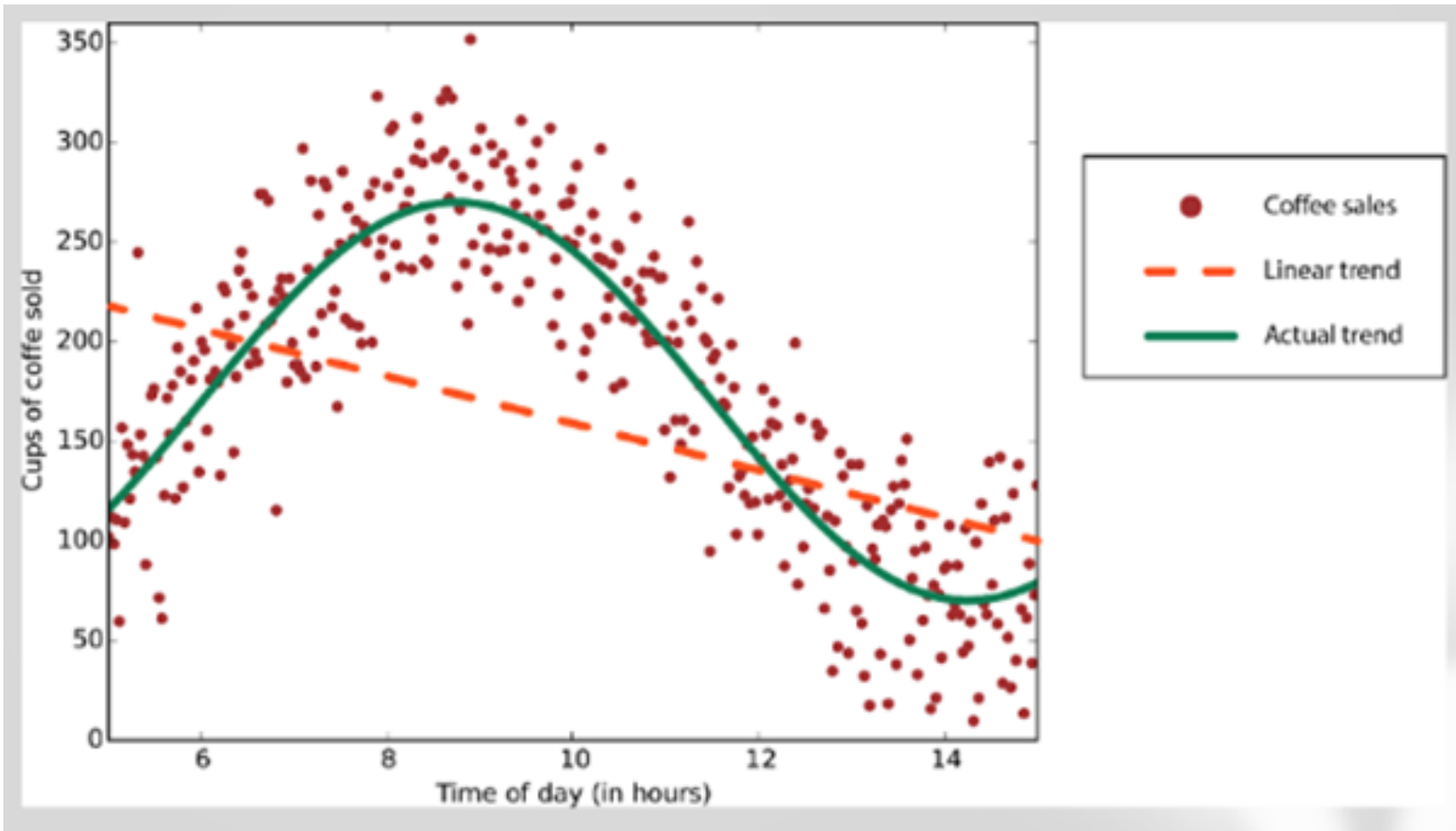
LINEAR REGRESSION

Dr. Srijith P K

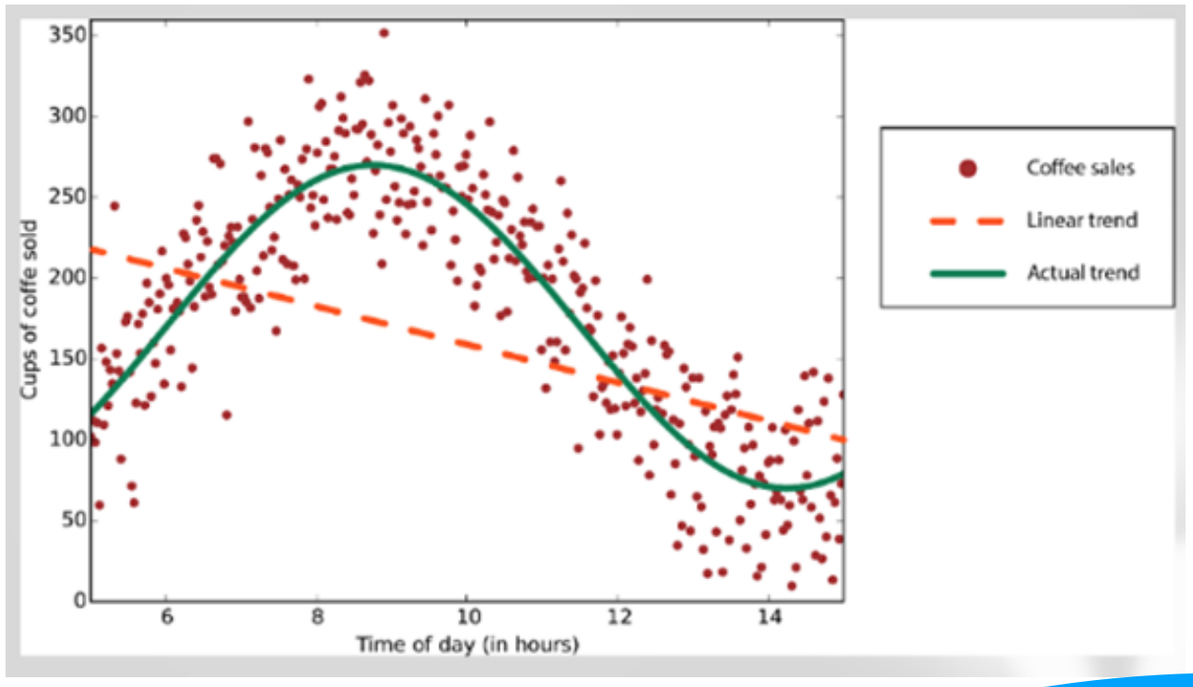
आई आई टी हैदराबाद
IIT Hyderabad

Supervised learning : Regression



https://medium.com/@heyozramos/regression-vs-classification-86d73c281c5e

Supervised learning : Regression



Real valued targets (outputs)

Generalization performance

Goal is to learn a function which maps inputs to outputs so that it will predict well on future data points

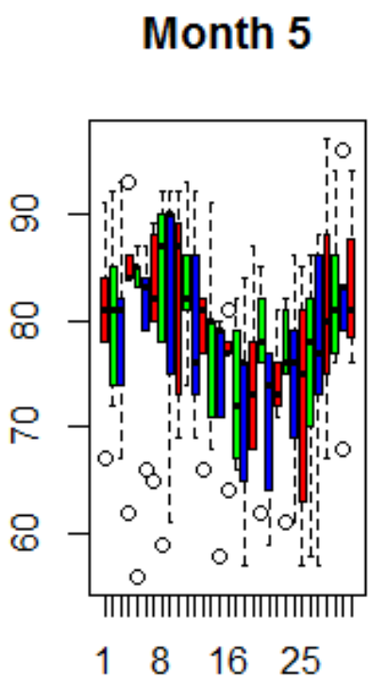
https://medium.com/@heyozramos/regression-vs-classification-86d73c281c5e

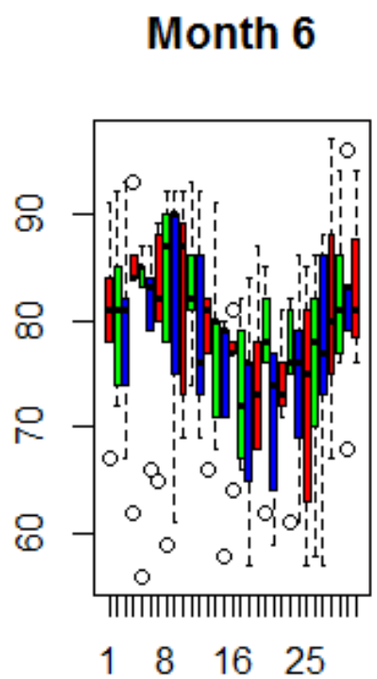
Airquality data.

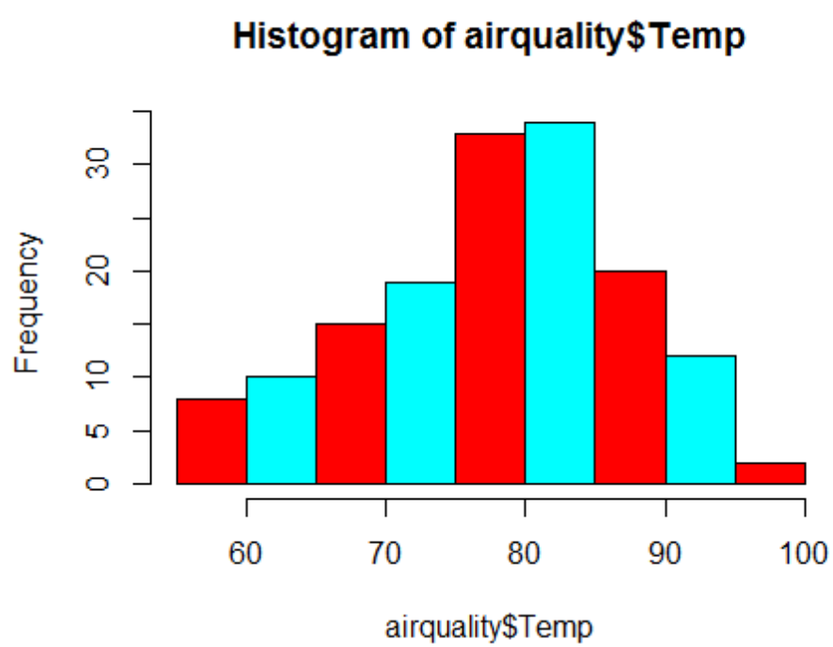
- Data set has various air quality parameters in New York city.
- These are the parameters in the data set:
- Daily temperature from May to August
- Solar radiation data
- Ozone data
- Wind data
- Goal : predict the temperature for a particular month in New York using solar radiation, ozone and wind data.

Airquality data

```
##    Ozone Solar.R Wind Temp Month Day
## 1     41     190  7.4   67     5   1
## 2     36     118  8.0   72     5   2
## 3     12     149 12.6   74     5   3
## 4     18     313 11.5   62     5   4
## 5     NA      NA 14.3   56     5   5
## 6     28      NA 14.9   66     5   6
## 7     23     299  8.6   65     5   7
## 8     19      99 13.8   59     5   8
## 9      8      19 20.1   61     5   9
## 10    NA     194  8.6   69     5  10
```

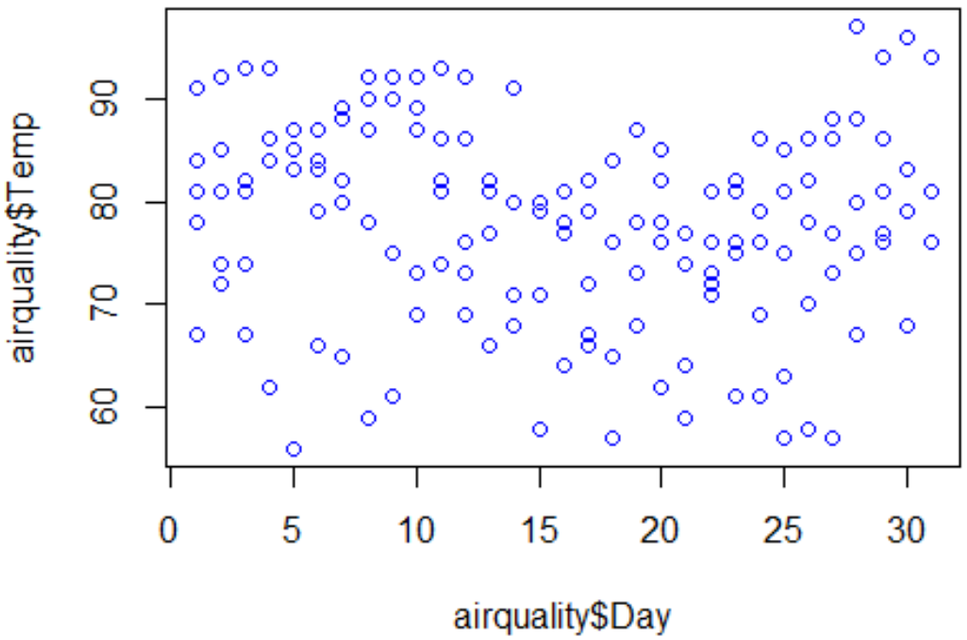


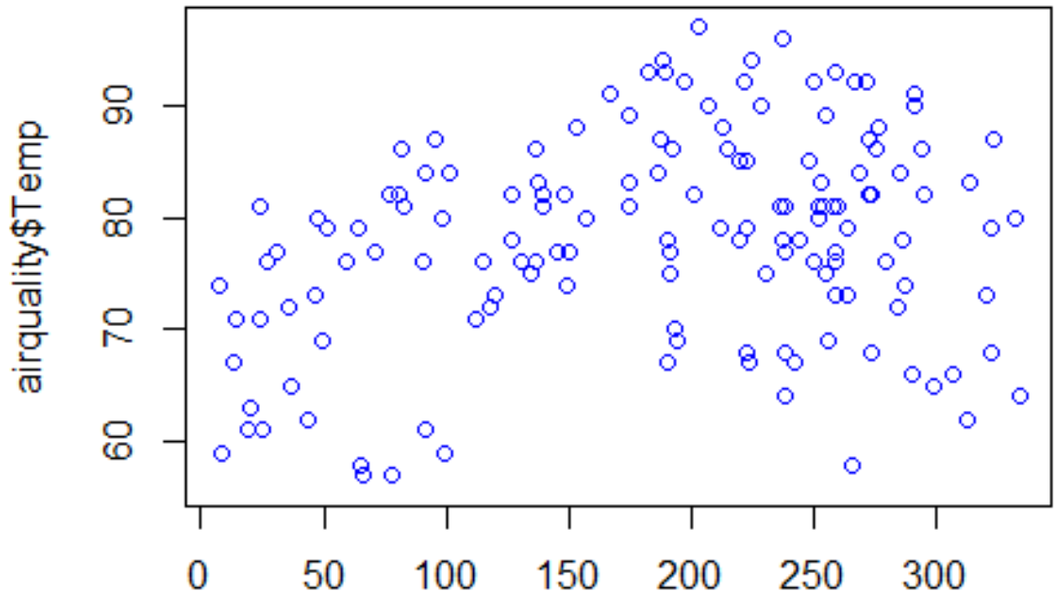


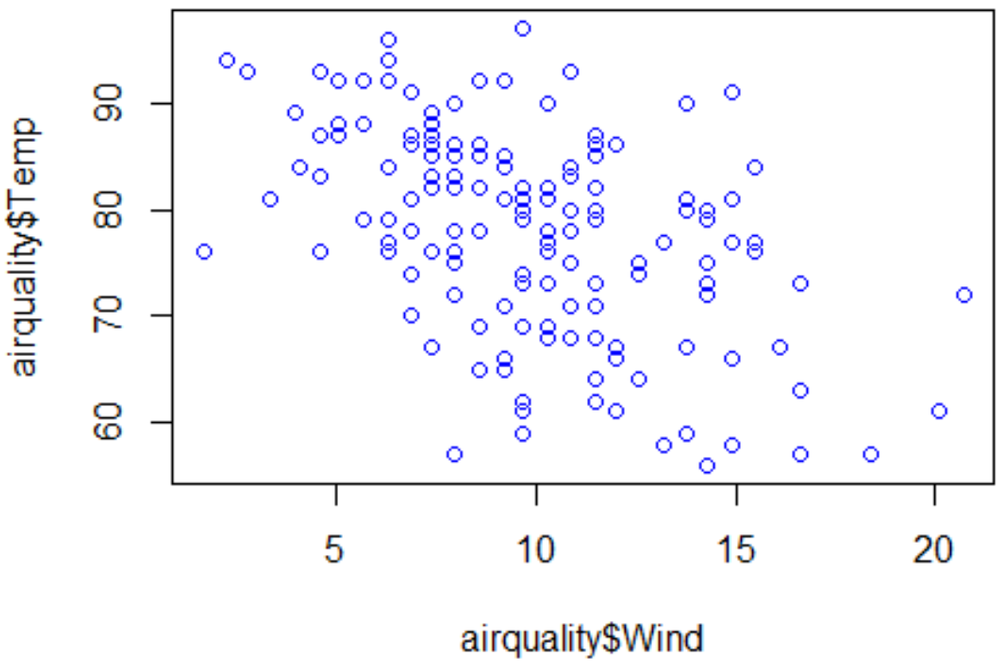


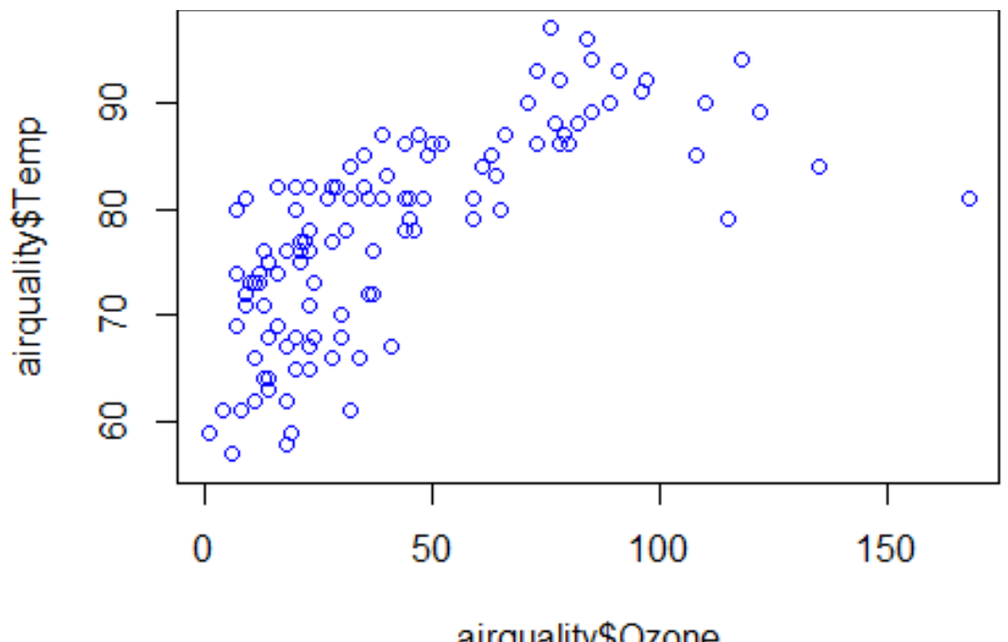
https://www.edvancer.in/step-step-guide-to-execute-linear-regression-r/

Airquality data









Linear regression

- Temp=w1.Solar.R +w2.Ozone + w3.Wind + error.

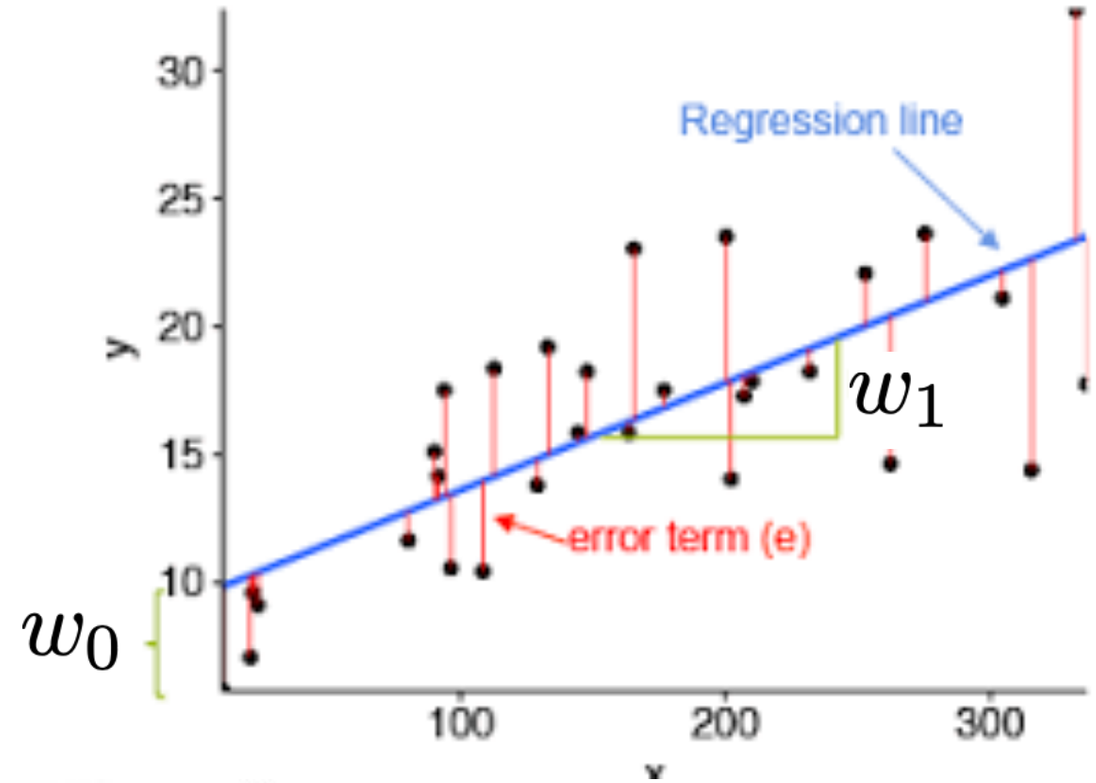
- Temperature of house depends on ozone, wind and solar radiations

- linear regression helps to discover relation between dependent and independent variables

Linear regression

Regression Output is real and scalar, $y \in \mathbb{R}$

- Learn a function which maps input to output f : X -> Y
- Learn the function which passes through as many points as possible



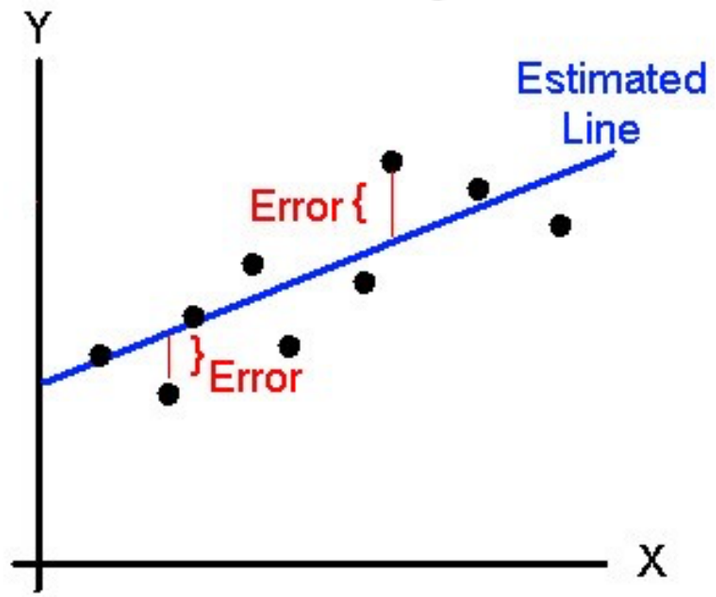
Estimated (or predicted) Y value for observation i

Estimate of the regression intercept

Estimate of the regression slope

Value of X for observation i

$$\hat{Y}_i = w_0 + w_1 X_i$$



Linear Regression

- Minimize the least squares error

$$E(w) = \frac{1}{2}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

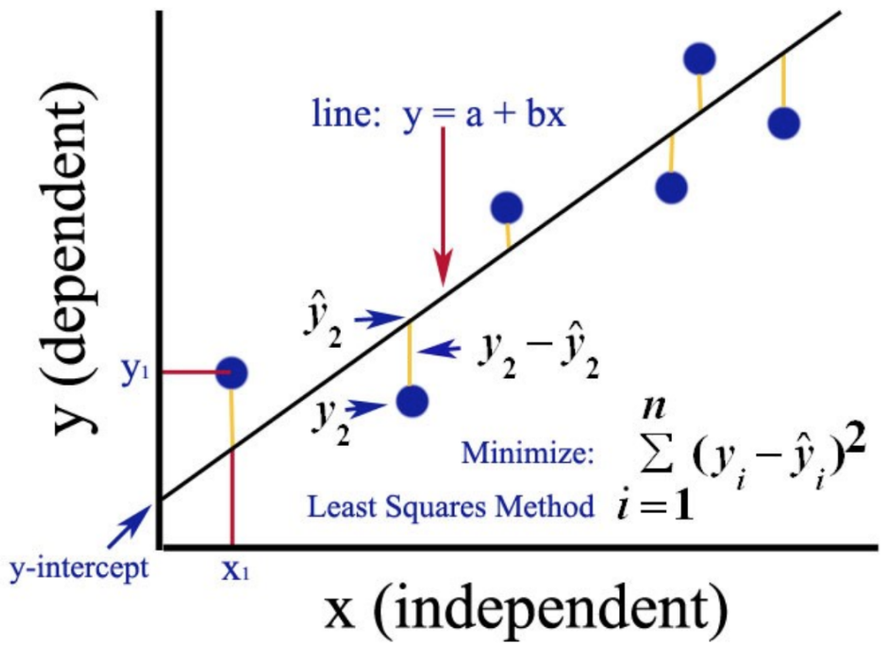
$$= \frac{1}{2}\sum_{i=1}^{N}(y_i - X_i^\top w)^2$$

$$\frac{1}{2}\|(y - X^\top w)\|^2$$

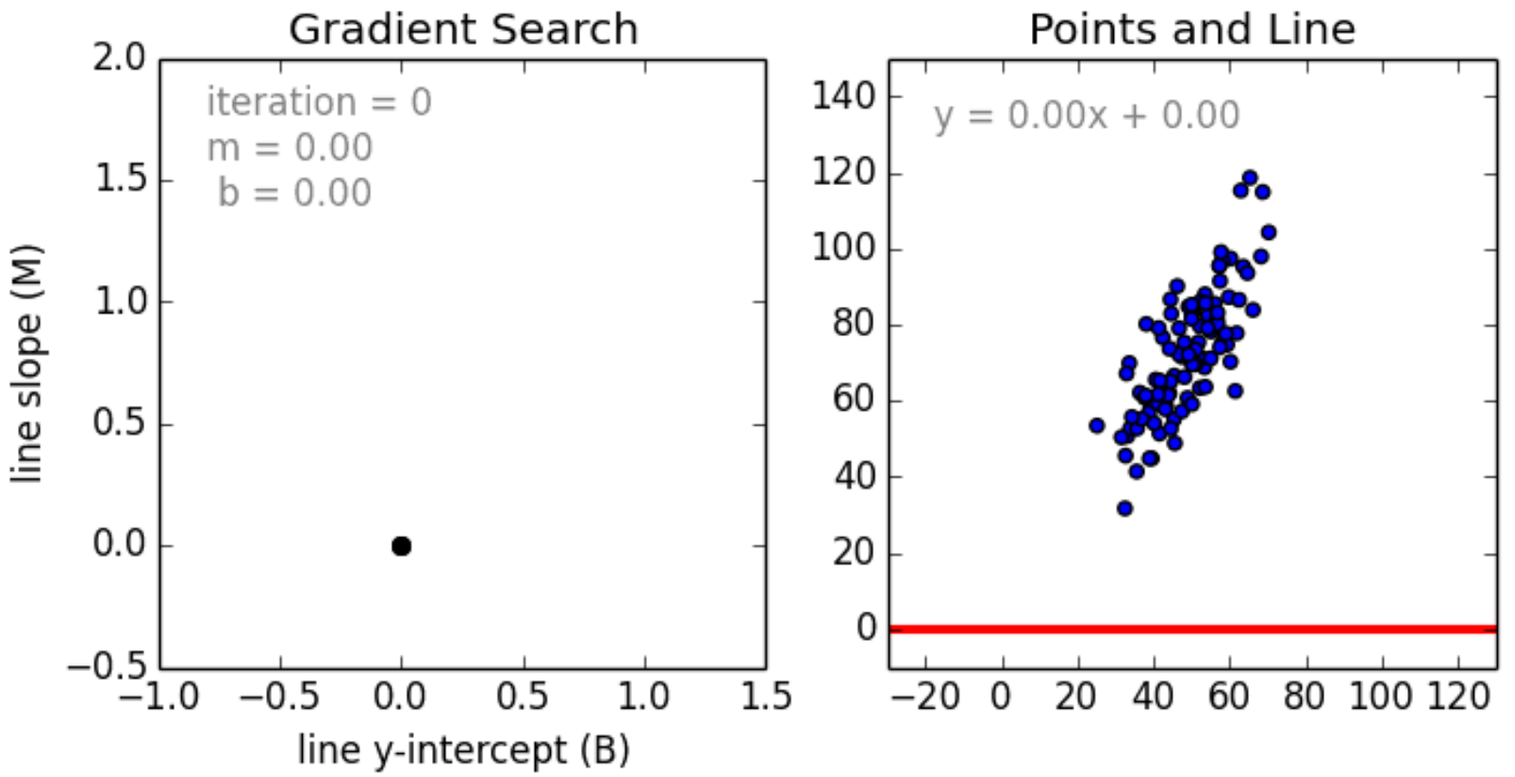
$$\nabla E(w) = Xy - XX^\top w = 0$$

$$w_{ML} = (XX^\top)^{-1}Xy$$

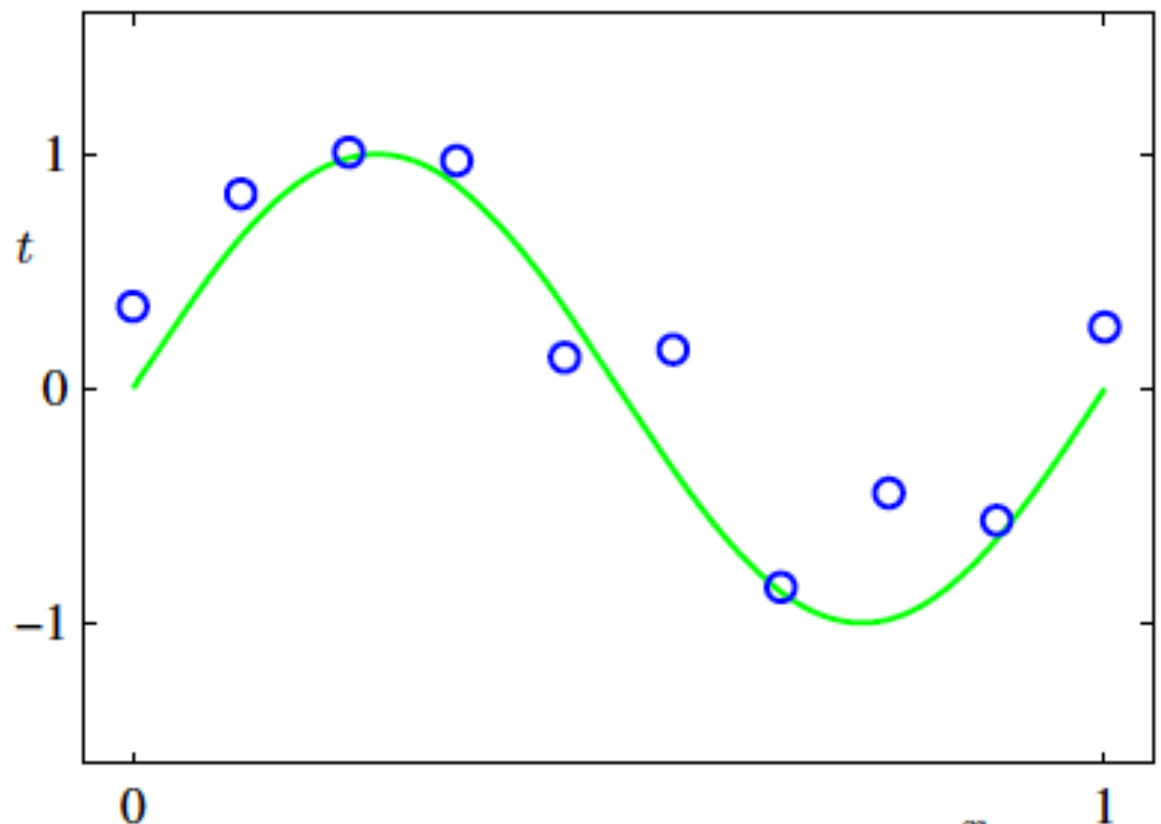
$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta\nabla E_n$$



$$Error_{(m,c)} = \frac{1}{N}\sum_{i=1}^{N}(y_i - (mx_i + c))^2$$



Regression - curve fitting

- Remember high school maths !
- Real-valued target variable t.
- Training set comprising N observations, features (input) x ≡ (x1, . . . , xN)T, and target (output) t ≡ (t1, . . . , tN)T.
- 

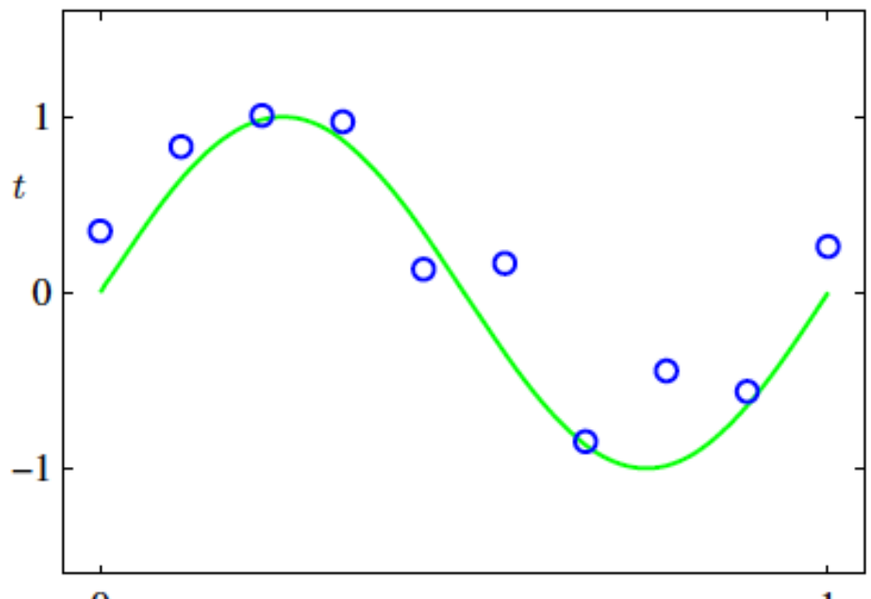
Regression - curve fitting

- M is the order of the polynomial, y(x,w) is a nonlinear function of x, it is a linear function of the coefficients w.

- Functions, such as the polynomial, which are linear in the unknown parameters have important properties and are called **linear models**

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

-

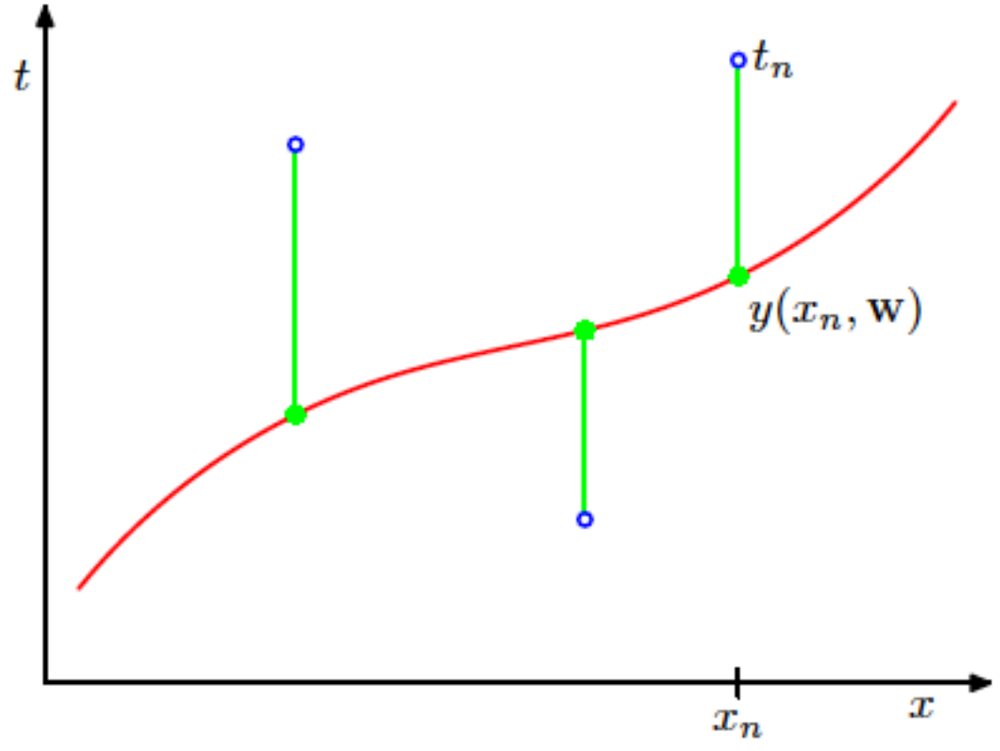


Regression - curve fitting

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

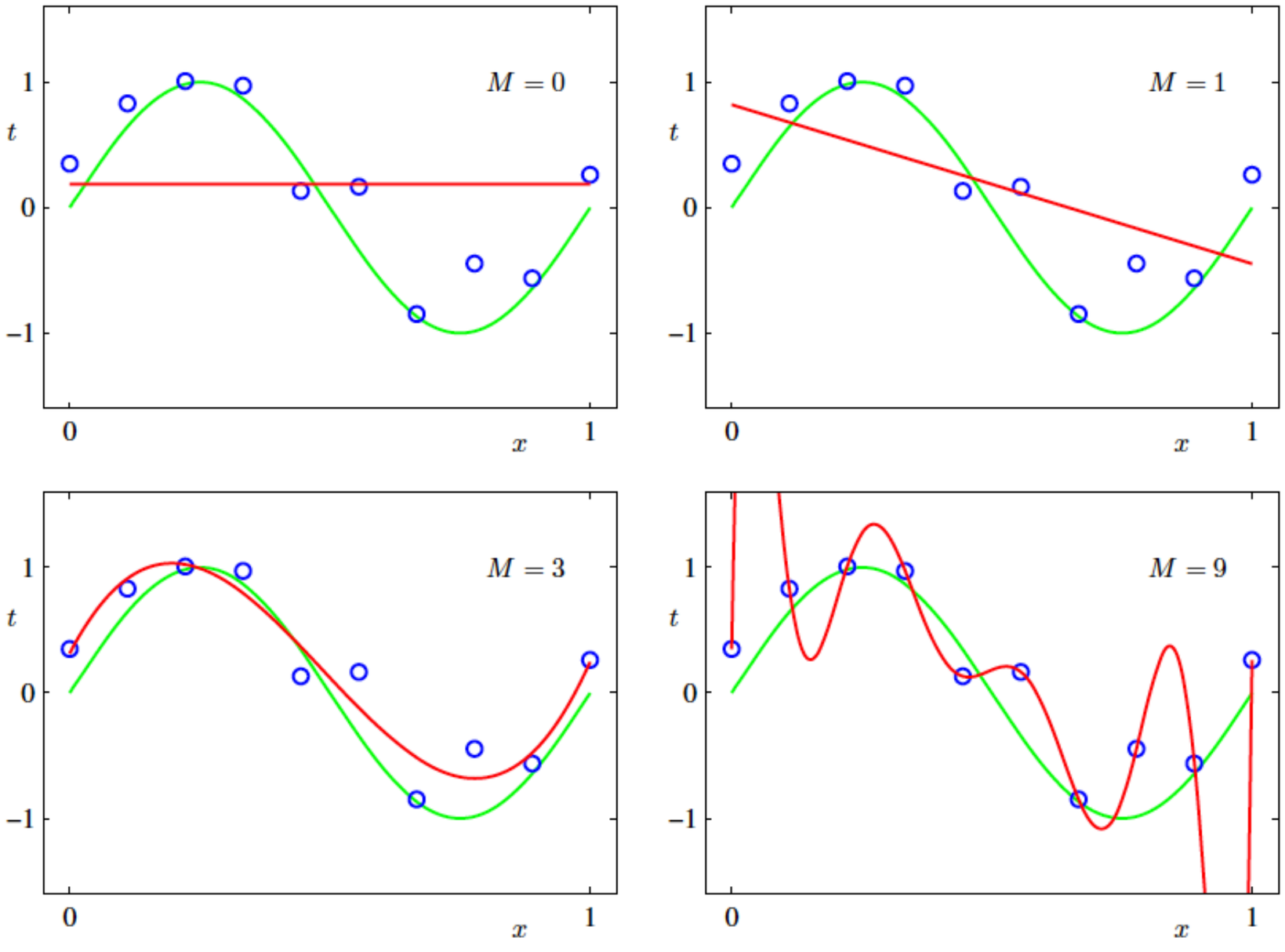
- Coefficients will be determined by fitting the polynomial to the training data. This can be done by minimizing an error function

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2$$

- 

Regression - curve fitting

- Model selection (choosing M) : higher order polynomial (M = 9), provide excellent fit to the training data but gives a very poor representation of the function

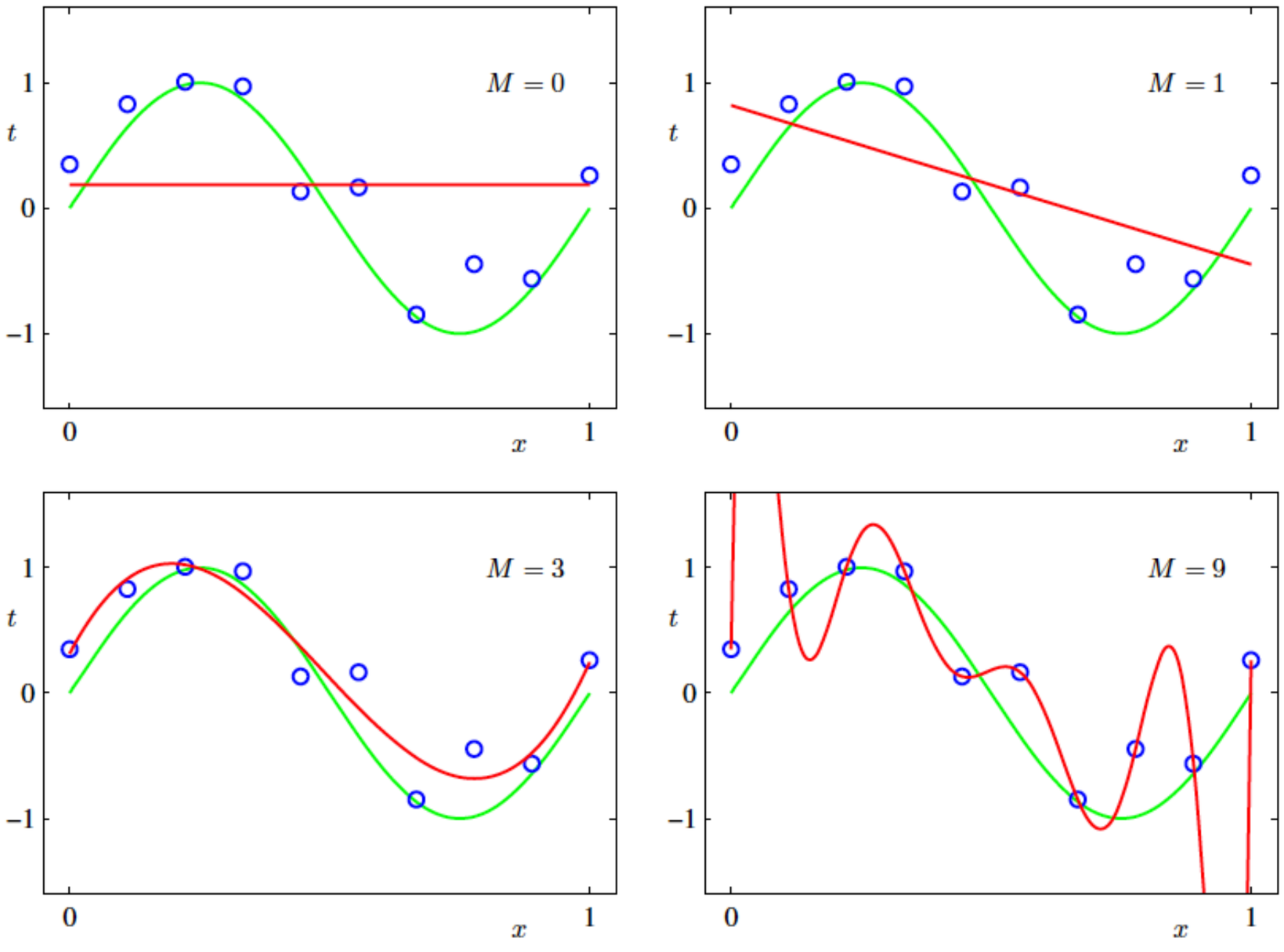


-

Regression - curve fitting

- Model selection (choosing M) : high… …lynomial (M = 9), provide excellent fit to the trainin… …a very poor representation of the function

Overfitting

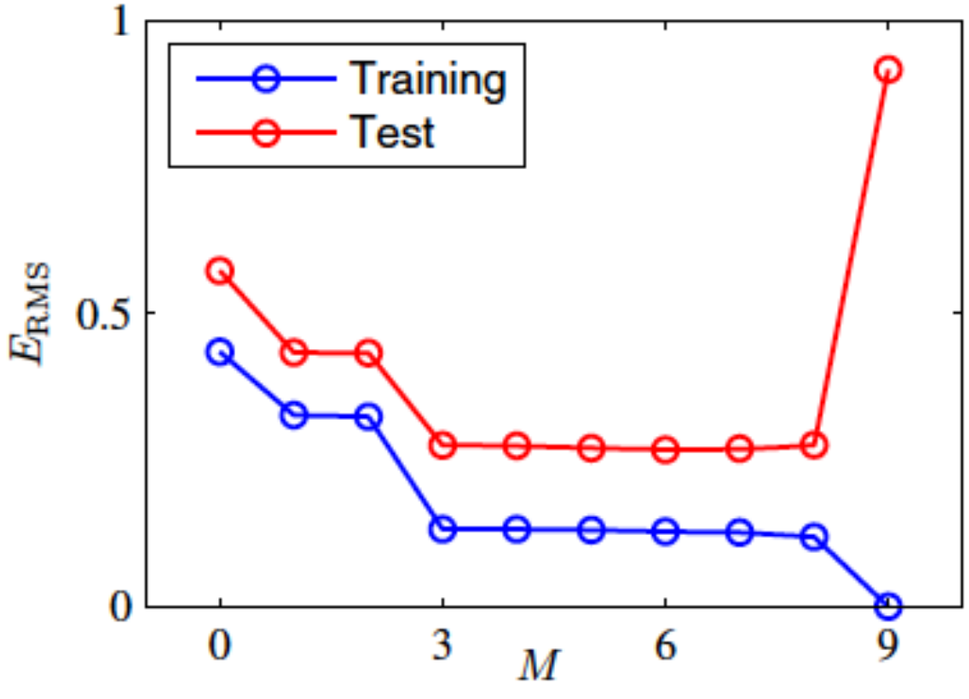


Regression - curve fitting

- Generalization performance measured using the root mean square error on test data

$$E_{\mathrm{RMS}} = \sqrt{2E(\mathbf{w}^\star)/N}$$

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2$$

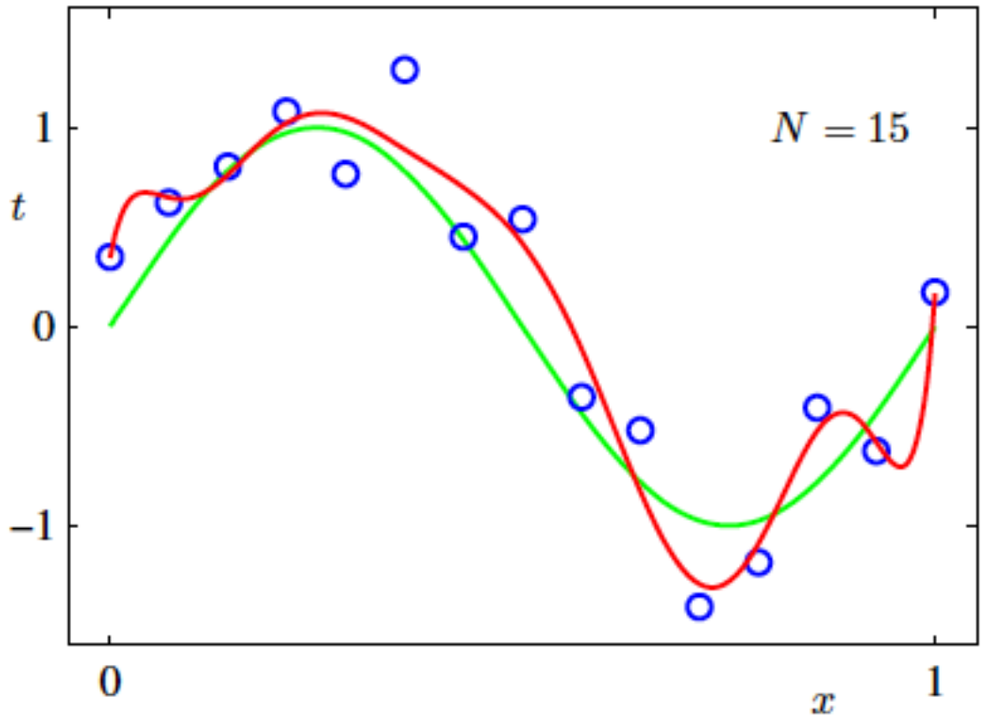


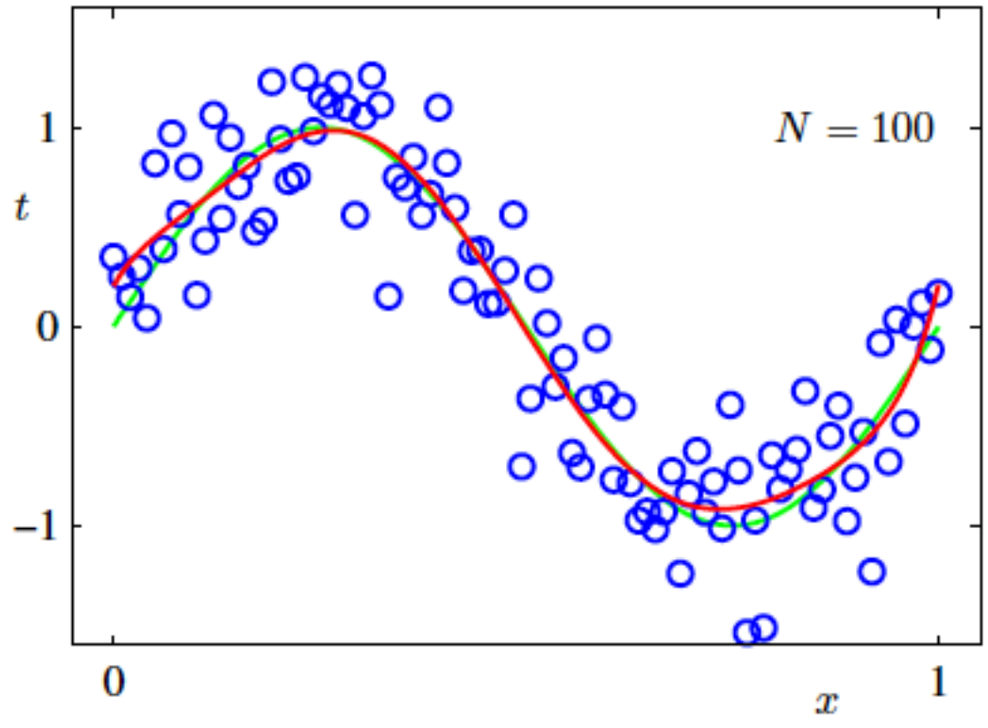
	$M=0$	$M=1$	$M=6$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

- More flexible polynomials with larger values of M are becoming increasingly tuned to the random noise on the target values.

Regression - curve fitting

- Given model complexity, the over-fitting problem become less severe as the size of the data set increases.





Curve fitting - regularization

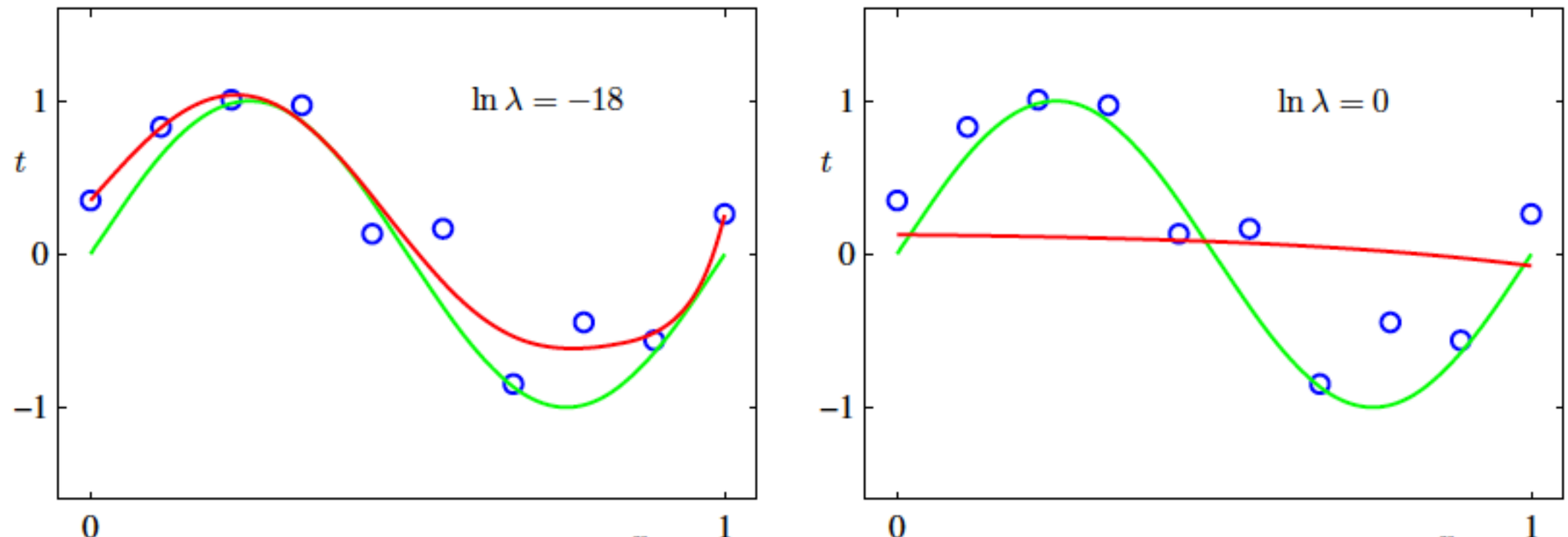
- Add a penalty term to the error function (1.2) in order to discourage the coefficients from reaching large values

$$\widetilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n,\mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

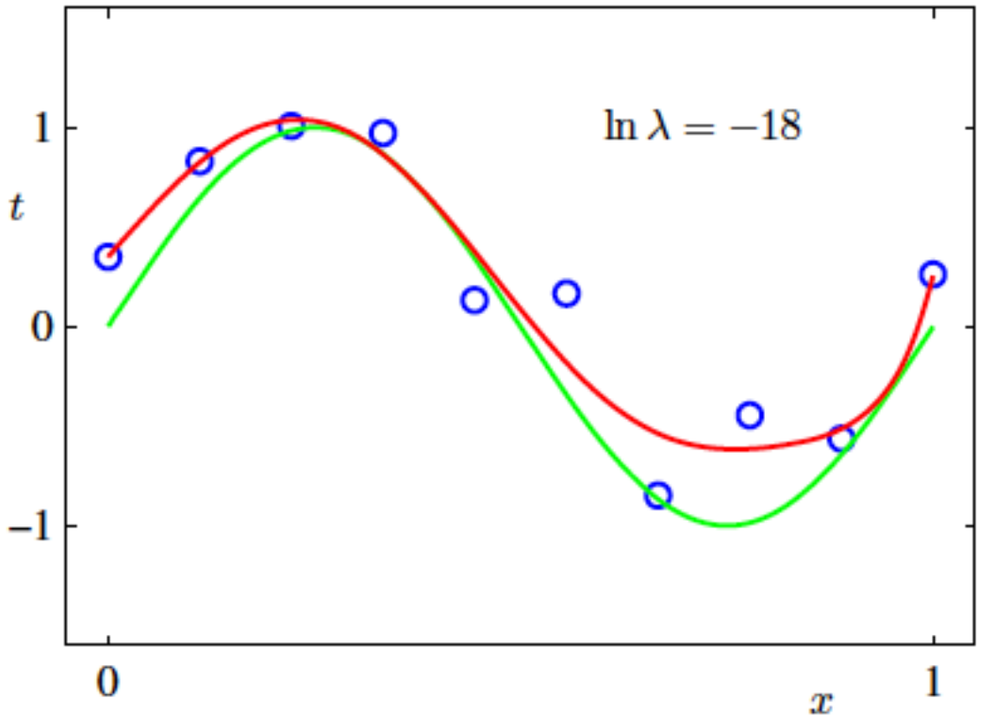
Regularization constant

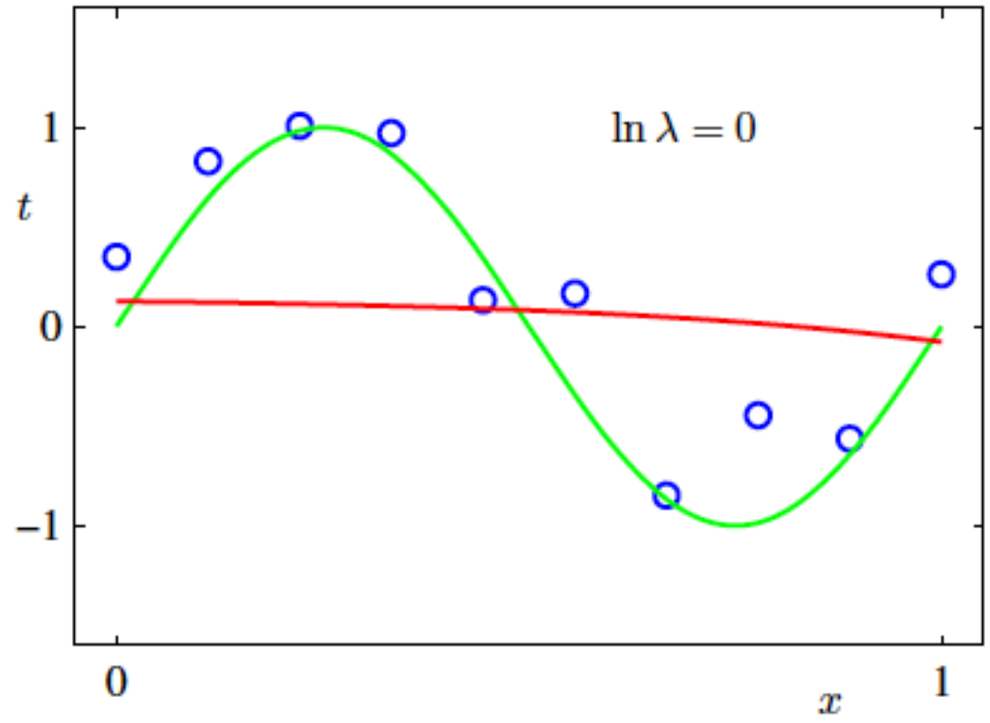
- Ridge regression : L2 norm

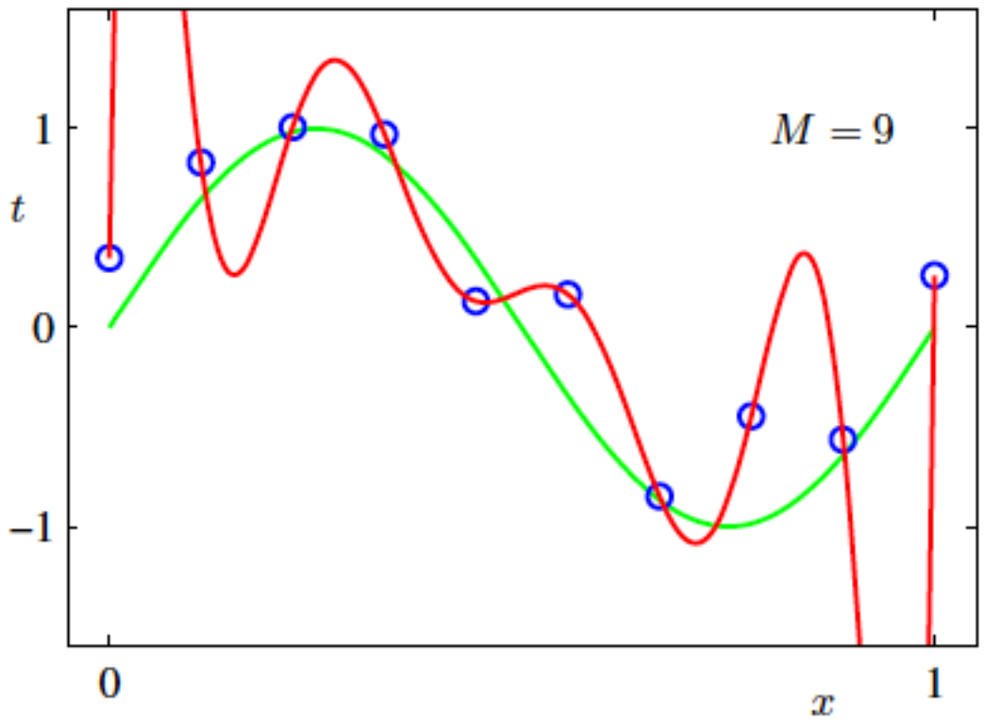
$$\|\mathbf{w}\|^2 \equiv \mathbf{w}^{\mathrm{T}}\mathbf{w} = w_0^2 + w_1^2 + \ldots + w_M^2$$

- 

Curve fitting - regularization







	$\ln \lambda = -\infty$	$\ln \lambda = -18$	$\ln \lambda = 0$
$w_0^\star$	0.35	0.35	0.13
$w_1^\star$	232.37	4.74	-0.05
$w_2^\star$	-5321.83	-0.77	-0.06
$w_3^\star$	48568.31	-31.97	-0.05
$w_4^\star$	-231639.30	-3.89	-0.03
$w_5^\star$	640042.26	55.28	-0.02
$w_6^\star$	-1061800.52	41.32	-0.01
$w_7^\star$	1042400.18	-45.95	-0.00
$w_8^\star$	-557682.99	-91.53	0.00
$w_9^\star$	125201.43	72.68	0.01

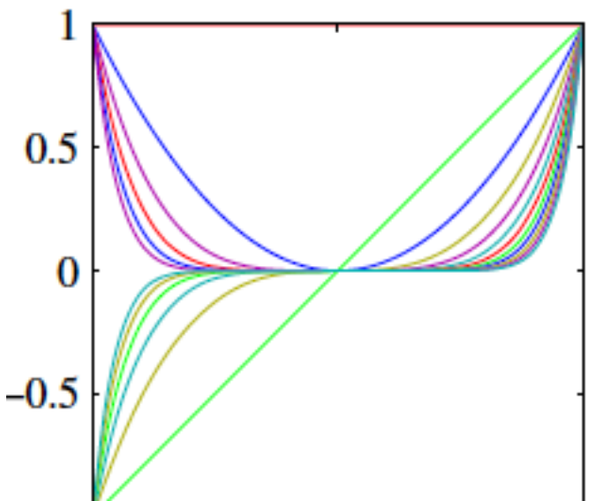
Generalized linear models

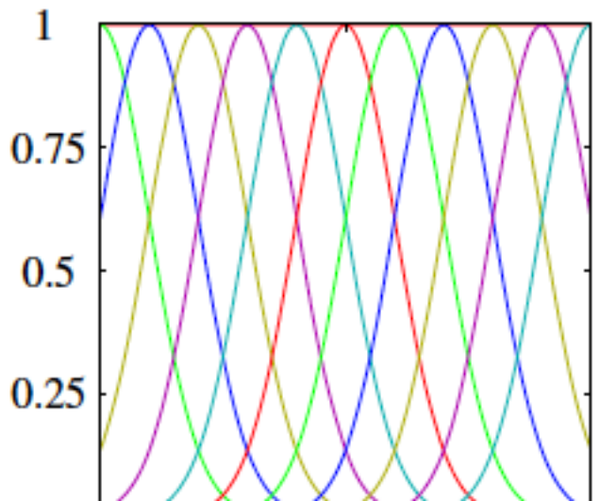
$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x}) \quad = \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x})$$

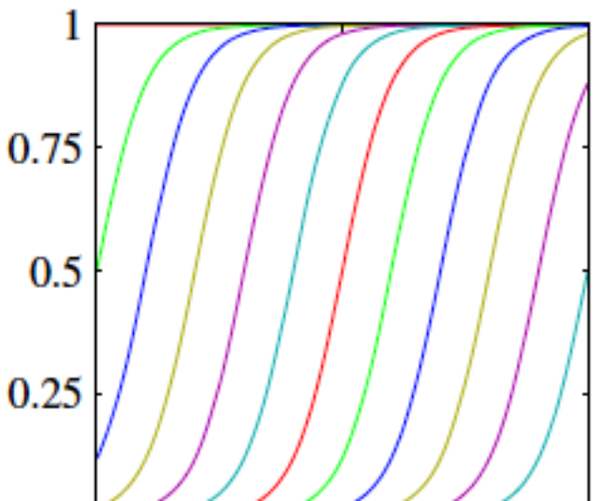
$$\mathbf{w} = (w_0, \ldots, w_{M-1})^{\mathrm{T}}$$

$$\boldsymbol{\phi} = (\phi_0, \ldots, \phi_{M-1})^{\mathrm{T}}$$

- ϕj(x) are known as basis functions, ϕj(x) = x^j .
- Gaussian basis function $\phi_j(x) = \exp\left\{-\frac{(x-\mu_j)^2}{2s^2}\right\}$
- Sigmoidal basis function $\phi_j(x) = \sigma\left(\frac{x-\mu_j}{s}\right) \qquad \sigma(a) = \frac{1}{1+\exp(-a)}.$





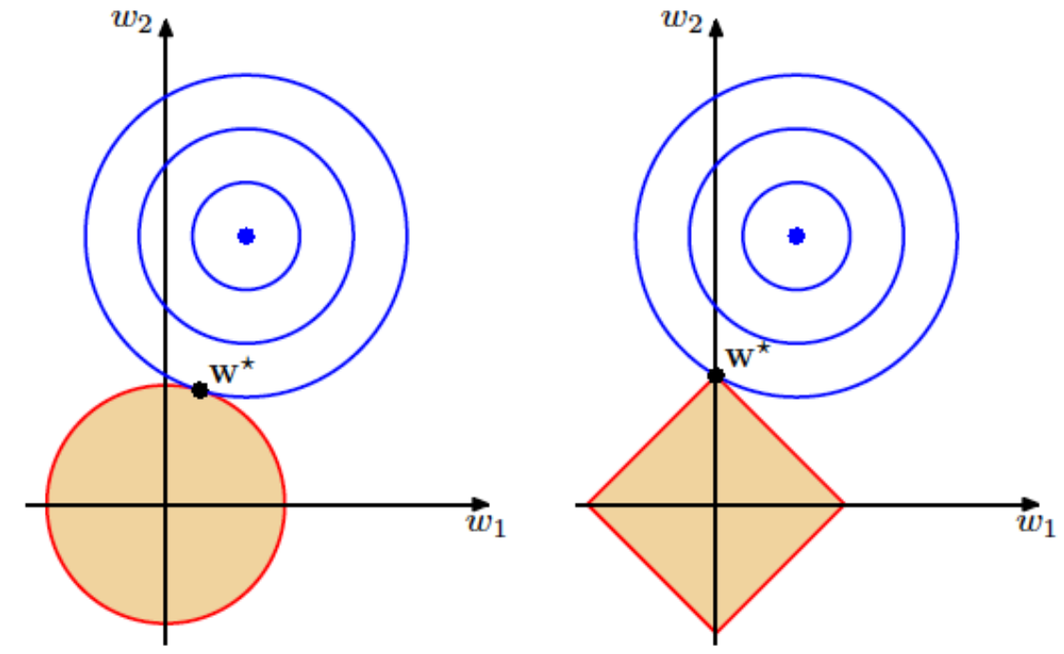


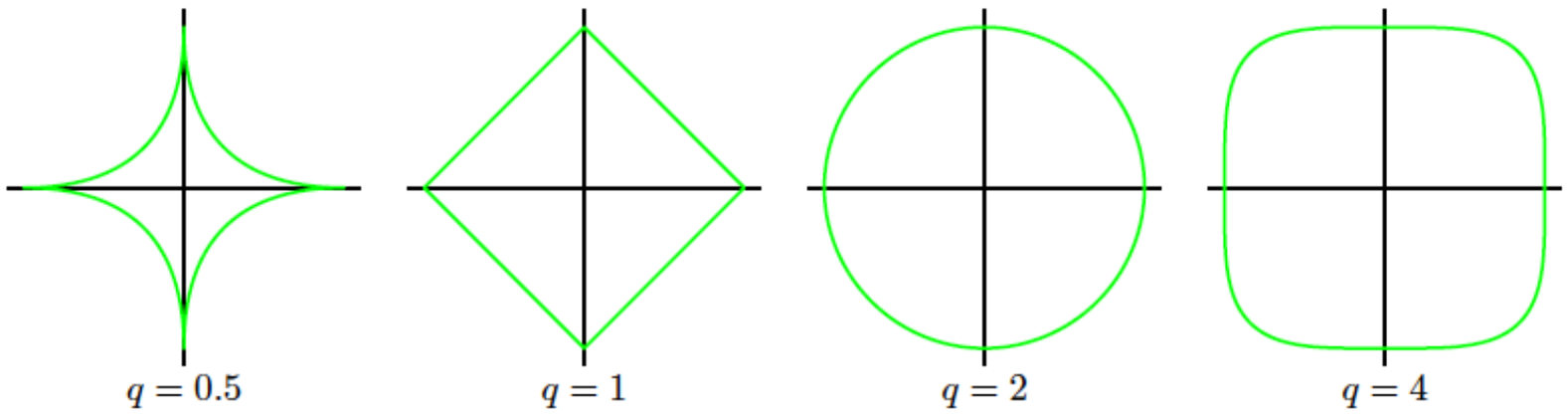
Regularized Least Squares

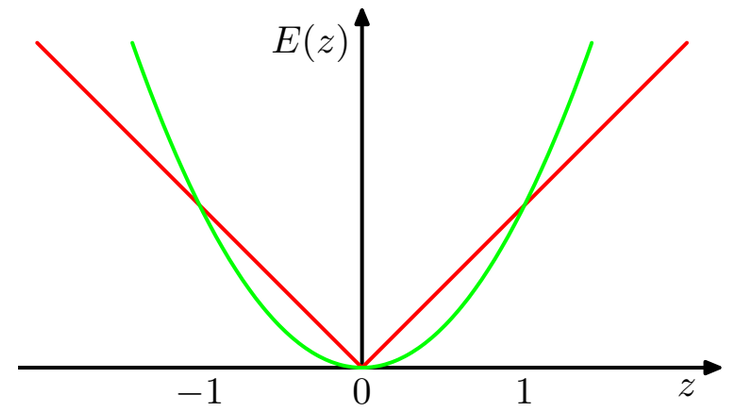
- Parameter shrinkage, weight decay
- **Ridge regression** q=2
- **Lasso regression** q=1, if λ is sufficiently large, some of the coefficients are driven to zero

$$\frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2 + \frac{\lambda}{2}\sum_{j=1}^{M}|w_j|^q$$

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2 \qquad \sum_{j=1}^{M}|w_j|^q \leqslant \eta$$







Rain prediction : ML & MAP estimation

X = [1,0,0, 0, 1, 0,….]

Model ?

Likelihood : p(X l model)

Learn model parameters :

  Maximum Likelihood (ML) estimation

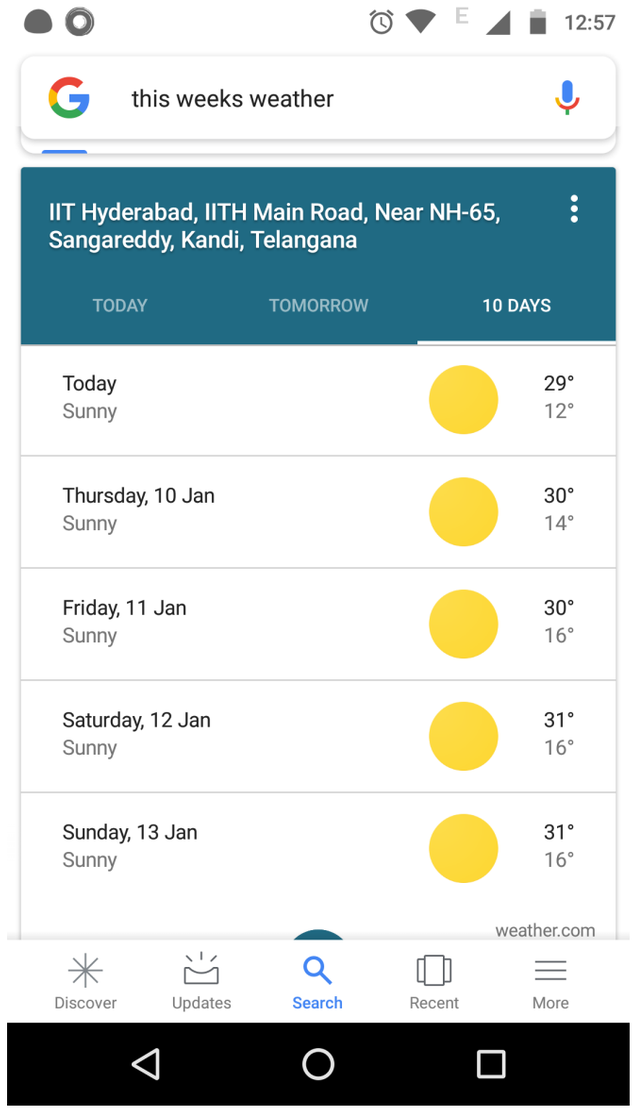
Maximum Aposteriori (MAP) estimation

$$\widehat{\Theta}_{ML} = \underset{\Theta}{\operatorname{argmax}}\ \mathcal{L}$$

$$\mathcal{L} = \sum_{\mathbf{x}_i \in \mathcal{X}} log\ prob(\mathbf{x}_i|\Theta)$$

$$\widehat{\Theta}_{MAP} = \underset{\Theta}{\operatorname{argmax}} \prod_{\mathbf{x}_i \in \mathcal{X}} prob(\mathbf{x}_i|\Theta) \cdot prob(\Theta)$$

$$\widehat{\Theta}_{MAP} = \underset{\Theta}{\operatorname{argmax}} \left(\sum_{\mathbf{x}_i \in \mathcal{X}} log\ prob(\mathbf{x}_i|\Theta) + log\ prob(\Theta) \right)$$

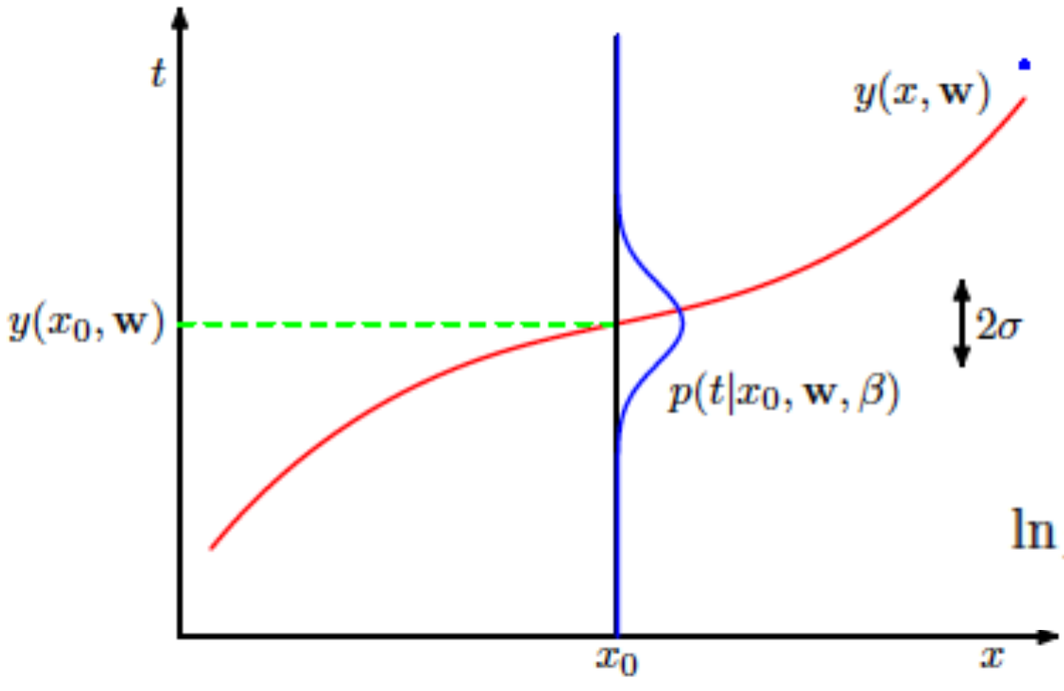


Curve fitting - A Probabilistic perspective

Least squares = maximum likelihood

- error function results from the maximum likelihood solution under an assumed Gaussian noise model

$$p(t|x, \mathbf{w}, \beta) = \mathcal{N}\left(t|y(x, \mathbf{w}), \beta^{-1}\right) \qquad t = y(\mathbf{x}, \mathbf{w}) + \epsilon \qquad \varepsilon \sim \mathcal{N}(0, \sigma_n^2).$$



data points are i.i.d

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}\left(t_n|y(x_n, \mathbf{w}), \beta^{-1}\right).$$

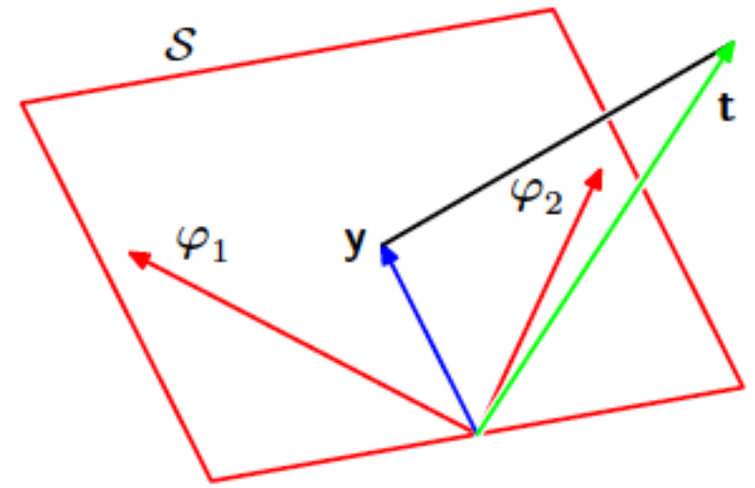
$$\ln p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = -\frac{\beta}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi).$$

Curve fitting - A Probabilistic perspective

Least squares = maximum likelihood

- error function results from the maximum likelihood solution under an assumed Gaussian noise model

$$\ln p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = -\frac{\beta}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi).$$



$$E_D(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n)\}^2.$$

$$\nabla \ln p(\mathbf{t}|\mathbf{w}, \beta) = \sum_{n=1}^{N}\left\{t_n - \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n)\right\}\phi(\mathbf{x}_n)^{\mathrm{T}}.$$

$$0 = \sum_{n=1}^{N} t_n\phi(\mathbf{x}_n)^{\mathrm{T}} - \mathbf{w}^{\mathrm{T}}\left(\sum_{n=1}^{N}\phi(\mathbf{x}_n)\phi(\mathbf{x}_n)^{\mathrm{T}}\right)$$

$$\mathbf{w}_{\mathrm{ML}} = \left(\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^{\mathrm{T}}\mathbf{t}$$

$$\mathbf{\Phi} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}$$

$$\mathbf{\Phi}^{\dagger} \equiv \left(\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^{\mathrm{T}}$$

Moore-Penrose pseudo-inverse

Regularized least squares regression = Maximum Aposteriori Estimate

- Ridge Regression

$$\widetilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n,\mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

- Compute Maximum aposteriori (MAP) estimate

- Prior over parameters

$$p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I}) = \left(\frac{\alpha}{2\pi}\right)^{(M+1)/2}\exp\left\{-\frac{\alpha}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}\right\}$$

- Posterior

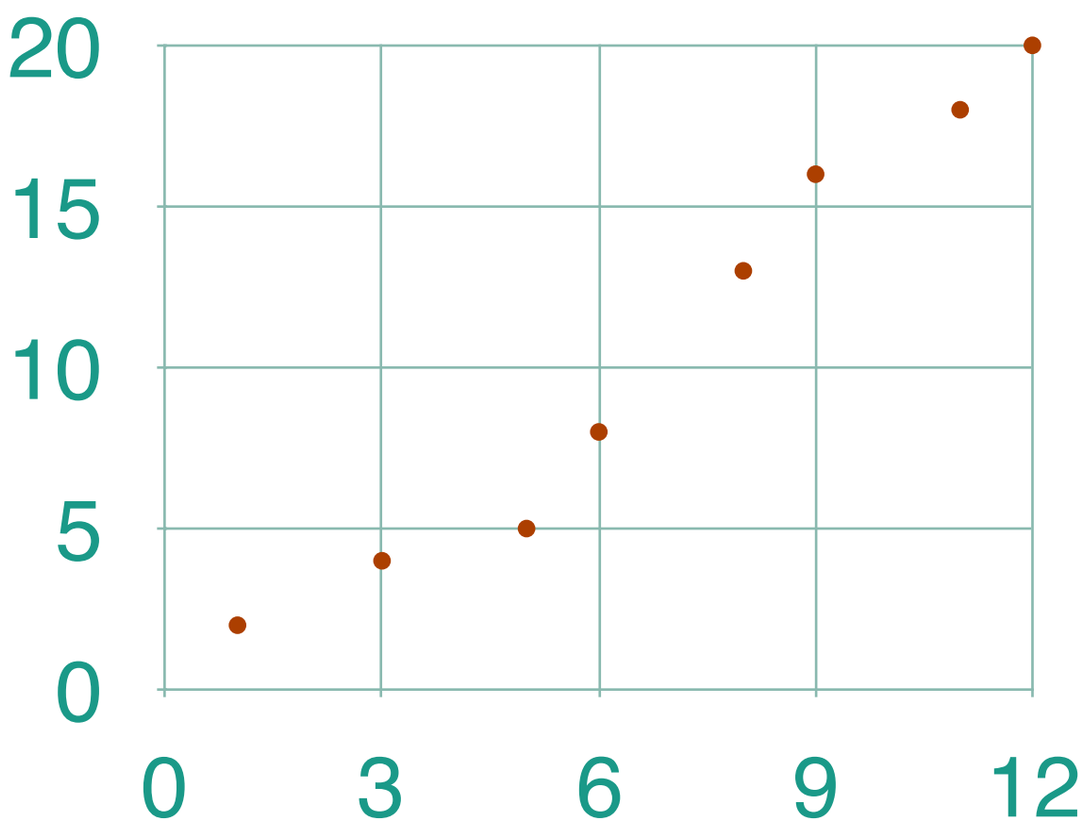
$$p(\mathbf{w}|\mathbf{x},\mathbf{t},\alpha,\beta) \propto p(\mathbf{t}|\mathbf{x},\mathbf{w},\beta)p(\mathbf{w}|\alpha).$$

- MAP estimate

$$\frac{\beta}{2}\sum_{n=1}^{N}\{y(x_n,\mathbf{w}) - t_n\}^2 + \frac{\alpha}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}.$$

$$\mathbf{w} = \left(\lambda\mathbf{I} + \mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^{\mathrm{T}}\mathbf{t}. \qquad \lambda = \alpha/\beta.$$

Bayesian Linear regression



- How to encode prior knowledge about the function e.g. slope
- Learn from limited data
- How to consider uncertainty
- Model selection
  - Which is the best function : linear, quadratic, cubic ?
  - regularisation constant
  - Cross-validation
    - Grid search is costly
    - only a faction of data (validation data) is used for model selection
- Here comes Bayesian linear regression

Bayesian learning

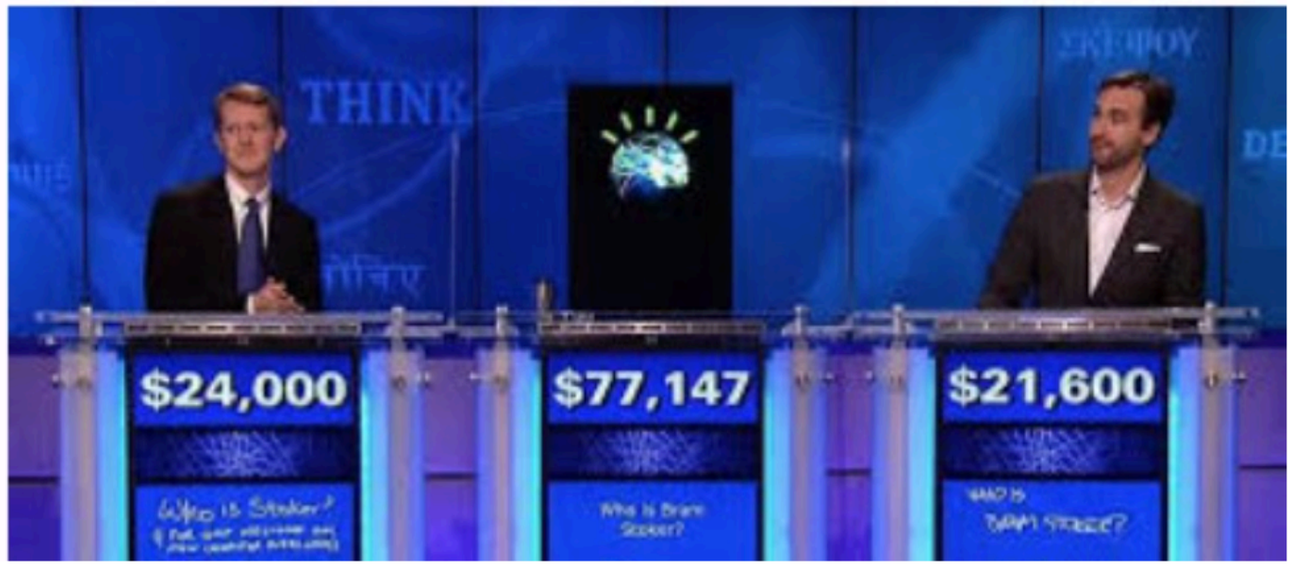
Human Inductive Learning

- how does the human mind go beyond the data of experience?
- how can it generalize well from the few noisy data ?

https://gadgets.ndtv.com/science/news/lee-sedol-scores-surprise-victory-over-googles-alphago-in-game-4-813248

Uncertainty in machine learning

- Safe AI : Uncertainty is important in Decision making
  - Important in high risk applications such as autonomous driving, disease diagnosis



Google Photos labeled black people 'gorillas'

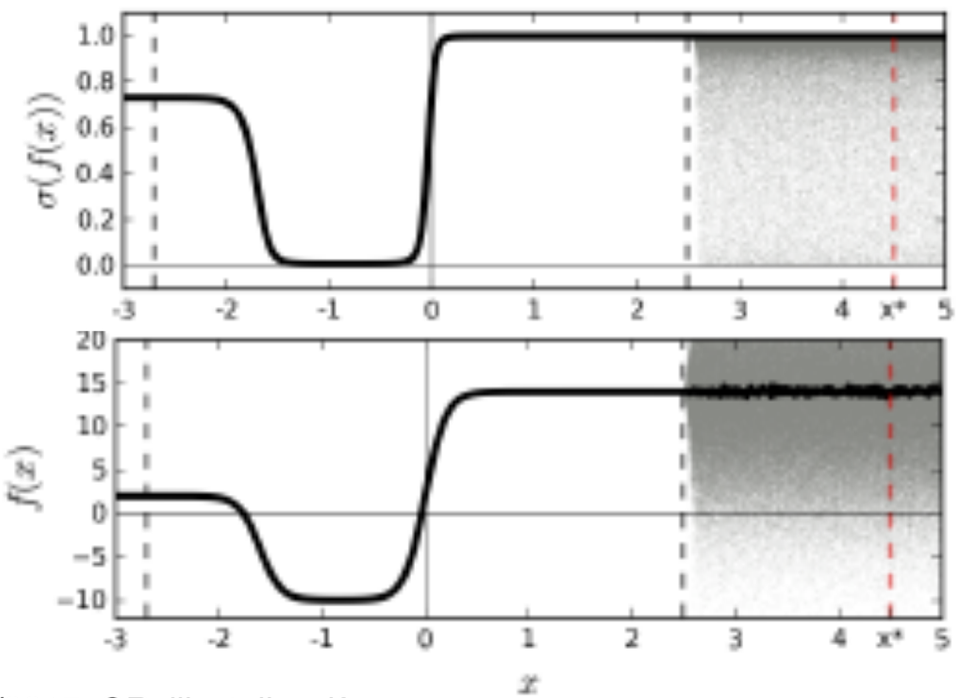
Jessica Guynn, USA TODAY Published 1:15 p.m. ET July 1, 2015 | **Updated 2:10 p.m. ET July 1, 2015**

https://www.topbots.com/10-major-fortune-500-brands-using-ibm-watson/
https://cleantechnica.com/2016/07/02/tesla-model-s-autopilot-crash-gets-bit-scary-negligent/

Uncertainty in machine learning

- Out of distribution data : Train the model on dogs breeds and provide picture of cat

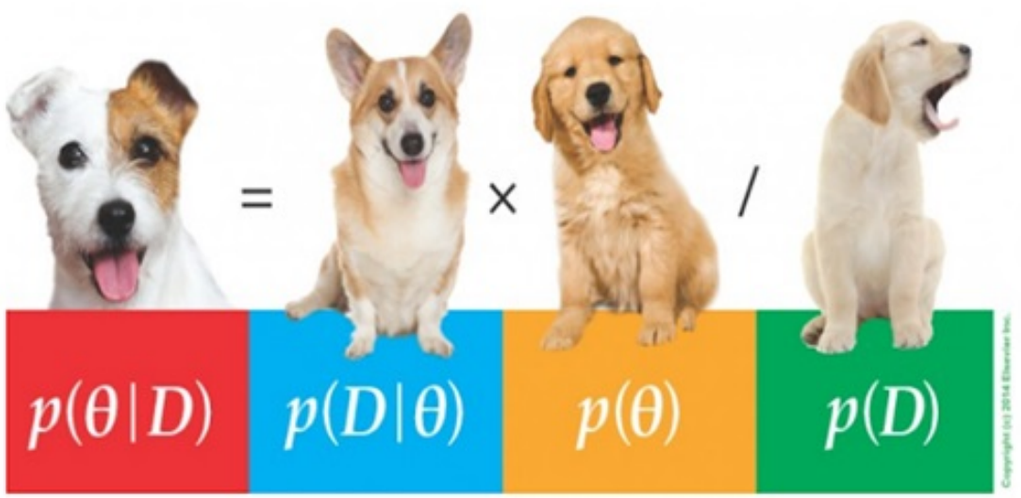
- Uncertainty provides an idea on what the model actually know about the data

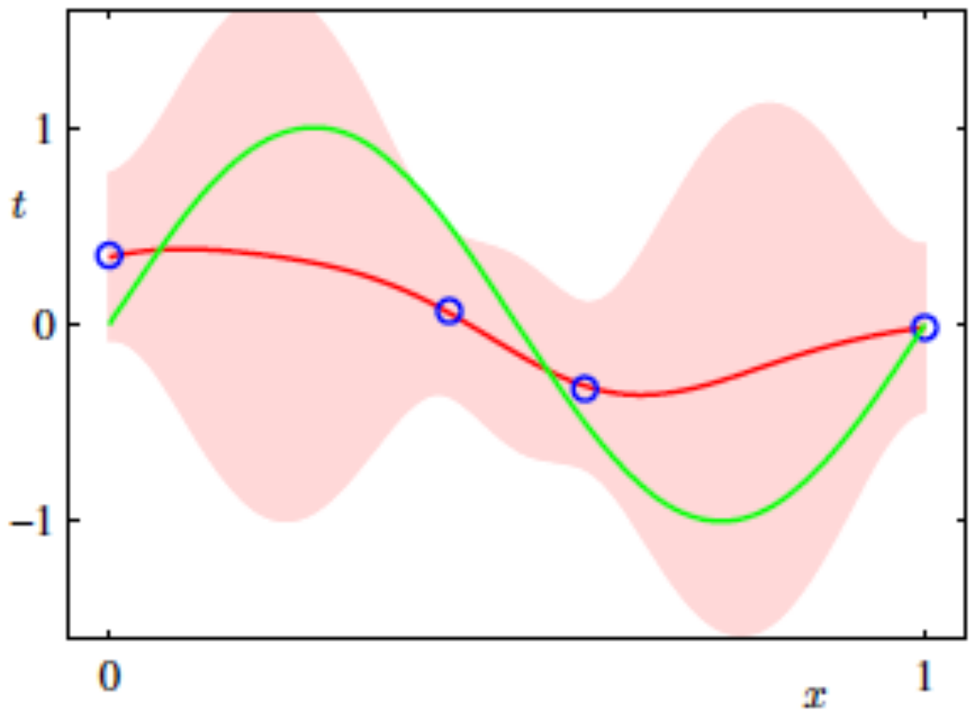


https://www.cs.ox.ac.uk/people/yarin.gal/website/PDFs/2017_OReilly_talk.pdf

Bayesian learning

- Bayesian learning helps in better generalization capability and decision making by considering domain knowledge and by providing uncertainty estimates.





$$prob(\tilde{\mathbf{x}}|\mathcal{X}) = \int_{\Theta} prob(\tilde{\mathbf{x}}|\Theta) \cdot prob(\Theta|\mathcal{X})\ d\Theta$$

- Bayesian linear regression, Gaussian processes, Latent Dirichlet allocation (LDA), Dirichlet Processes

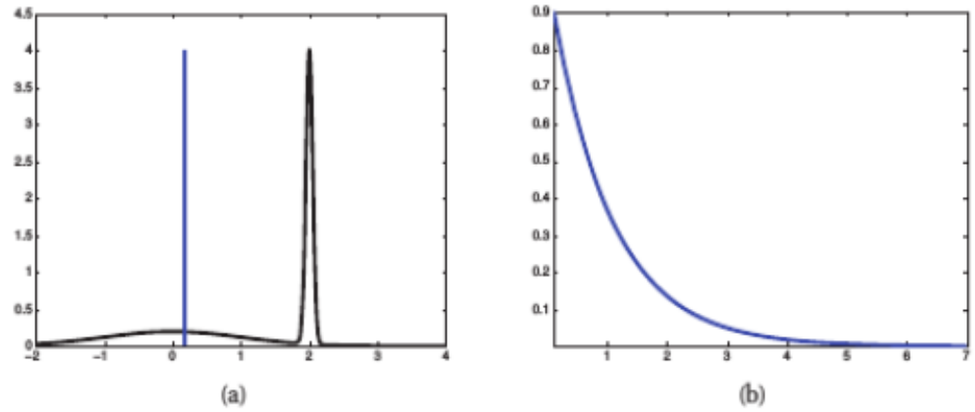
Rain prediction : Bayesian estimation

Both ML and MAP return only single and specific values for the parameter $\ominus$.

$$prob(\ominus|\mathcal{X}) = \frac{prob(\mathcal{X}|\ominus) \cdot prob(\ominus)}{prob(\mathcal{X})}$$

$$prob(\mathcal{X}) = \int_{\ominus} prob(\mathcal{X}|\ominus) \cdot prob(\ominus) \; d\ominus$$

$$prob(\tilde{\mathbf{x}}|\mathcal{X}) = \int_{\ominus} prob(\tilde{\mathbf{x}}|\ominus) \cdot prob(\ominus|\mathcal{X}) \; d\ominus$$



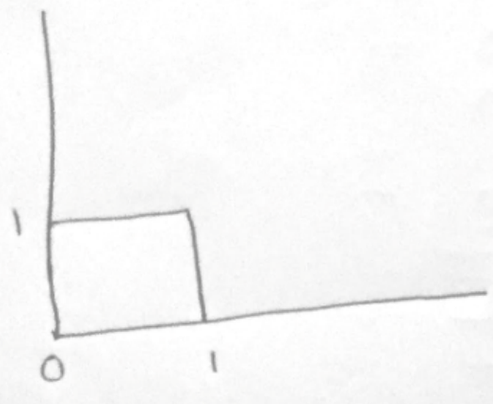
ML, MAP, Bayesian Estimate : Rain prediction

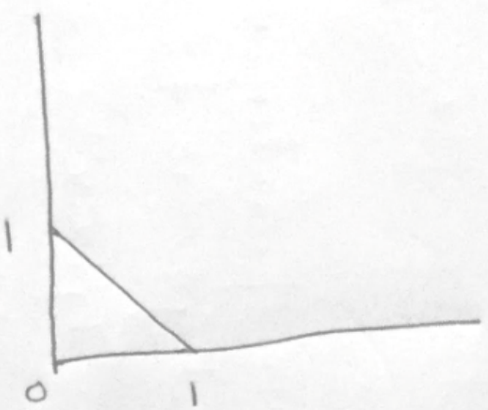
X : BINARY R.V.

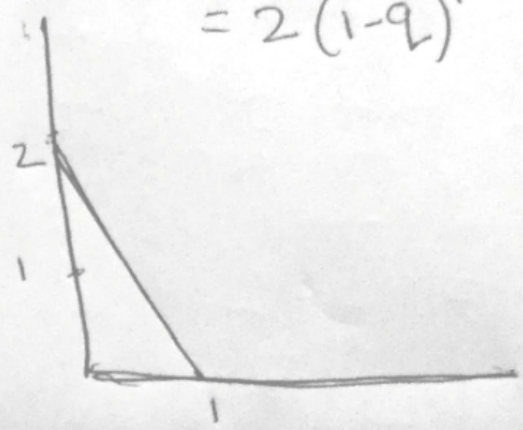
$X \sim$ BERNOULLI (q) $\quad P(X|q) = q^X (1-q)^{1-X}$

PRIOR $\quad$ LIKELIHOOD $\quad$ POSTERIOR

$P(q) \sim \text{UNIF}(0,1)$ $\quad P(X=0|q) = 1-q$ $\quad P(q|X=0) = \dfrac{P(X=0|q)\,P(q)}{P(X)} = 2(1-q)$







PREDICTION : $P(x^* = 1 \mid x) = \int P(x^* = 1 \mid q)\, P(q|x)\, dq$

Bayesian Machine learning : Bayesian linear regression

Learn $f : \mathcal{X} \to \mathcal{Y}$.

Bayesian approach : Allows to encode prior belief over functions

Parametric model : $f(x) = w.x$

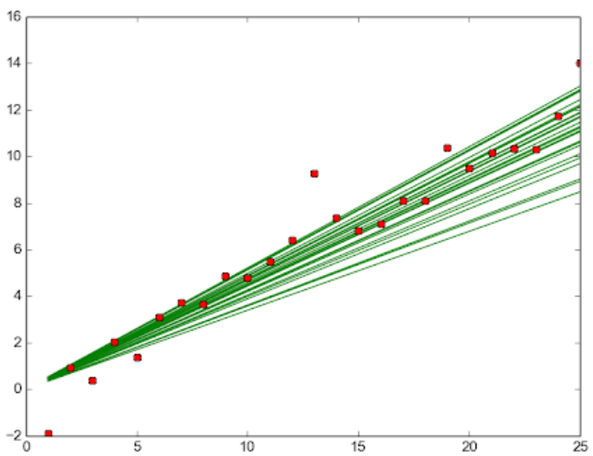
$$p(w|D) \propto p(D|w)\ p(w)$$

Posterior $\propto$ Likelihood Prior

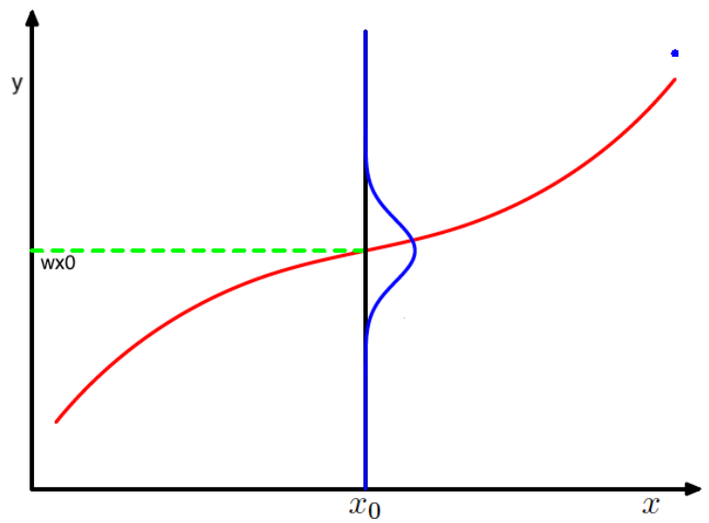
$$f(\mathbf{x}) = \mathbf{x}^\top \mathbf{w}, \qquad y = f(\mathbf{x}) + \varepsilon,$$

$$\varepsilon \sim \mathcal{N}(0, \sigma_n^2).$$

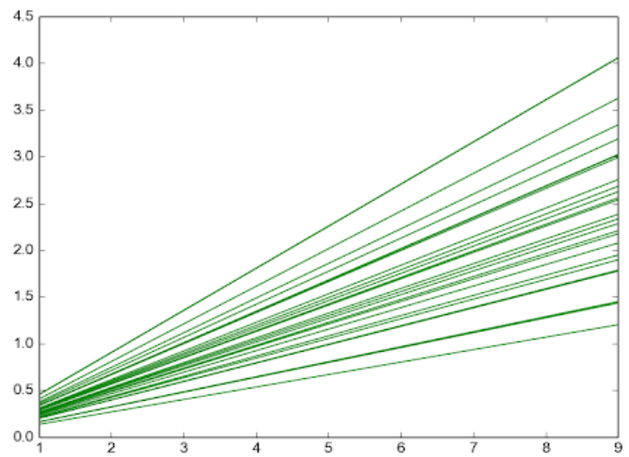
$$p(w|y, x) \propto p(y|w, x)p(w)$$



w ~ N(w; 0.45, 0.05)



$$p(y|w, x) = N(y; wx, \sigma^2)$$



w ~ N(w; 0.3, 0.1)

Bayesian linear regression

- Prior : $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \Sigma_p).$
- Likelihood : $p(\mathbf{y}|X, \mathbf{w}) = \prod_{i=1}^{n} p(y_i|\mathbf{x}_i, \mathbf{w}) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left(-\frac{(y_i - \mathbf{x}_i^\top \mathbf{w})^2}{2\sigma_n^2}\right)$

$$= \frac{1}{(2\pi\sigma_n^2)^{n/2}} \exp\left(-\frac{1}{2\sigma_n^2}|\mathbf{y} - X^\top \mathbf{w}|^2\right) = \mathcal{N}(X^\top \mathbf{w}, \sigma_n^2 I),$$

- Maximum likelihood estimate $w_{ML} = (XX^\top)^{-1}Xy$
- Maximum a-posteriori estimate $\bar{\mathbf{w}} = \sigma_n^{-2}(\sigma_n^{-2}XX^\top + \Sigma_p^{-1})^{-1}X\mathbf{y}$

Bayesian linear regression

- Prior :

$$\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \Sigma_p).$$

- Likelihood :

$$p(\mathbf{y}|X, \mathbf{w}) = \prod_{i=1}^{n} p(y_i|\mathbf{x}_i, \mathbf{w}) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left(-\frac{(y_i - \mathbf{x}_i^\top \mathbf{w})^2}{2\sigma_n^2}\right)$$
$$= \frac{1}{(2\pi\sigma_n^2)^{n/2}} \exp\left(-\frac{1}{2\sigma_n^2}|\mathbf{y} - X^\top \mathbf{w}|^2\right) = \mathcal{N}(X^\top \mathbf{w}, \sigma_n^2 I),$$

- Posterior :

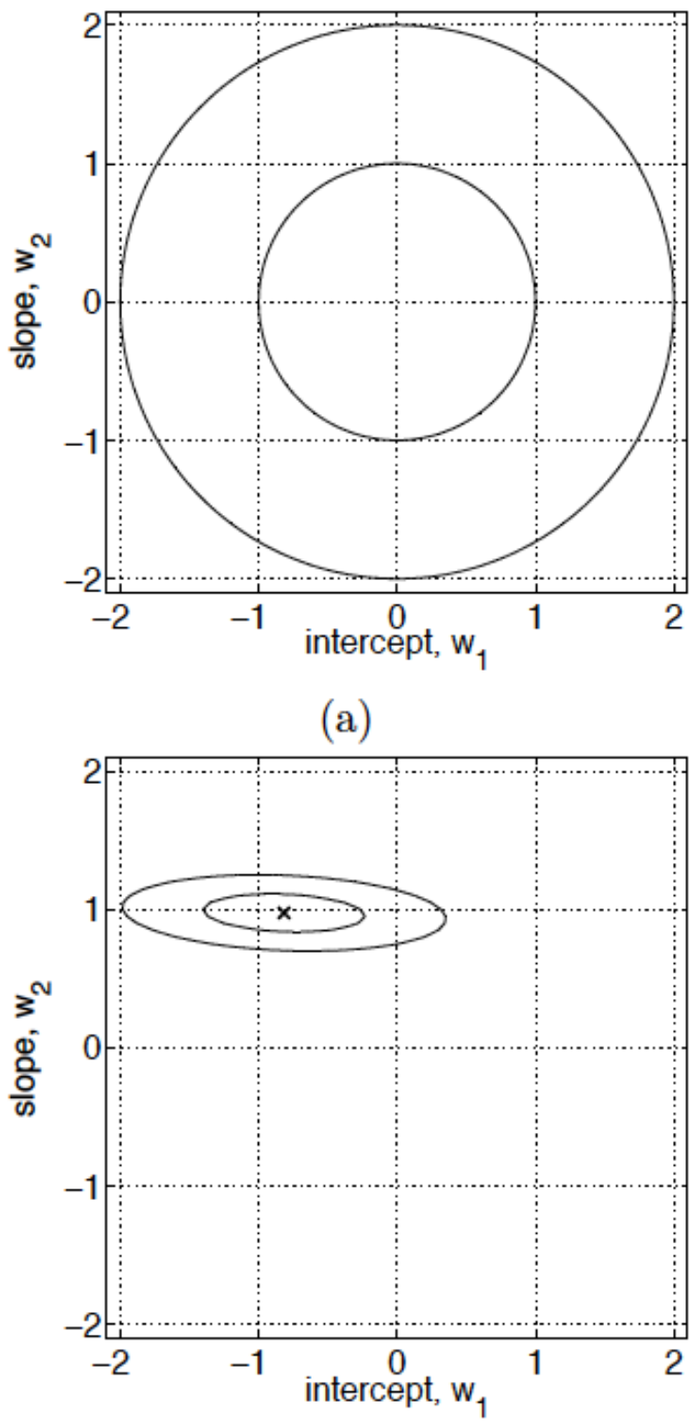
$$p(\mathbf{w}|X, \mathbf{y}) \propto \exp\left(-\frac{1}{2\sigma_n^2}(\mathbf{y} - X^\top \mathbf{w})^\top (\mathbf{y} - X^\top \mathbf{w})\right) \exp\left(-\frac{1}{2}\mathbf{w}^\top \Sigma_p^{-1} \mathbf{w}\right)$$
$$\propto \exp\left(-\frac{1}{2}(\mathbf{w} - \bar{\mathbf{w}})^\top \left(\frac{1}{\sigma_n^2} X X^\top + \Sigma_p^{-1}\right)(\mathbf{w} - \bar{\mathbf{w}})\right)$$

$$p(\mathbf{w}|X, \mathbf{y}) \sim \mathcal{N}(\bar{\mathbf{w}} = \frac{1}{\sigma_n^2} A^{-1} X \mathbf{y},\ A^{-1}), \quad A = \sigma_n^{-2} X X^\top + \Sigma_p^{-1}.$$

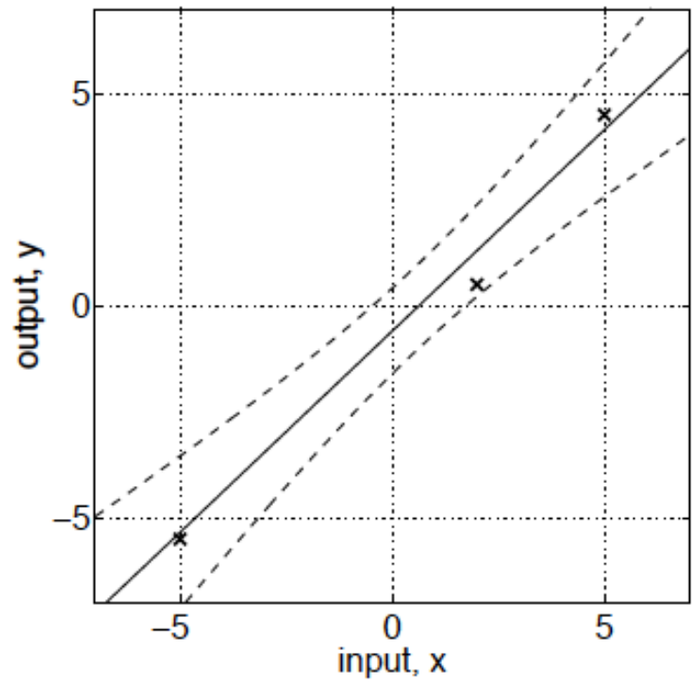
- Mean, MAP (prior mean 0) and ML estimates (prior variance large)

$$f(x) = w_1 + w_2 x$$

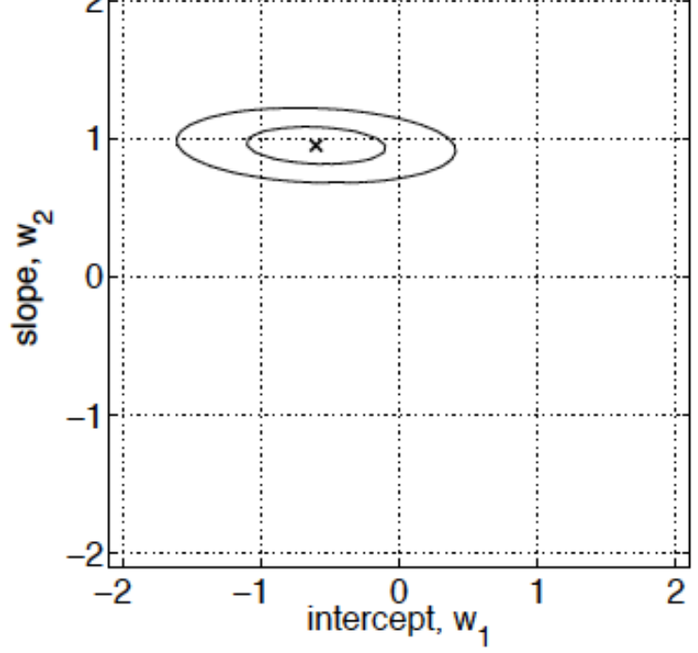
$$p(\mathbf{w}) \sim \mathcal{N}(\bar{0}, \bar{I}),$$



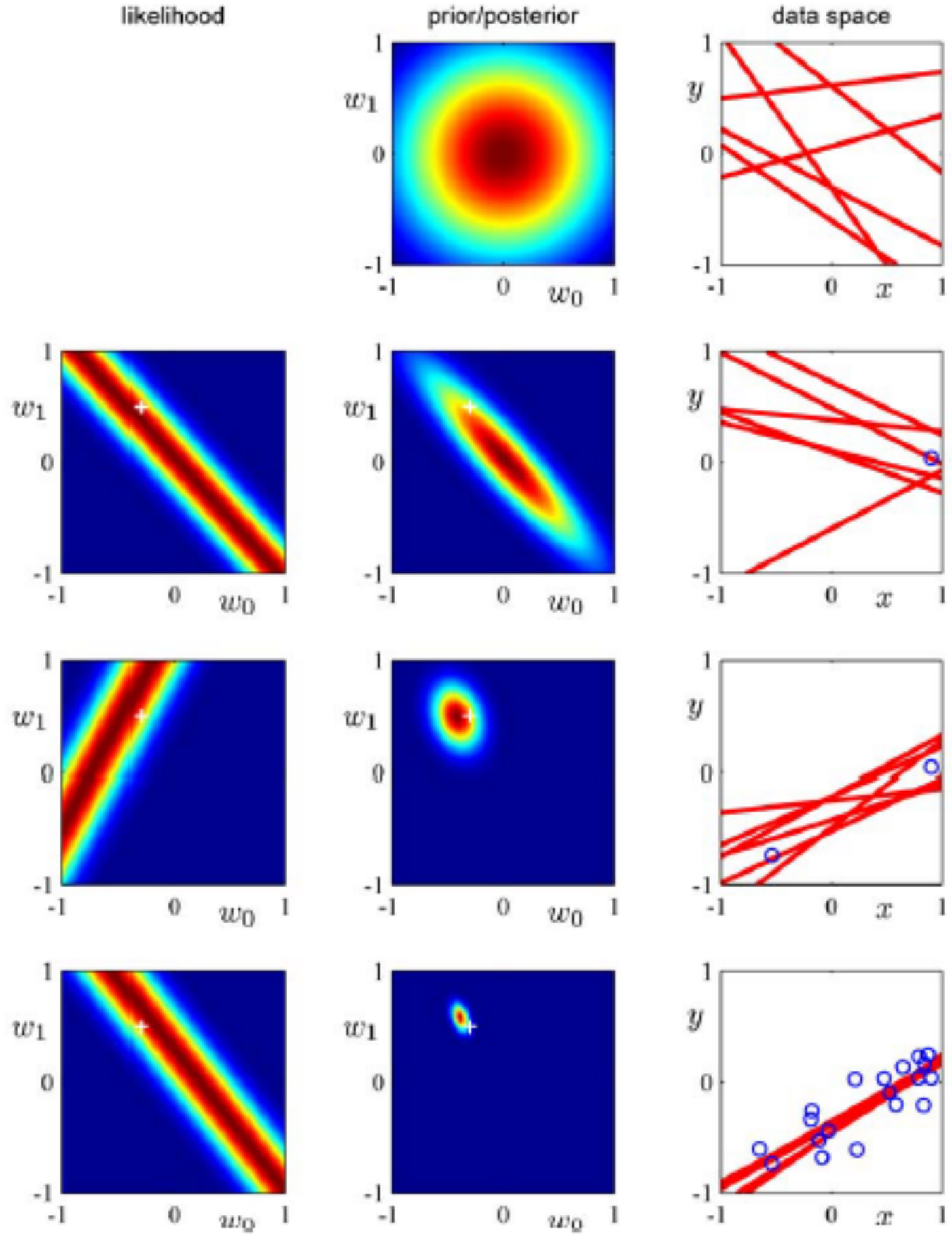
(a)



(b)



- Sequential update with Bayesian linear regression



Bayesian linear regression : Prediction

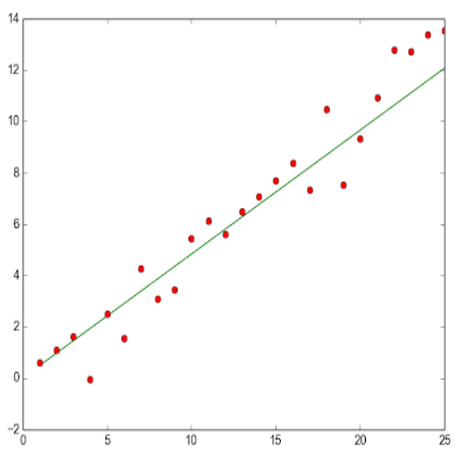
$$Prediction : p(y_*|x_*) = \int p(y_*|x_*, w)p(w|D)dw$$

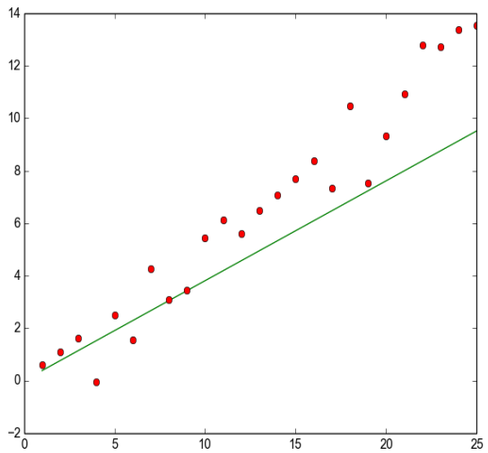
$$\begin{aligned} p(f_*|\mathbf{x}_*, X, \mathbf{y}) &= \int p(f_*|\mathbf{x}_*, \mathbf{w})p(\mathbf{w}|X, \mathbf{y})\, d\mathbf{w} \\ &= \mathcal{N}\big(\frac{1}{\sigma_n^2}\mathbf{x}_*^\top A^{-1} X\mathbf{y},\ \mathbf{x}_*^\top A^{-1}\mathbf{x}_*\big). \end{aligned}$$

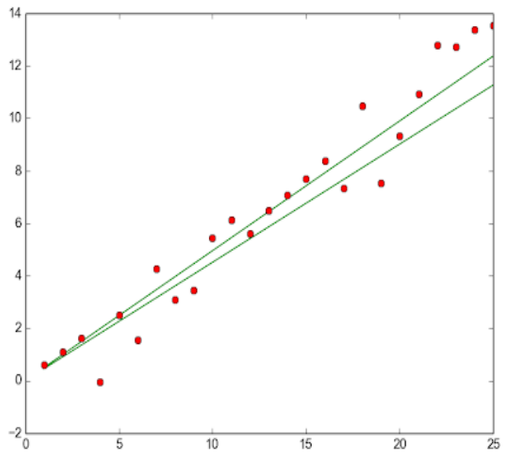
$$A = \sigma_n^{-2} X X^\top + \Sigma_p^{-1}.$$

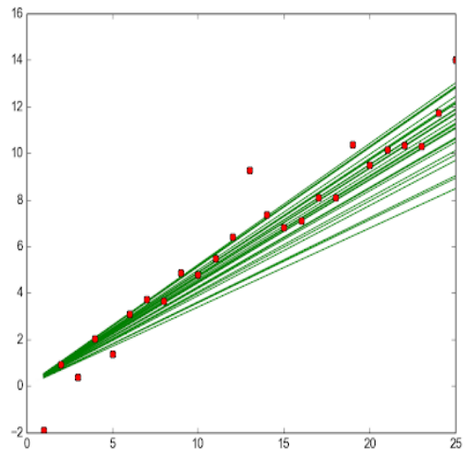
$$N(x_*^\top \Sigma_p X (X^\top \Sigma_p X + \sigma_n^2 I)^{-1} y,$$

$$x_*^\top \Sigma_p x_* - x_* \Sigma_p X (X^\top \Sigma_p X + \sigma_n^2 I)^{-1} X^\top \Sigma_p x_*)$$

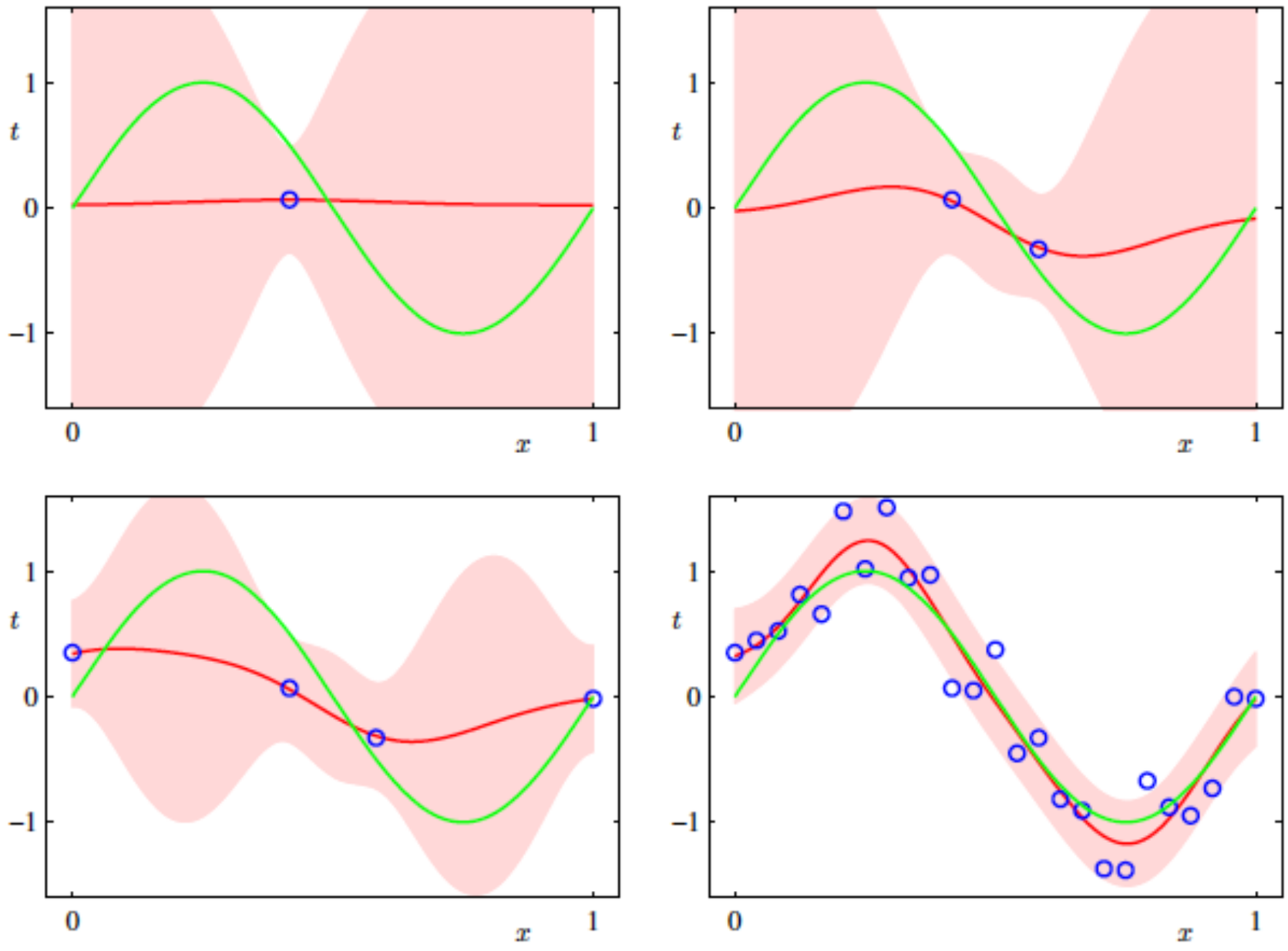








Bayesian linear regression



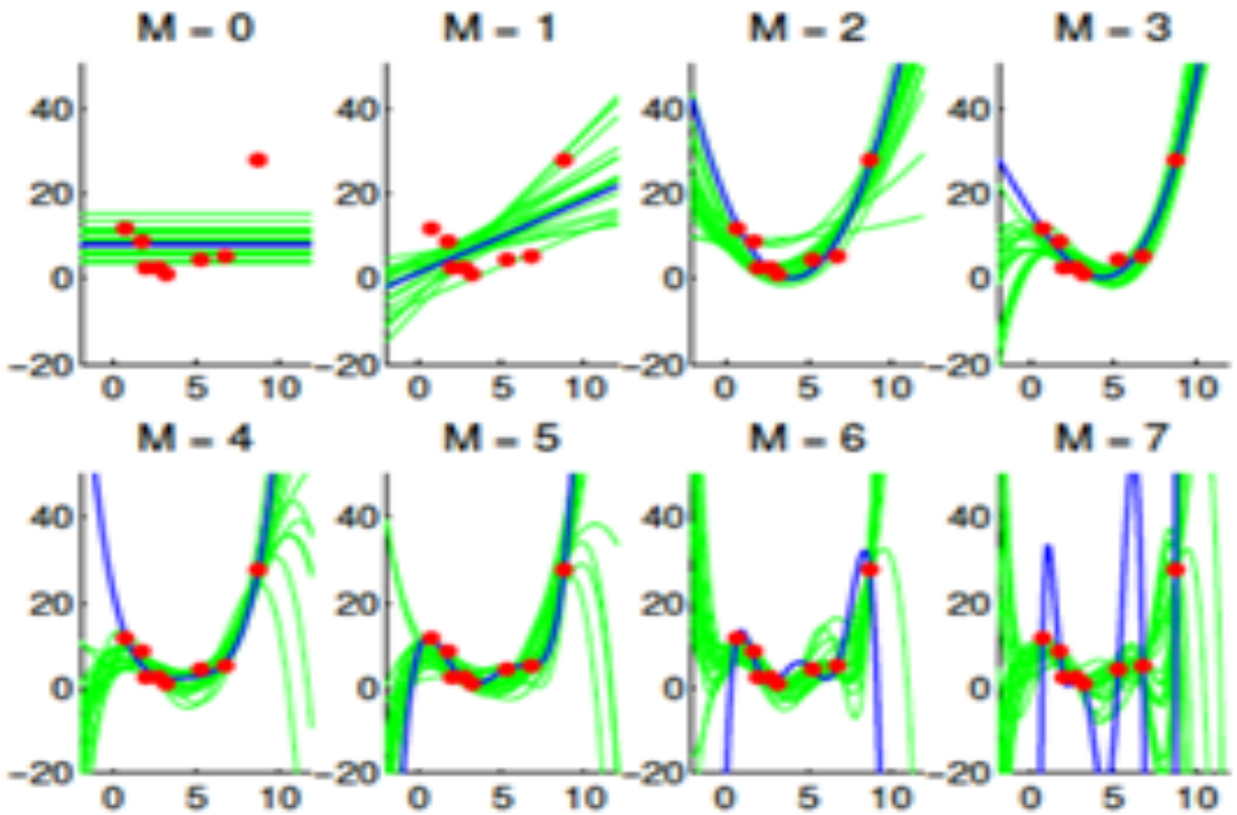
Bayesian linear regression : Model selection

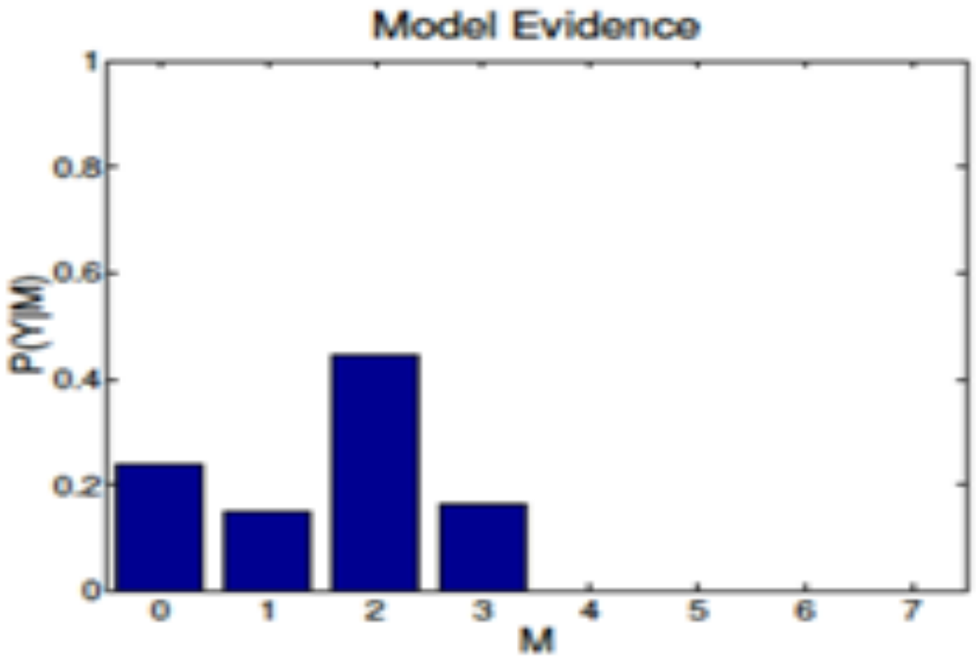
$$p(w|D,M) = \frac{p(D|w,M)p(w|M)}{p(D|M)}$$

$$ModelSelection: maximize \quad evidence \quad p(D|M) = \int p(D|w,M)p(w|M)dw$$

$$p(y|x,M) = N(y;0,XX^\top + \sigma^2 I)$$

$$logp(y|x,M) = -\frac{1}{2}y(XX^\top + \sigma^2 I)^{-1}y - \frac{1}{2}\log|(XX^\top + \sigma^2 I)|$$





Bayesian ML

Treat parameters as random variables to capture model uncertainty
Provides a framework to sequentially update belief about the model/parameters

- Incorporate domain knowledge
  - Through prior
- Modelling uncertainty for safe AI
  - through predictive variance
- Automate machine learning
  - through Evidence
  - Model selection from full data
- Generalize from small data
  - Bayesian model averaging prevents overfitting
- Sequential/online learning
  - Posterior as Prior

Bayesian Machine Learning

Everything follows from two simple rules:
Sum rule: $P(x) = \sum_y P(x, y)$
Product rule: $P(x, y) = P(x)P(y|x)$

$$P(\theta|\mathcal{D}) = \frac{P(\mathcal{D}|\theta)P(\theta)}{P(\mathcal{D})}$$

$P(\mathcal{D}|\theta)$ likelihood of θ
$P(\theta)$ prior probability of θ
$P(\theta|\mathcal{D})$ posterior of θ given $\mathcal{D}$

Prediction:

$$P(x|\mathcal{D}, m) = \int P(x|\theta, \mathcal{D}, m)P(\theta|\mathcal{D}, m)d\theta$$

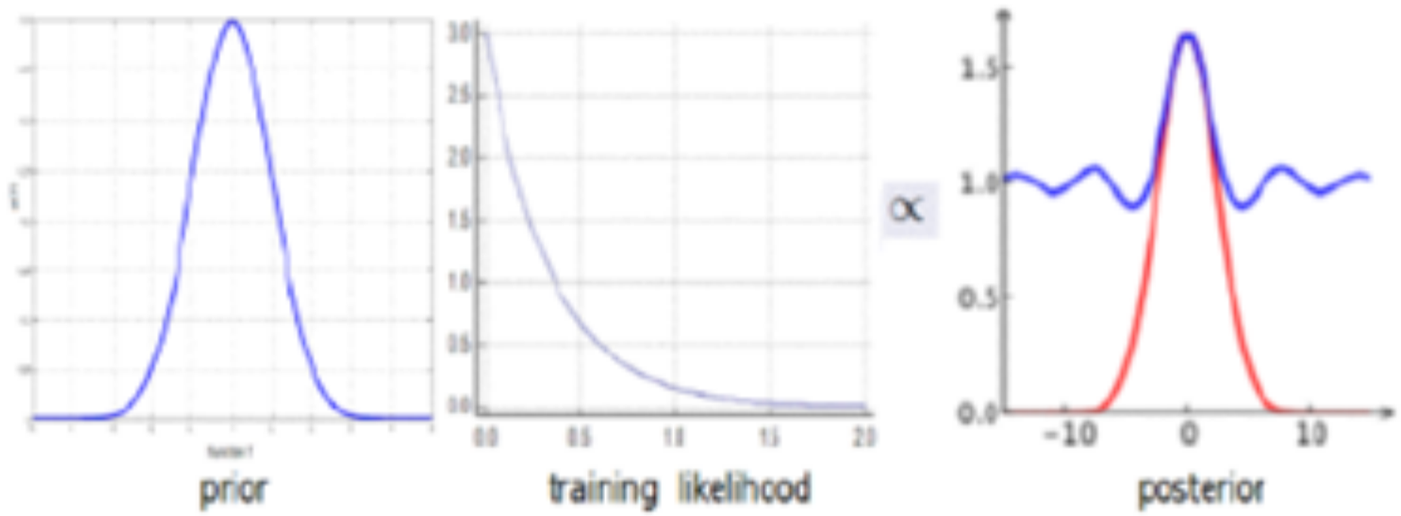
Model Comparison:

$$P(m|\mathcal{D}) = \frac{P(\mathcal{D}|m)P(m)}{P(\mathcal{D})}$$

$$P(\mathcal{D}|m) = \int P(\mathcal{D}|\theta, m)P(\theta|m)\, d\theta$$

Bayesian machine learning : Challenges

- Choice of prior
- Obtaining the posterior distribution
- What if likelihood and prior are non-conjugates ?
- Bayesian logistic regression
  - Monte-carlo techniques
  - Laplace Approximation
  - Variational Inference
  - Expectation Propagation



References

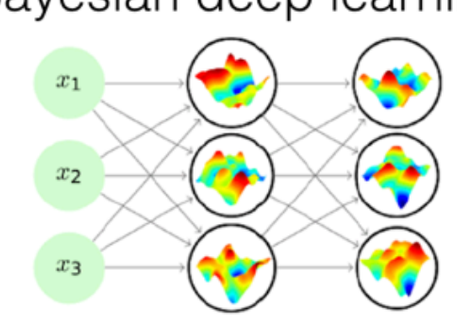
- Pattern Recognition and Machine learning by Christopher Bishop
- Probabilistic machine learning by Kevin Murphy

Bayesian Reasoning And INference (BRAIN)

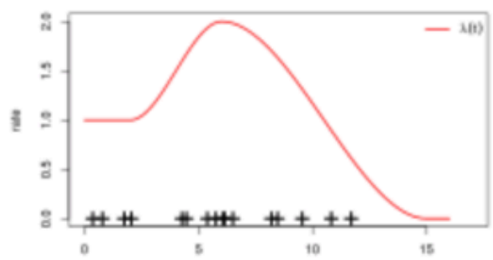
- To understand human learning process and develop machine learning models for artificial intelligence and data science

Models

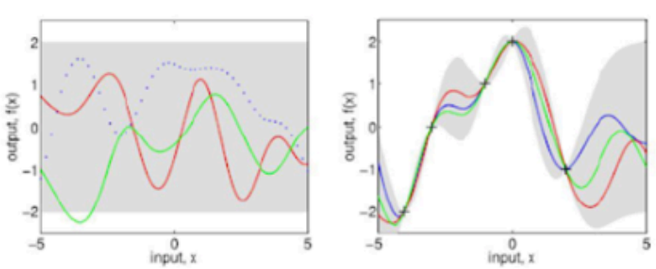
Bayesian deep learning



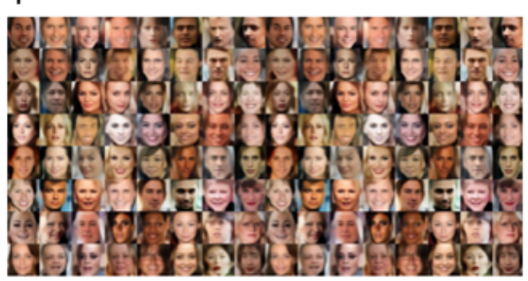
Point Processes



Bayesian non-parametric



Implicit Generative Model



Applications

Vision and Language

Traffic data analysis

social media analysis

Astrophysical data analysis

https://sites.google.com/view/brainiith/home