets" and we assume are real numbers
We want to learn a model that given d inputs $\bar{x}_{n+1}$ can predict an output: $y(x_{n+1})$

① TRAINING $\{(\bar{x}_1, y_1), \ldots (\bar{x}_n, y_n)\}$



parameters $\hat{\theta}$ of a linear model

Example : 2D



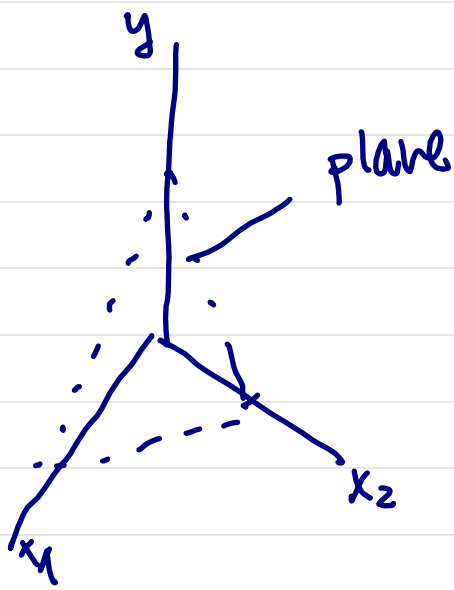
$y = a + bx$

or $\theta_0 + \theta_1 x_1$

$\theta = (a, b)$ or (θ_0, θ_1)

$\bar{x} = (1, x)$

3D



$y = a + bx_1 + cx_2$

or $y = \theta_0 + \theta_1 x_1 + \theta_2 x_2$

$\bar{\theta} = (\theta_0, \theta_2, \theta_3)$

$\bar{x} = (1, x_1, x_2)$

② PREDICTION

$\bar{x}_{n+1}$, $\hat{\theta}$ → [predict] → $\hat{y}_{n+1}$

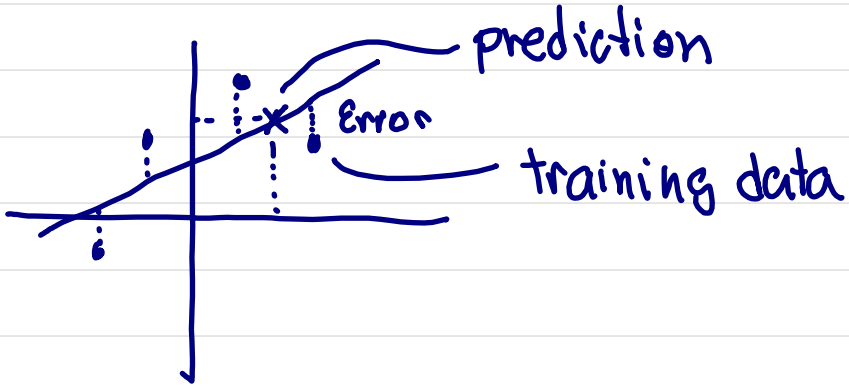
(the "hat" indicates "prediction")

Example: 2D $y(x_i) = \theta_0 + \theta_1 x_i = y_i$ model

we need to define a "cost function"

$$J(\bar{\theta}) = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} (y_i - \theta_0 - \theta_1 x_i)^2$$

↑ "error"



Linear prediction

In general $\hat{y}_i = \sum_{j=0}^{d} \theta_j x_{ij}$

Typically one assumes $x_{i0} = 1$ so that θ_0 is the constant term or "bias" or "offset", ε

$$\hat{y}_i = \sum_{j=1}^{d} \theta_j x_{ij} + \underbrace{\varepsilon}_{\theta_0}$$

In matrix form:

$$\hat{\bar{y}} = \bar{\bar{X}}\,\bar{\theta}$$

$$\begin{bmatrix} \hat{y}_0 \\ \hat{y}_1 \\ \vdots \\ \hat{y}_n \end{bmatrix} = \begin{bmatrix} x_{00} & \cdots & x_{0d} \\ \vdots & & \vdots \\ x_{n0} & & x_{nd} \end{bmatrix} \begin{bmatrix} \theta_0 \\ \vdots \\ \theta_d \end{bmatrix}$$

$n \times 1$ $\quad$ $n \times (d+1)$ $\quad$ $(d+1) \times 1$

If we know $\bar{\theta}$, we multiply $\bar{\bar{X}}$ times $\bar{\theta}$ and we get the prediction for the training dataset.

What if we have a new point that is not in the training dataset?

$$\hat{y} = \bar{x}^T \bar{\theta}$$

$$\hat{y} = [x_{0_1} \cdots x_d] \begin{bmatrix} \theta_0 \\ \vdots \\ \theta_d \end{bmatrix}$$

How do we obtain $\bar{\theta}$?

Optimization

$$J(\theta) = (\bar{y} - \bar{\bar{X}}\bar{\theta})^T (\bar{y} - \bar{\bar{X}}\bar{\theta}) = \sum_{i=1}^{n} (y_i - \bar{x}_i^T \bar{\theta})^2$$

$$= \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Minimization: $\frac{\partial J}{\partial \theta_i} = 0$; or

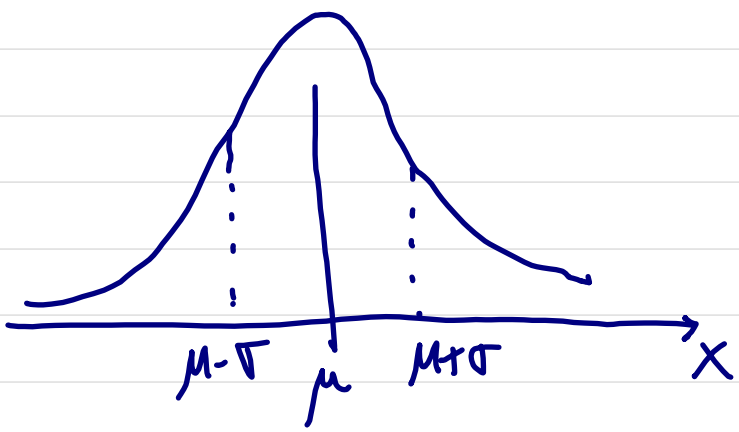
we can use $\frac{\partial \bar{\bar{A}}\bar{\theta}}{\partial \bar{\theta}} = \bar{\bar{A}}^t$; $\frac{\partial \bar{\theta}^T \bar{\bar{A}} \bar{\theta}}{\partial \bar{\theta}} = 2\bar{\bar{A}}^T \bar{\theta}$

$$\rightarrow \boxed{\bar{\theta} = (\bar{\bar{X}}^T \bar{\bar{X}})^{-1} \bar{\bar{X}}^T \bar{y}}$$

Univariate Gaussian distribution

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2} \qquad x = \mathcal{N}(\mu, \sigma^2)$$

μ: mean
σ: variance



p is a probability density: $\int_{-\infty}^{\infty} p(x)\,dx = 1$

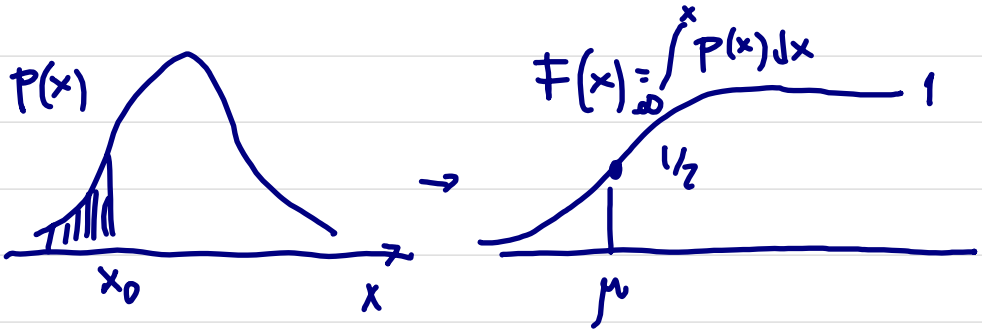
$p(x) \geq 0$

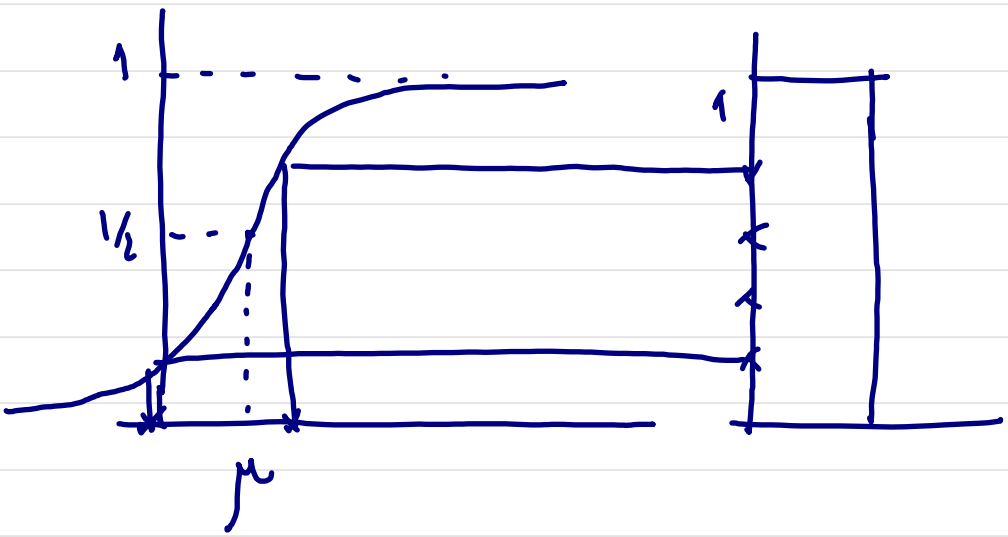
Samplins from a Gaussian dist:

$$x \sim \mathcal{N}(\mu, \sigma^2)$$

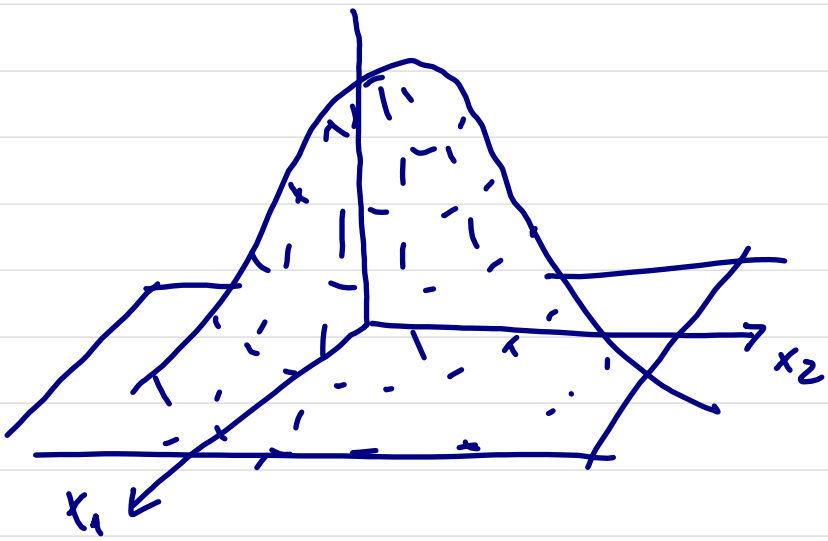
See github.

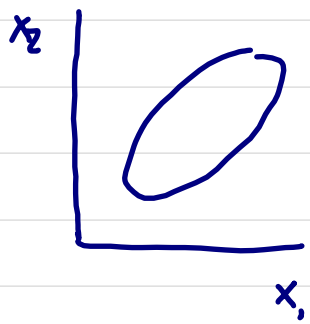
Cummulative distribution

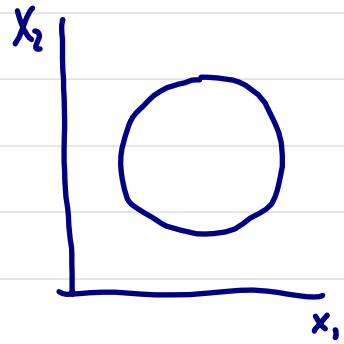




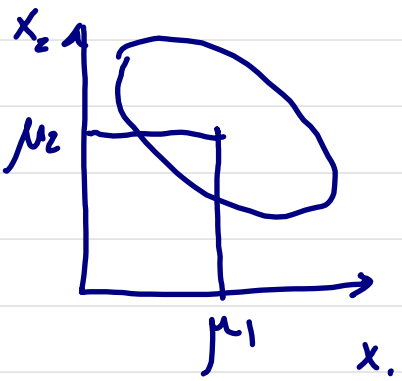
the multi-variate Gaussian distribution







$$\bar{\mu} = \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_n \end{pmatrix} = \begin{pmatrix} \mathbb{E}(x_1) \\ \vdots \\ \mathbb{E}(x_n) \end{pmatrix}$$

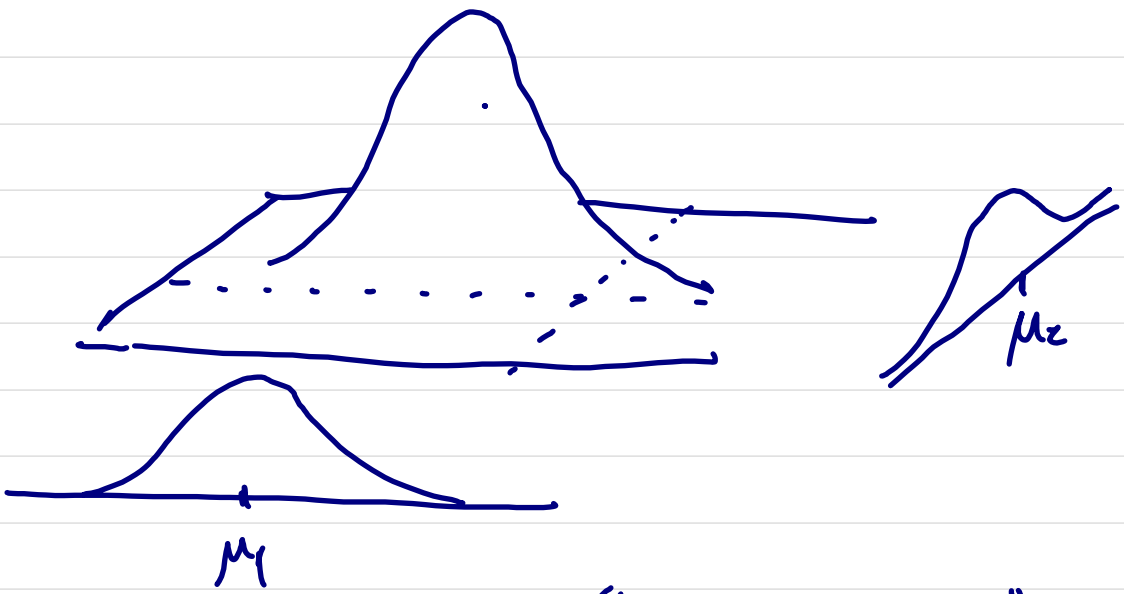


$$\bar{\bar{\Sigma}} = \begin{pmatrix} \sigma_{11} & \cdots & \sigma_{1n} \\ & & \\ \sigma_{n1} & \cdots & \sigma_{nn} \end{pmatrix}$$

$$= \mathbb{E}\left[(\bar{x} - \bar{\mu})(\bar{x} - \bar{\mu})^T \right]$$

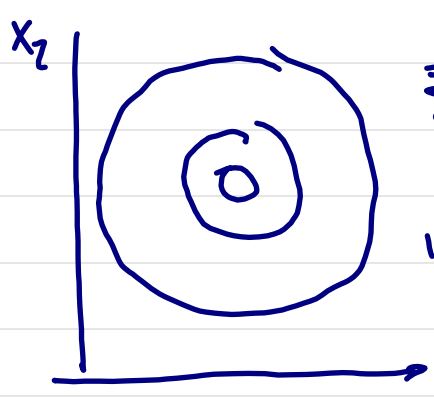
$$p(\bar{x}) = \left| 2\pi \bar{\bar{\Sigma}} \right|^{-1/2} e^{-1/2 (\bar{x} - \bar{\mu})^T \bar{\bar{\Sigma}}^{-1} (\bar{x} - \bar{\mu})}$$

Example : bi-variate distribution



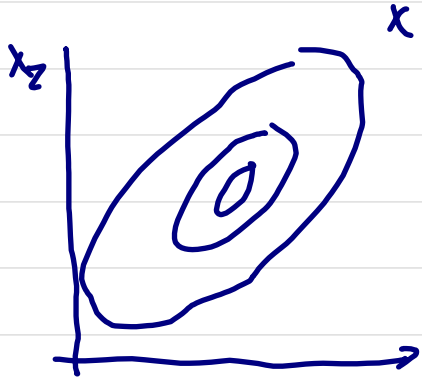
"cross correlation"

$$\overline{\overline{\Sigma}} = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{bmatrix}$$ covariance matrix



$$\overline{\overline{\Sigma}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

independent variables



$$\overline{\overline{\Sigma}} = \begin{bmatrix} 1 & 1/2 \\ 1/2 & 1 \end{bmatrix}$$

Example: two independent Gaussian variables

$$x_1 = \mathcal{N}(\mu_1, \sigma^2) \quad \text{and} \quad x_2 = \mathcal{N}(\mu_2, \sigma^2)$$

Their joint distribution is

$$P(x_1, x_2) = P(x_1)P(x_2)$$

$$= \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma}(x_1-\mu_1)^2} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma}(x_2-\mu_2)^2}$$

$$= \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2}\begin{bmatrix}(x-\mu_1) & (x-\mu_2)\end{bmatrix}\underbrace{\begin{bmatrix}\sigma^2 & 0 \\ 0 & \sigma^2\end{bmatrix}^{-1}}_{\Sigma}\begin{bmatrix}x-\mu_1 \\ x-\mu_2\end{bmatrix}}$$

→ The covariance matrix is diagonal

Sampling from a multivariate Gaussian dist.

We want to draw a vector $\bar{x} \sim \mathcal{N}(\bar{\mu}, \bar{\bar{\Sigma}})$

We need to carry out a Cholesky decomposition

$$\bar{\bar{\Sigma}} = \bar{\bar{B}}^{t} \bar{\bar{B}}$$

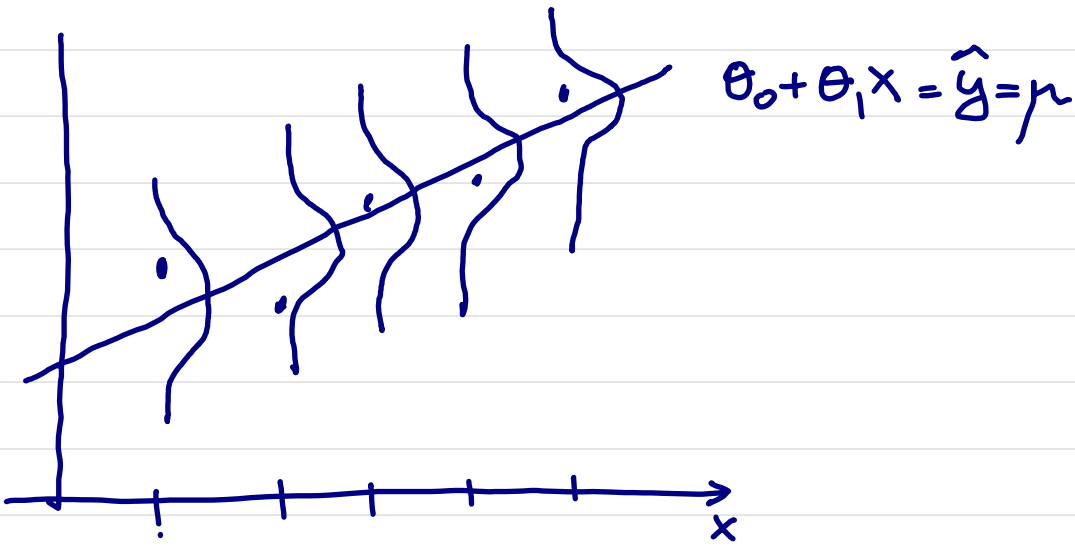
$$\bar{x} = \bar{\mu} + \bar{\bar{B}}\, \mathcal{N}(0, I)$$

In 1d

$$x = \mu + \sigma \mathcal{N}(0,1)$$

The likelihood for linear regression

We assume that $\bar{y}_i$ is Gaussian distributed with mean $\bar{x}_i^T\bar{\theta}$ and variance σ



$$\bar{y}_i = \mathcal{N}(\bar{x}_i^T\bar{\theta}, \sigma^2) = \bar{x}_i^T\bar{\theta} + \mathcal{N}(0, \sigma^2)$$

Since the variables are independent

$$p(\bar{y}|\bar{X}\bar{\theta}\sigma) = \prod_{i=1}^{n} p(y_i|\bar{x}_i\bar{\theta}\sigma)$$

↑ prob. of $\bar{y}$ given $\bar{X}, \bar{\theta}, \sigma$

$$= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-1}{2\sigma^2}(y_i - \bar{x}_i^T\bar{\theta})^2}$$

$$= (2\pi\sigma^2)^{-n/2} e^{-\frac{1}{2\sigma^2}\sum_{n=1}^{n}(y_i - \bar{x}_i^T\bar{\theta})^2}$$

$$e_i = (y_i - \hat{y}_i)^2$$

Maximum likelihood

The maximum likelihood estimate (MLE) of $\bar{\theta}$ is obtained by taking the derivatives of the log-likelihood

$$P(y|\bar{\bar{X}}\bar{\theta}\sigma) = (2\pi\sigma^2)^{-n/2} e^{-\frac{1}{2\sigma^2}(\bar{y}-\bar{\bar{X}}\theta)^T(\bar{y}-\bar{\bar{X}}\bar{\theta})}$$

(we take logs to avoid exponentials)

$$\log(P) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}(\bar{y}-\bar{\bar{X}}\theta)^T(\bar{y}-\bar{\bar{X}}\theta)$$

$$\frac{\partial \log(P)}{\partial \theta} = 0 - \frac{1}{2\sigma^2}\left[0 - 2\bar{\bar{X}}^T\bar{y} + \bar{\bar{X}}^T\bar{\bar{X}}\theta\right] = 0$$

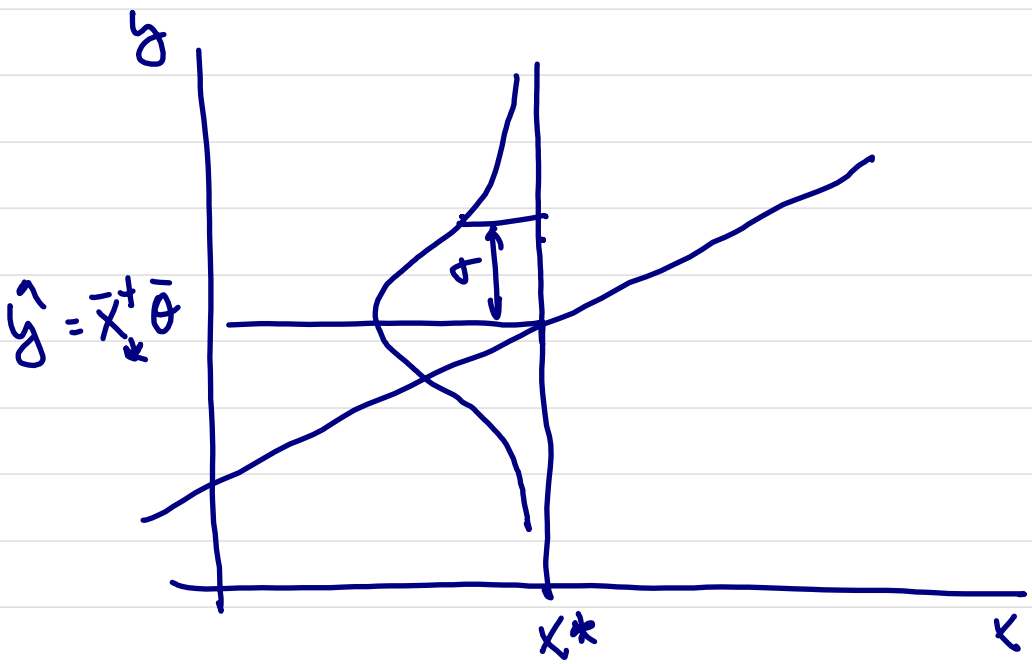
$$\rightarrow \boxed{\hat{\bar{\theta}} = (\bar{\bar{X}}^T\bar{\bar{X}})^{-1}\bar{\bar{X}}^T\bar{y}} \quad !!!$$

Now we can also estimate the variance σ

$$\frac{\partial \log(P)}{\partial \sigma} = 0 \rightarrow \sigma^2 = \frac{1}{n}(\bar{y}-\bar{\bar{X}}\bar{\theta})^T(\bar{y}-\bar{\bar{X}}\bar{\theta})$$

$$= \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{x}_i\bar{\theta})^2$$

Making predictions



Regularization

$$\hat{\theta} = (\bar{\bar{X}}^t \bar{\bar{X}})^{-1} \bar{\bar{X}}^t \bar{y}$$

$\bar{\bar{X}}^t \bar{\bar{X}}$ needs to be inverted and can be ill conditioned

Solution: Add a constant

$$\hat{\theta} = (\bar{\bar{X}}^t \bar{\bar{X}} + \delta^2 I)^{-1} \bar{\bar{X}}^t \bar{y}$$

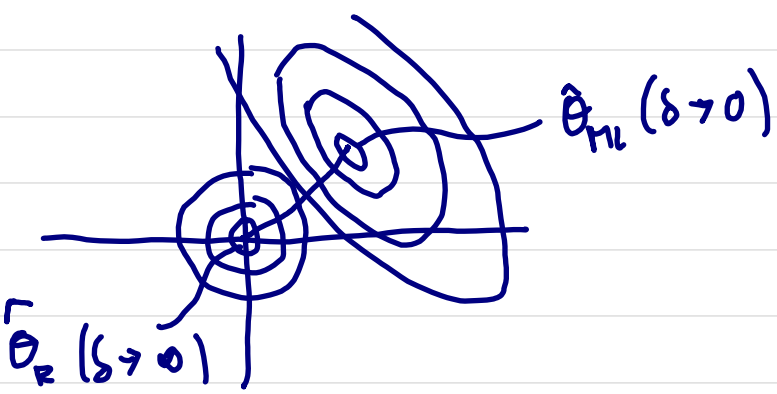
This is the "ridge regression estimate" solution to the regularized quadratic cost function: "penalized least squares"

$$J(\bar{\theta}) = (\bar{y} - \bar{\bar{X}}\bar{\theta})^t (\bar{y} - \bar{\theta}\bar{\bar{X}}) + \underbrace{\delta^2 \bar{\theta}^t \bar{\theta}}_{\text{penalty}}$$

Example: 2D $\bar{\theta} = (\theta_0, \theta_1)$

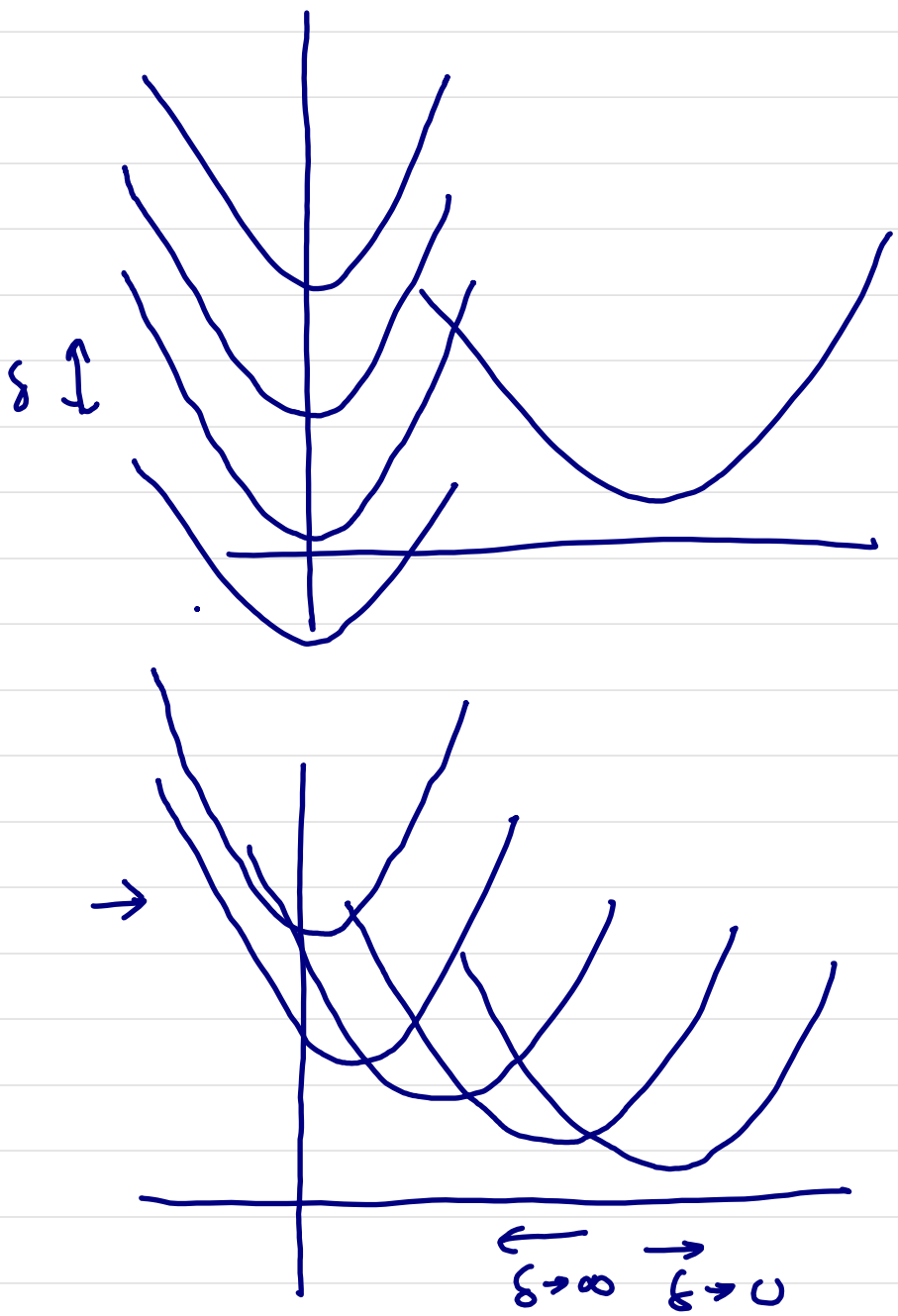
penalty $\downarrow$ $\theta_0^2 + \theta_1^2$

$J(\theta)$ = paraboloids



θ follows the points where the curves touch tangentially

A physical analogy would be finding the minimum of a potential

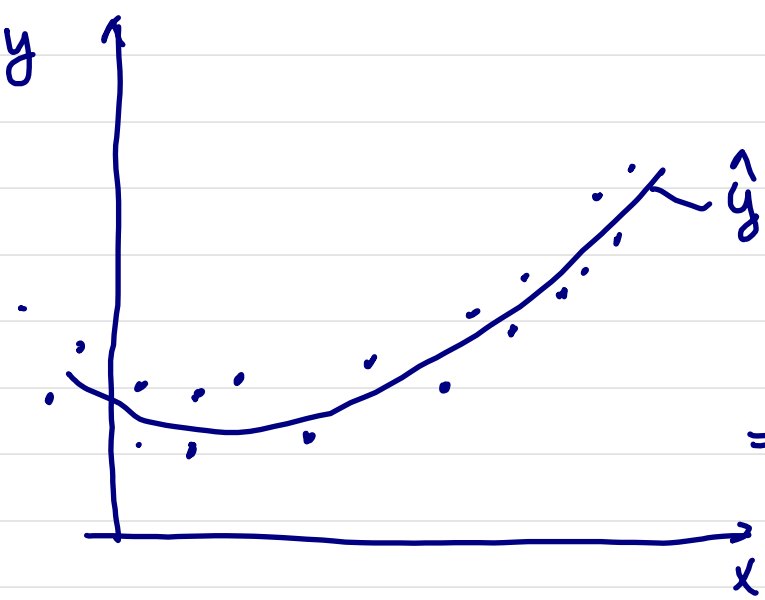


Non linear fit

We introduce a basis of functions $\phi(\cdot)$

$$y(\bar{x}) = \bar{\phi}(\bar{x})\bar{\theta}$$

Example: $\bar{\phi}(x) = [1, x, x^2]$



Example: multi-variate

$\bar{\phi}(\bar{x}) = [1, x_1, x_2]$

linear
(plane)

$\bar{\phi}(\bar{x}) = [1, x_1, x_2, x_1^2, x_2^2]$

quadratic
(paraboloid)

what about cross terms? $(x_1 x_2)$

Optimization

$$\hat{y}(\bar{x}_i) = \bar{\theta}_0 + \bar{x}_i \bar{\theta}_1 + \bar{x}_i^2 \bar{\theta}_2 + \bar{x}_i^3 \bar{\theta}_3 + \cdots$$

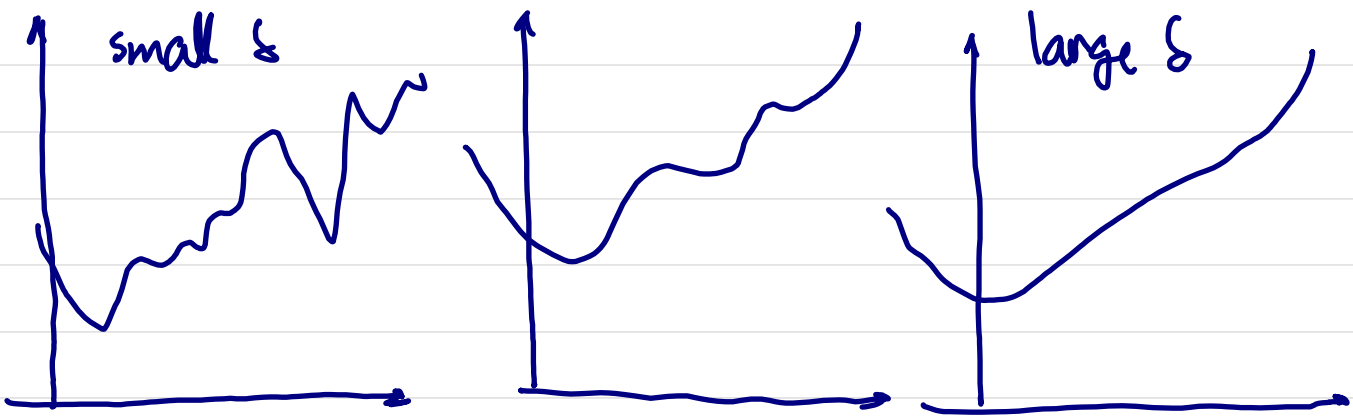
In 1D: $\hat{y}(x_i) = \theta_0 + x_i \theta_1 + x_i^2 \theta_2 + x_i^3 \theta_3 + \cdots$

We define the matrix

$$\bar{\bar{\Phi}} = \begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 & \cdots \\ 1 & x_2 & x_2^2 & x_2^3 & \cdots \\ & & & & \\ & & & & \\ 1 & x_n & x_n^2 & x_n^3 & \cdots \end{bmatrix}$$

$$\hat{y} = \bar{\bar{\Phi}} \bar{\theta}$$

$$J(\bar{\theta}) = (\bar{y} - \bar{\bar{\Phi}} \bar{\theta})^T (\bar{y} - \bar{\bar{\Phi}} \bar{\theta}) + \delta \bar{\theta}^T \bar{\theta}$$



for small δ, the polynomial will try to fit as many points as possible. In fact, if the number of points equals the degree of the polynomial minus 1 $(p-1)$ the fit will go through all the points...

Is this a "good" fit? Not necessarily.

This is an example of overfitting. The regularizer δ will "kill" some of the θ's making the curve "smoother".

How do we pick δ?

Kernel regression

Also referred-to as radial basis functions

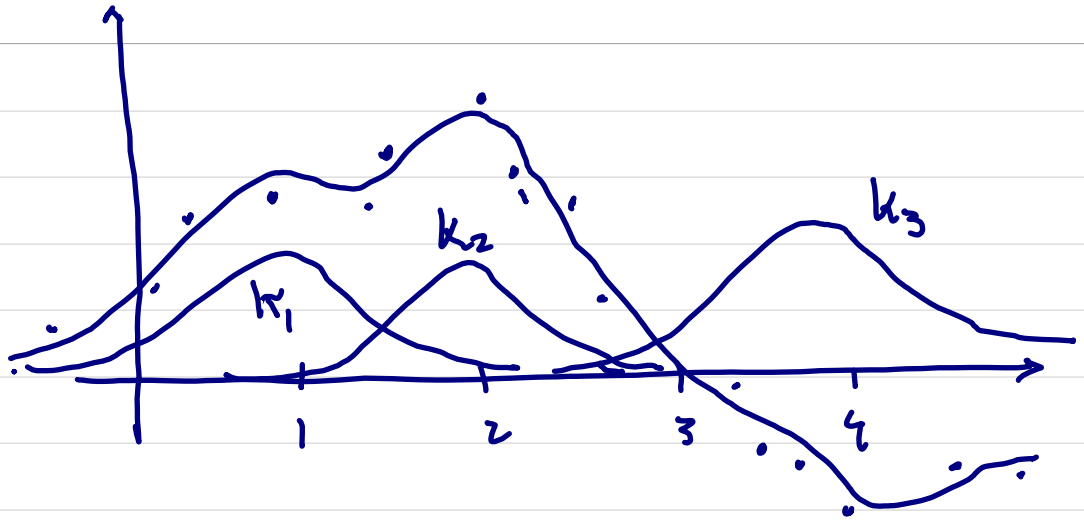
$$\vec{\phi}(\vec{x}) = \left[K(\vec{x}, \vec{\mu}_1, \lambda) \cdots K(\vec{x}, \vec{\mu}_d, \lambda) \right]$$

with $K_i = e^{\left(-\frac{1}{\lambda}\|\vec{x} - \vec{\mu}_i\|^2\right)}$

$$\hat{y}(\vec{x}_i) = \theta_0 + K(x_i, \mu_1, \lambda)\theta_1 + \cdots K(x_i, \mu_d, \lambda)\theta_d$$

Example: $\hat{y}(x) = e^{-(x-1)^2}\theta_1 + e^{-(x-2)^2}\theta_2 + e^{-(x-4)^2}\theta_3$

$(\lambda = 1)$



Some considerations about fitting

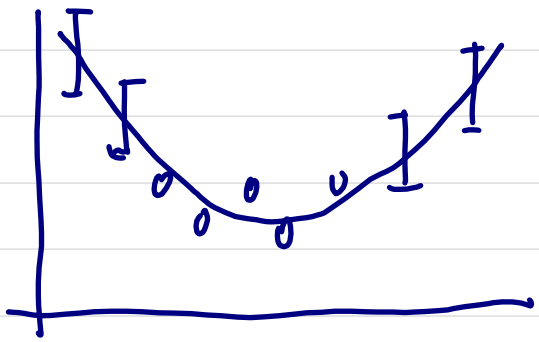
- When the size of our training dataset gets larger the error converges to the "optimal" error
- When the model is too simple, it will not be able to capture reality → bias, systematic error

Problem: many times we don't know the "right" model.
We can try by adding more complexity, for instance, more free parameters, higher degree polynomial, basis functions.
This can lead to overfitting and not necessarily to a better model. this can improve with regularization.

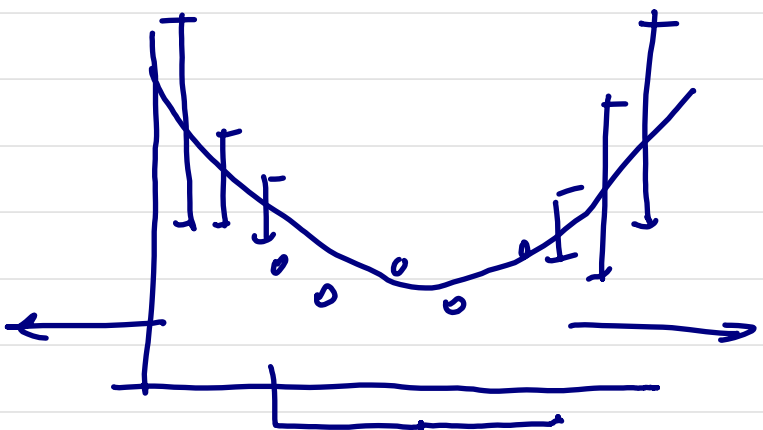
- More data improve results, only if model has the right complexity.

Confidence in the prediction

What MLE gives us is a constant variance or uncertainty



But in reality, uncertainty should increase away from the known data



→ Bayesian learning !

Probability: some definitions

- Probability of the union of two events

$$P(A \vee B) = P(A) + P(B) - P(A \wedge B)$$

($\vee$: or; $\wedge$: and)

$$= P(A) + P(B) \quad \text{if mutually exclusive}$$

- Joint probability

$$P(A \wedge B) = P(B \wedge A) = P(AB) =$$

$$= P(A|B)P(B) = P(B|A)P(A)$$

- Marginal probability

$$P(A) = \sum_b P(AB)_b = \sum_b P(A|B=b)P(b)$$

- Conditional probability

$$P(A|B) = \frac{P(AB)}{P(B)} \quad \text{if } P(B) > 0$$

($|$: given)

Bayes rule

posterior

$$p(X=x \mid Y=y) = \frac{p(X=x, Y=y)}{p(Y=y)} = \frac{p(X=x)\, p(Y=y \mid X=x)}{\sum_{x'} p(X=x')\, p(Y=y \mid X=x')}$$

given

likelihood

prior (marginal)

Example: Medical diagnosis

Mamogram sensitivity is 80% → you test positive with 80% if you have cancer

$$\rightarrow p(T=1 \mid D=1) = 0.8$$

The probability of having breast cancer is 4 in 1000

$$\rightarrow p(D=1) = 0.004$$

False positives are also quite common, 10%

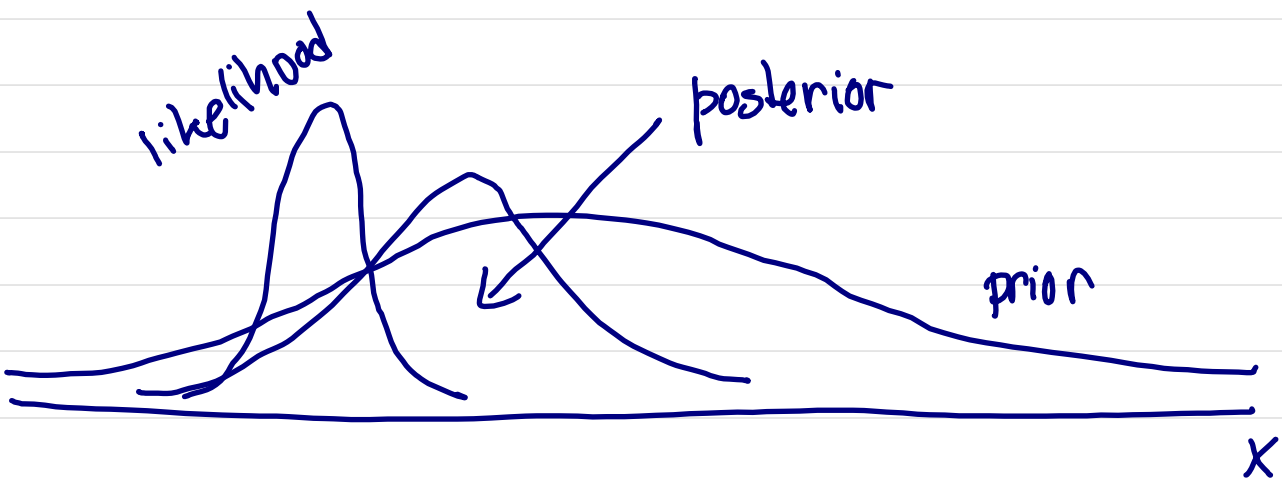
$$p(T=1 \mid D=0) = 0.1$$

We can now obtain the probability of having cancer if the test is positive

$$P(D=1|T=1) = \frac{P(T=1|D=1)P(D=1)}{P(T=1|D=1)P(D=1) + P(T=1,D=0)P(D=0)}$$

$$= 0.031 \quad \text{or} \quad 3\%$$

This explains why the great majority of cases are self diagnosed by inspection.



Bayesian linear regression

The likelihood is a Gaussian $\mathcal{N}(\bar{y}, \bar{\bar{X}}\bar{\theta}, \sigma^2)$

The prior is also a Gaussian

$$p(\bar{\theta}) = \mathcal{N}(\bar{\theta}, \bar{\theta}_0 \bar{\bar{V}}_0)$$

($\bar{\theta}$: mean, $\bar{\bar{V}}_0$: variance(s))

Using Baye's rules and properties of Gaussians, we obtain the posterior

$$p(\bar{\theta} \mid \bar{\bar{X}}, \bar{y}, \sigma^2) \propto \mathcal{N}(\bar{\theta} \mid \bar{\theta}_0, \bar{\bar{V}}_0)\, \mathcal{N}(\bar{y} \mid \bar{\bar{X}}\bar{\theta}, \sigma^2)$$

$$= \mathcal{N}(\bar{\theta} \mid \bar{\theta}_n, \bar{\bar{V}}_n)$$

with $\bar{\theta}_n = \bar{\bar{V}}_n \bar{\bar{V}}_0^{-1} \bar{\theta}_0 - \frac{1}{\sigma^2} \bar{\bar{V}}_n \bar{\bar{X}}^T \bar{y}$

$\bar{\bar{V}}_n^{-1} = \bar{\bar{V}}_0^{-1} - \frac{1}{\sigma^2} \bar{\bar{X}}^T \bar{\bar{X}}$

"n" is the # of datapoints

$$p(\bar{\theta} \mid \bar{\bar{X}}, \bar{y}, \sigma^2) \propto e^{-\frac{1}{2}(\bar{\theta} - \bar{\theta}_n) \bar{\bar{V}}_n^{-1} (\bar{\theta} - \bar{\theta}_n)}$$

Normalization: $|2\pi \bar{\bar{V}}_n|^{-1/2}$

Consider the special case $\bar{\theta}_0 = \bar{0}$ and $\bar{\bar{V}}_0 = \tau_0 \mathbb{I}$

$$\bar{\theta}_n = \left(\lambda \mathbb{I} + \bar{\bar{X}}^T \bar{\bar{X}}\right)^{-1} \bar{\bar{X}}^T \bar{y} \quad \text{with } \lambda = \frac{\sigma^2}{\tau_0^2}$$

and we recover the ridge regression!

Note: If we don't have any knowledge about the prior, meaning that $p(\theta)$ is very flat or constant, then $\tau_0 \to \infty$ and $\lambda \to 0$ → linear regression

Prediction

Posterior mean: $\bar{\theta}_n = \left(\lambda \mathbb{I} + \bar{\bar{X}}^T \bar{\bar{X}}\right)^{-1} \bar{\bar{X}}^T \bar{y}$

Posterior variance: $\bar{\bar{V}}_n = \sigma^2 \left(\lambda \mathbb{I} + \bar{\bar{X}}^T \bar{\bar{X}}\right)^{-1}$

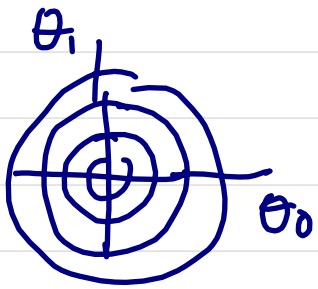
The prediction, given the training data $D(\bar{\bar{X}}, \bar{y})$ is a distribution

$$p(\bar{y} | x_* D \sigma^2) = \mathcal{N}\left(y | \bar{x}_*^T \bar{\theta}, \underbrace{\sigma^2 + \bar{x}_*^T \bar{\bar{V}}_n \bar{x}_*}\right)$$

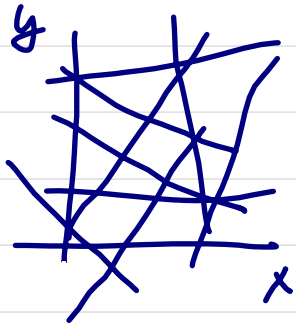
Example: $y(x, \bar{\theta}) = \theta_0 + \theta_1 x$

$$p(y|\bar{x}) = \mathcal{N}(y|\theta_0 + \theta_1 x, \sigma^2)$$

1 step: No points

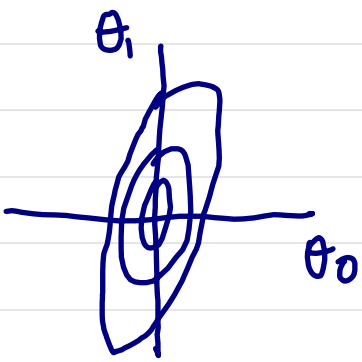


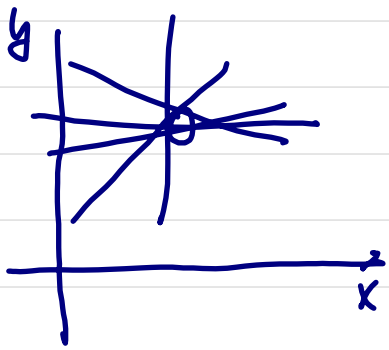
prior



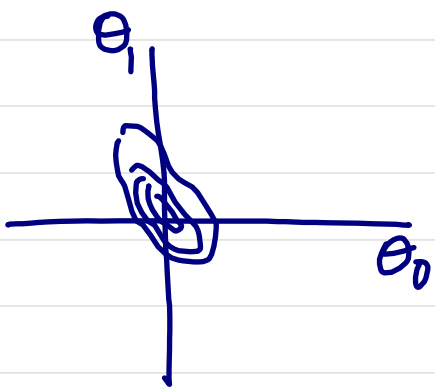
possible curves

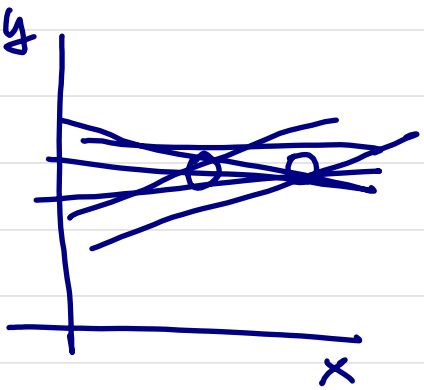
1 point





2 points





Many points