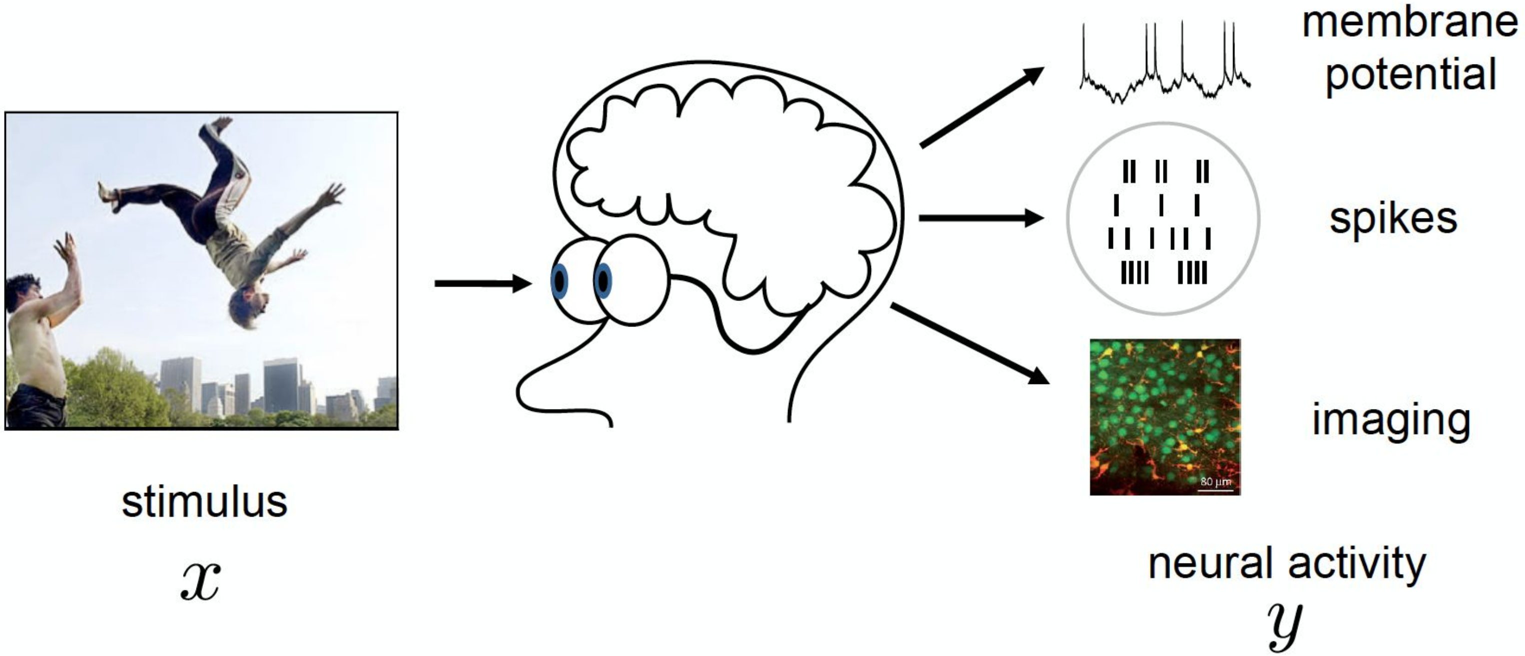
Advanced Theory Seminar
Linear and non-linear regression

Juri Minxha

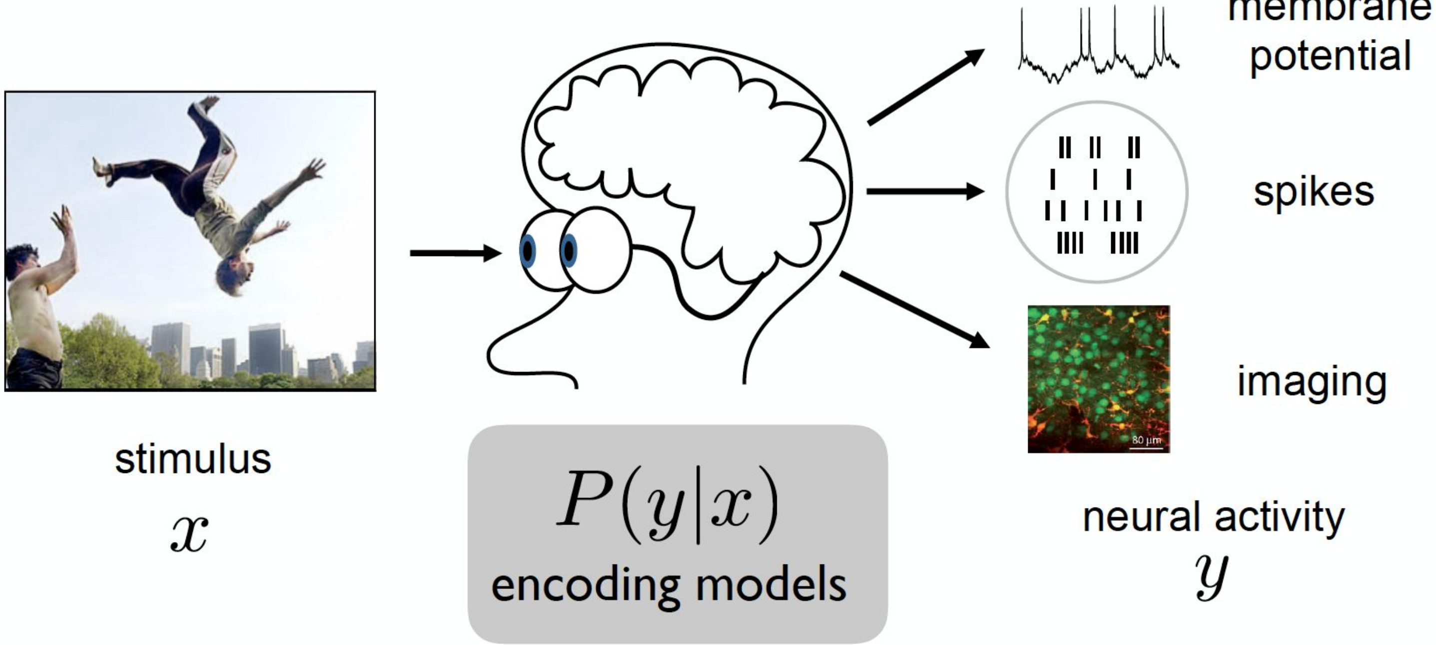
March 18, 2020

Sources

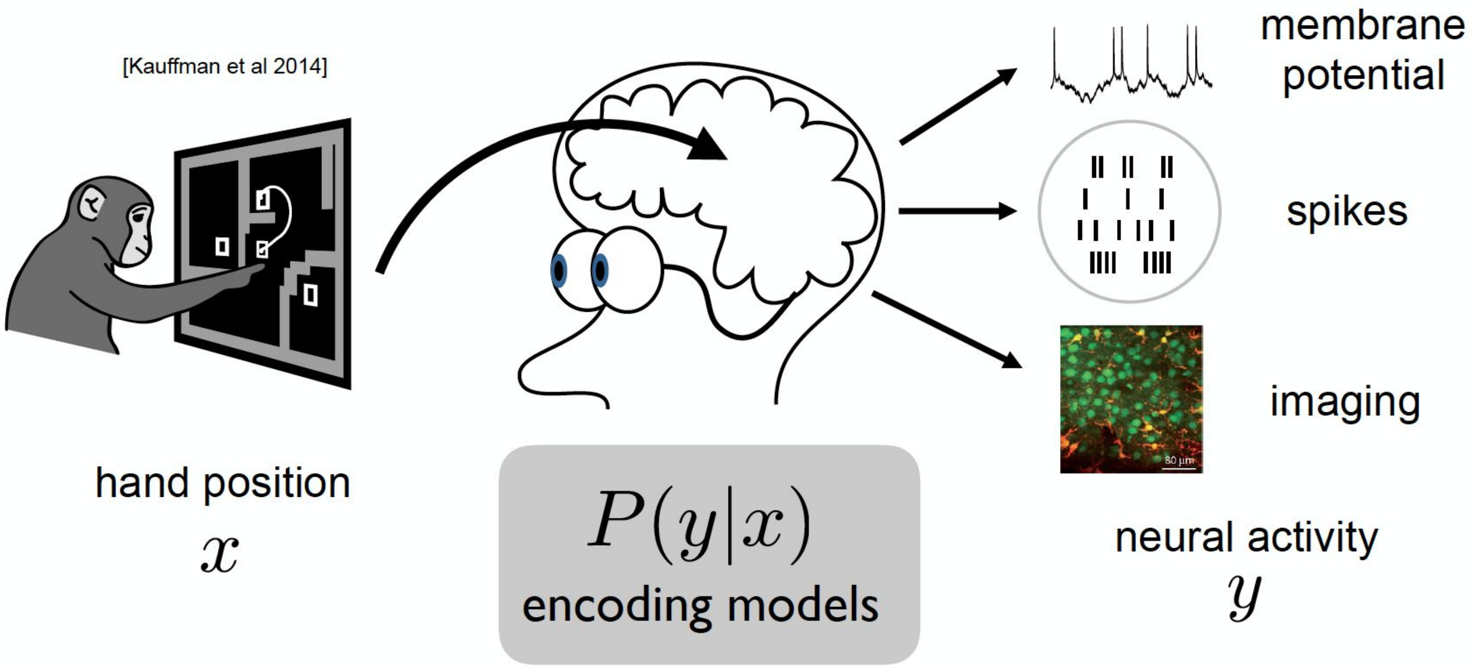
1. *Pattern recognition and Machine Learning*, **Christopher M. Bishop**
    a. predominantly chapter 3: Linear Models for Regression
2. *Statistical Models for Neural Data: from Regression/GLMs to Latent Variables*, **Jonathan Pillow**
    a. Cosyne 2018 tutorial
3. *Machine learning: A probabilistic perspective,* **Kevin Murphy**
    a. predominantly chapter 7
4. *mathematicalmonk* lectures on Youtube (highly recommend), **Jeff Miller**
5. CS 155: Machine Learning & Data Mining, **Yisong Yue**



- How are stimuli and actions encoded in neural activity?
- What aspects of neural activity carry information?



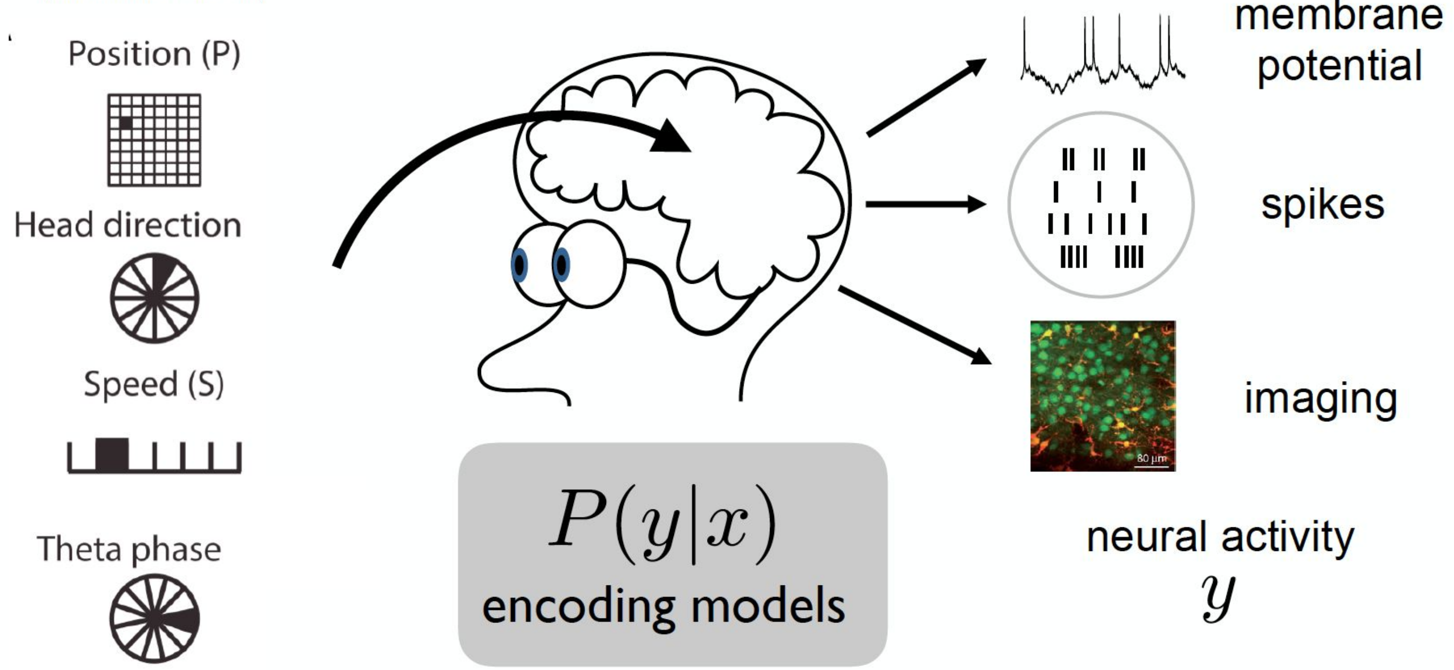
Approach:
- develop flexible statistical models of P(y|x)
- quantify information carried in neural responses



“regression models”

- not restricted to sensory variables

[Hardcastle et al 2015]

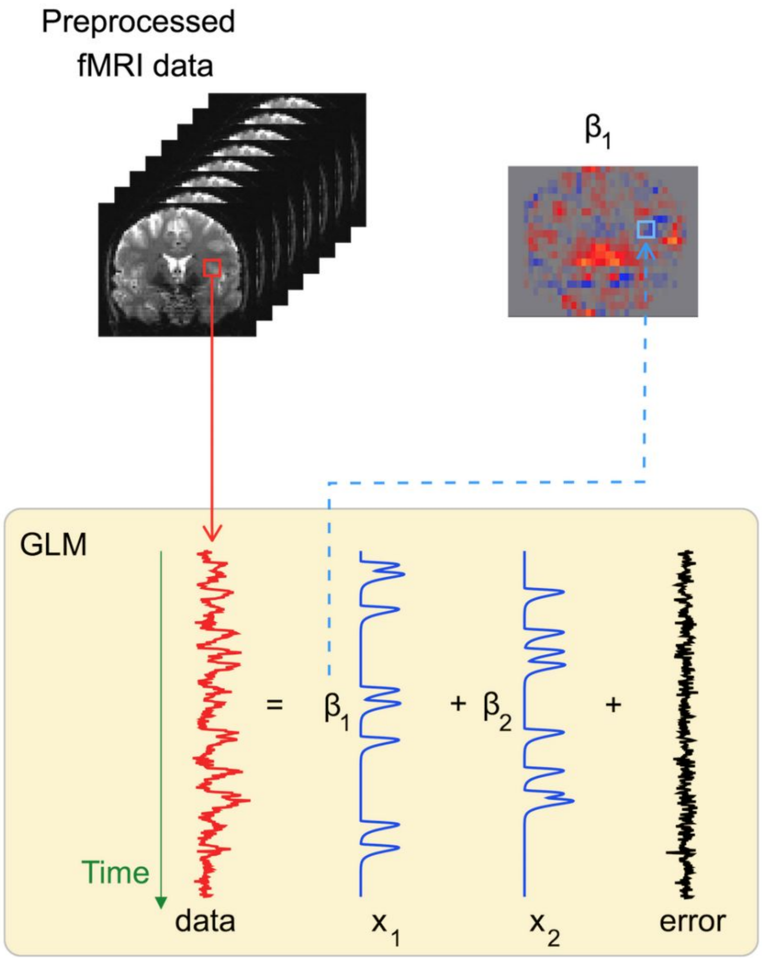


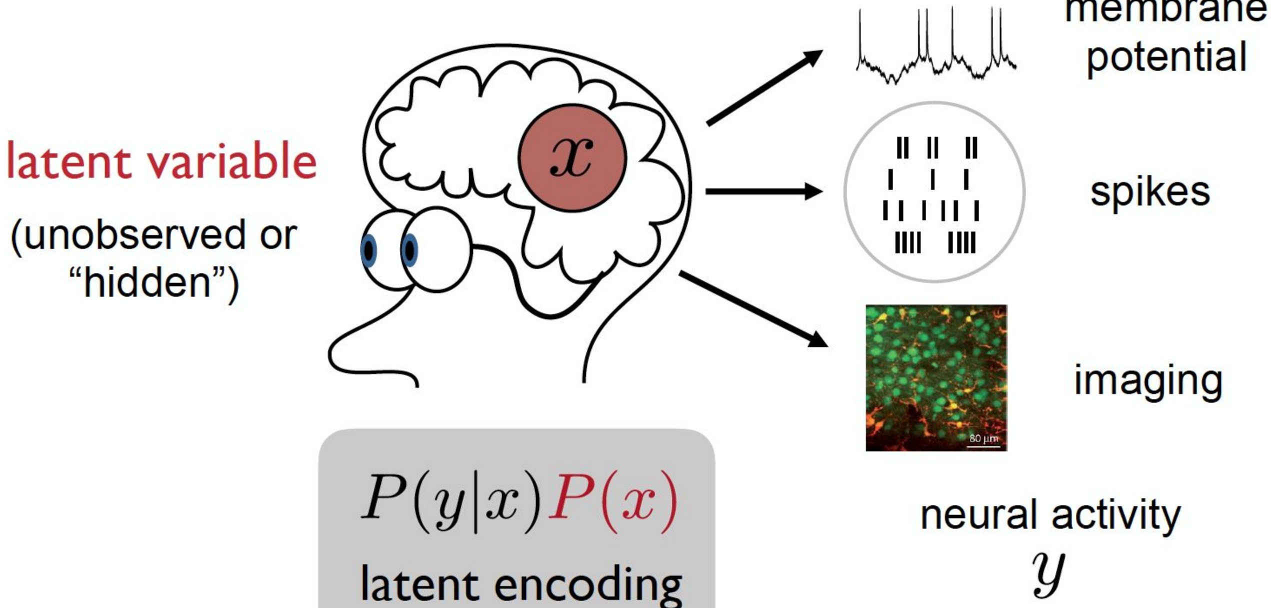
x

"external covariates"

"regression models"

- not restricted to sensory variables

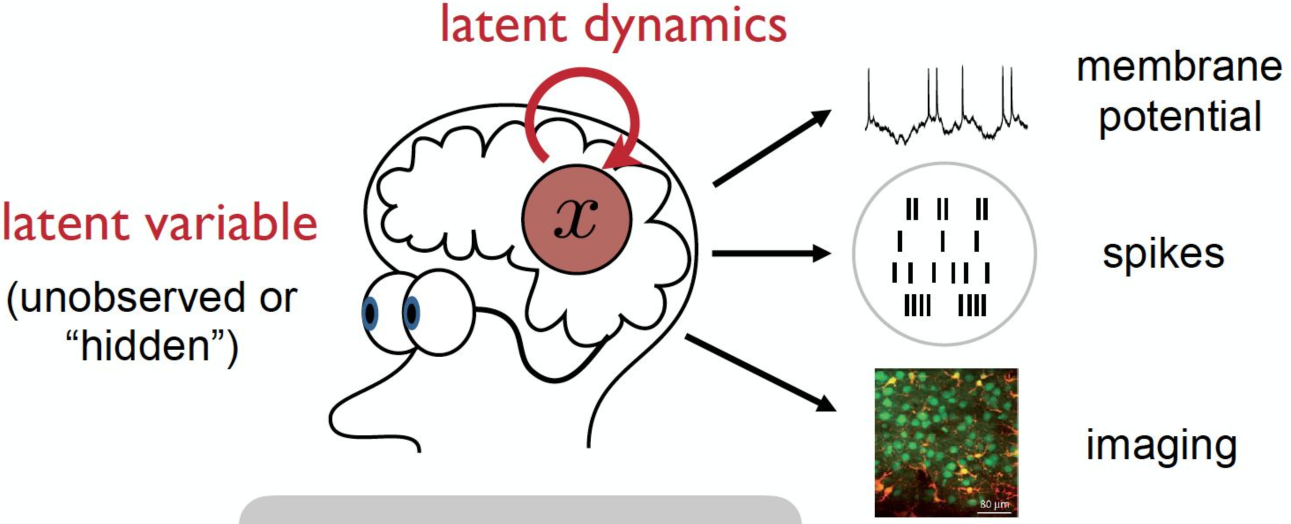

Preprocessed
fMRI data
β1
GLM
= β1 + β2 +
Time
data
x1
x2
error



$$P(y|x)P(x)$$

latent encoding models

- capture hidden structure underlying neural activity
  (eg. low-dimensional or discrete states)



$$P(y_t|x_t)P(x_t|x_{t-1})$$

latent dynamical
encoding models

neural activity
y

- capture hidden dynamics underlying neural activity

normative theories
(e.g. “efficient coding”)

Why does the code
take this form?

descriptive
statistical models

$$P(y|x)$$

What is the code?

anatomy,
biophysics

How is it implemented?

Topics

1. Introduction
2. General regression framework
   a. data, model, cost function, fitting procedures
3. Linear models
4. Maximum likelihood and least squares
5. Bayesian linear regression
6. Regularization
7. Bias-variance trade-off

General framework

Supervised setting:

1. Data $x_i \in \mathbb{R}^d$ i = 1,2,3,4…..N

General framework

Supervised setting:

1. Data $x_i \in \mathbb{R}^d$ i = 1,2,3,4…..N

2. Data transformation $\phi_j(\mathbf{x})$, here we will use $\phi(\mathbf{x}) = \mathbf{x}$. for simplicity.

General framework

Supervised setting:

1. Data $x_i \in \mathbb{R}^d$ i = 1,2,3,4…..N
2. Data transformation $\phi_j(\mathbf{x})$, here we will use $\phi(\mathbf{x}) = \mathbf{x}$ for simplicity.
3. Targets/Labels $y_i \in \mathbb{R}$ i = 1,2,3,4…..N

General framework

Supervised setting:

1. Data $x_i \in \mathbb{R}^d$ i = 1,2,3,4…..N
2. Data transformation $\phi_j(\mathbf{x})$, here we will use $\phi(\mathbf{x}) = \mathbf{x}$ for simplicity.
3. Targets/Labels $y_i \in \mathbb{R}$ i = 1,2,3,4…..N
4. Goal is to select $f : \mathbb{R}^d \rightarrow \mathbb{R}$ such that we can predict y from new x

General framework

Supervised setting:

1. Data $x_i \in \mathbb{R}^d$ i = 1,2,3,4…..N
2. Data transformation $\phi_j(\mathbf{x})$, here we will use $\boldsymbol{\phi}(\mathbf{x}) = \mathbf{x}$ for simplicity.
3. Targets/Labels $y_i \in \mathbb{R}$ i = 1,2,3,4…..N
4. Goal is to select $f : \mathbb{R}^d \to \mathbb{R}$ such that we can predict y from new x
5. Model class $f(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x})$

General framework

Supervised setting:

1. Data $x_i \in \mathbb{R}^d$ i = 1,2,3,4…..N
2. Data transformation $\phi_j(\mathbf{x})$, here we will use $\phi(\mathbf{x}) = \mathbf{x}$ for simplicity.
3. Targets/Labels $y_i \in \mathbb{R}$ i = 1,2,3,4…..N
4. Goal is to select $f : \mathbb{R}^d \to \mathbb{R}$ such that we can predict y from new x
5. Model class $$f(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \phi(\mathbf{x})$$
6. Model parameters w_j

General framework

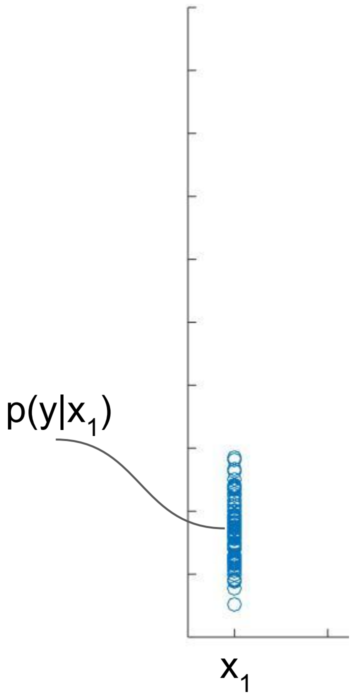
Supervised setting:

1. Data $x_i \in \mathbb{R}^d$ i = 1,2,3,4…..N
2. Data transformation $\phi_j(\mathbf{x})$, here we will use $\phi(\mathbf{x}) = \mathbf{x}$ for simplicity.
3. Targets/Labels $y_i \in \mathbb{R}$ i = 1,2,3,4…..N
4. Goal is to select $f : \mathbb{R}^d \to \mathbb{R}$ such that we can predict y from new x
5. Model class $$f(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \phi(\mathbf{x})$$
6. Model parameters w_j
7. What's being optimized $$E_D(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^{N} \{y_i - \mathbf{w}^{\mathrm{T}} \phi(x_i)\}^2$$

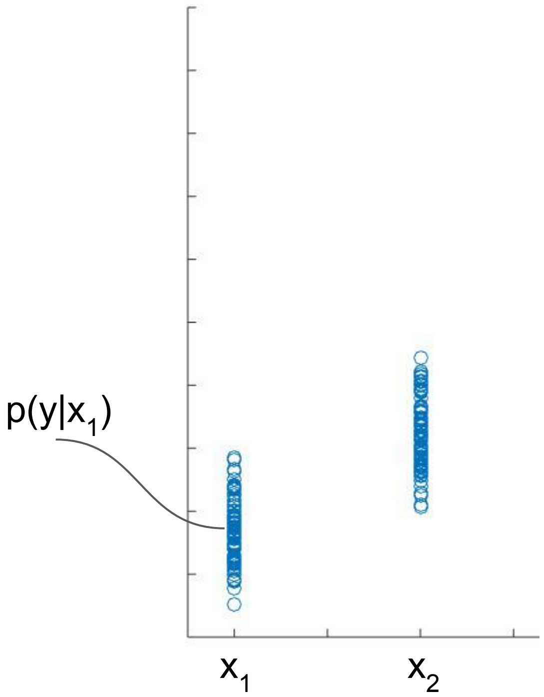
General framework

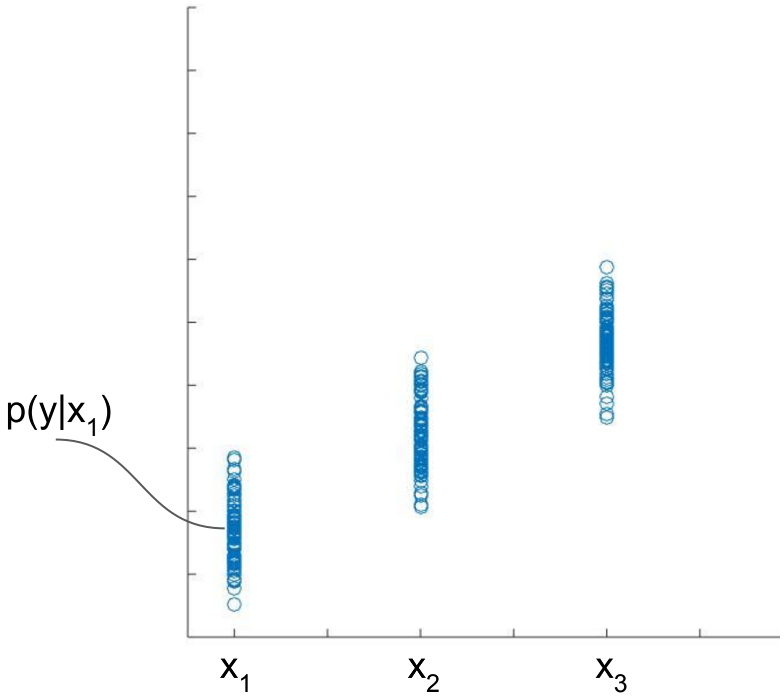
Supervised setting:

1. Data $x_i \in \mathbb{R}^d$ i = 1,2,3,4…..N
2. Data transformation $\phi_j(\mathbf{x})$, here we will use $\phi(\mathbf{x}) = \mathbf{x}$ for simplicity.
3. Targets/Labels $y_i \in \mathbb{R}$ i = 1,2,3,4…..N
4. Goal is to select $f : \mathbb{R}^d \rightarrow \mathbb{R}$ such that we can predict y from new x
5. Model class $f(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \phi(\mathbf{x})$
6. Model parameters w_j
7. What's being optimized $E_D(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^{N} \{y_i - \mathbf{w}^{\mathrm{T}} \phi(x_i)\}^2$
8. Inference method ex. Maximum Likelihood (MLE)

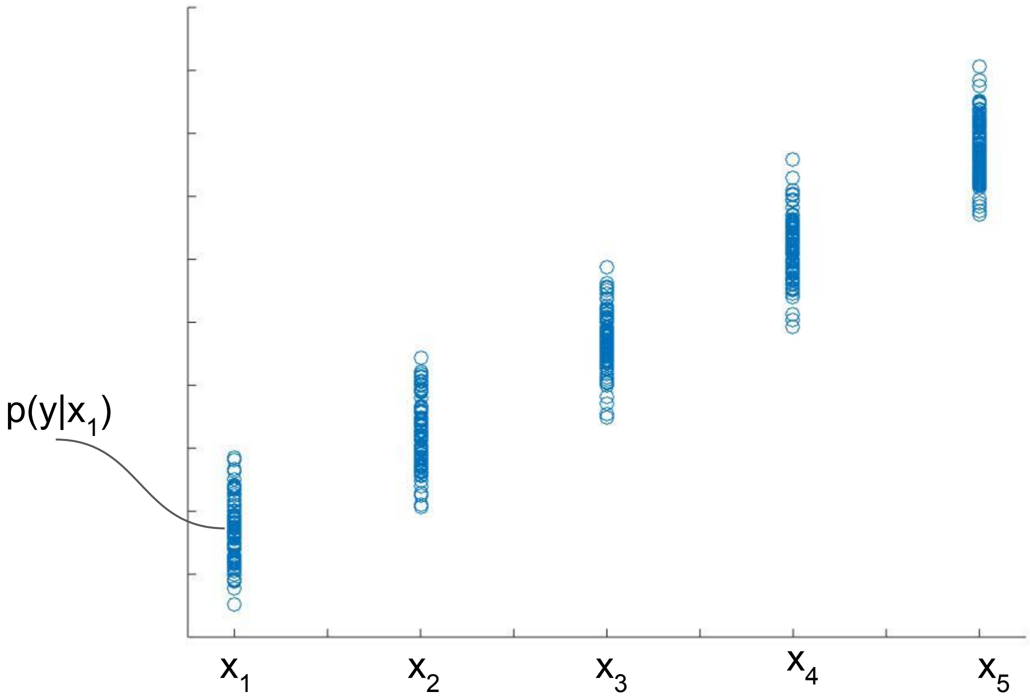


$p(y|x_1)$
x_1

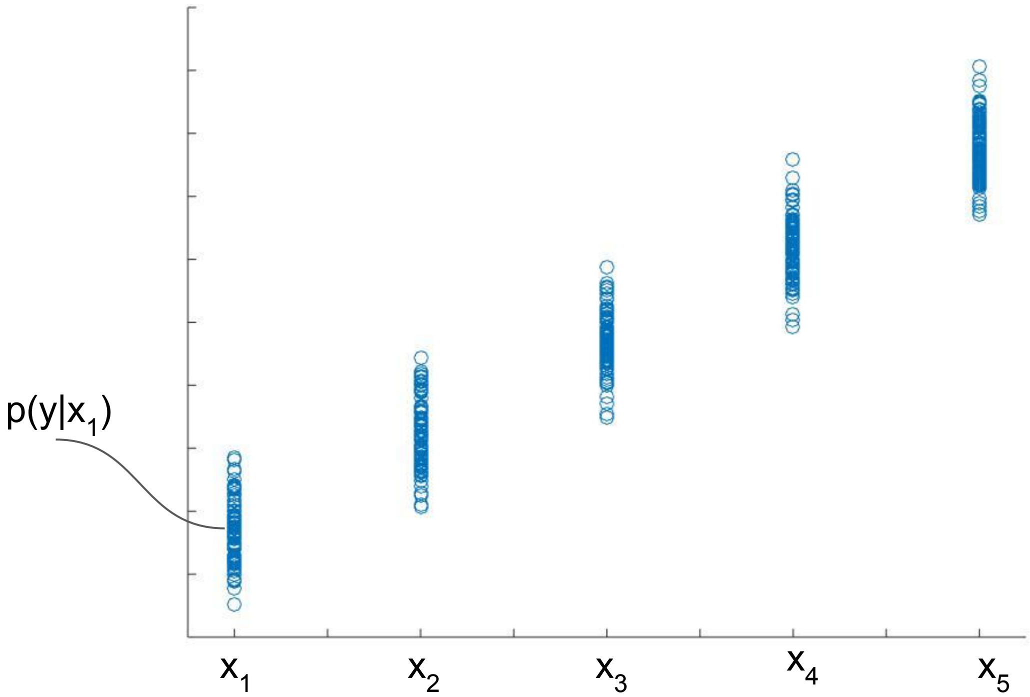

p(y|x1)
x1
x2



p(y|x1)
x1
x2
x3

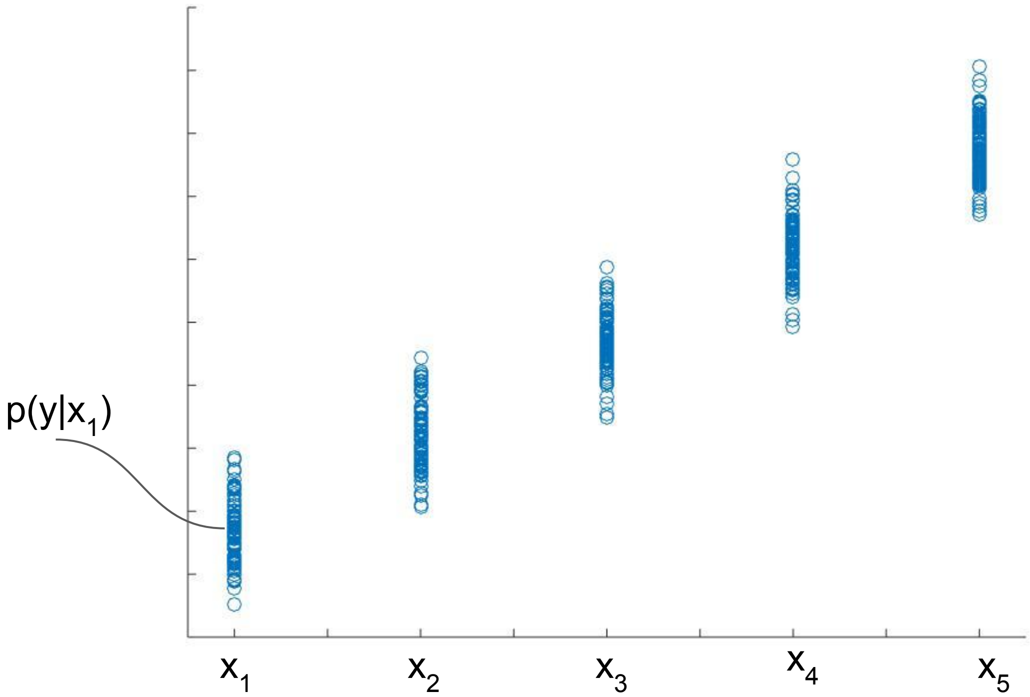


p(y|x1)
x1
x2
x3
x4
x5



<u>Gaussian linear regression:</u>

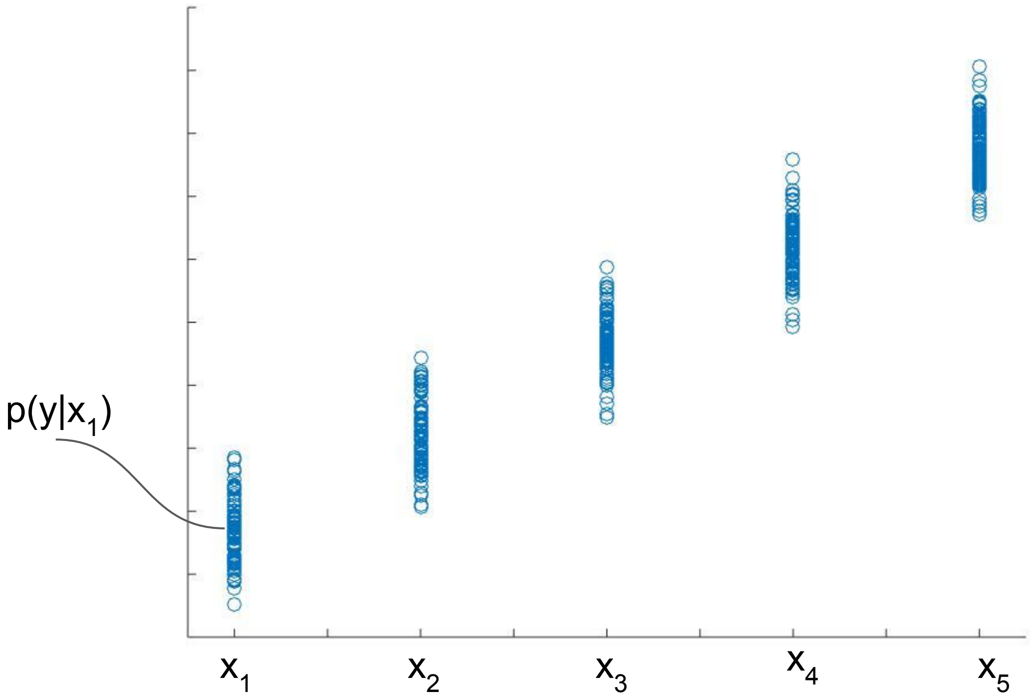
$$p_\theta(y|x) = N(y|\mu(x), \sigma^2(x)),$$



Gaussian linear regression:

$$p_\theta(y|x) = N(y|\mu(x), \sigma^2(x)),$$

$\theta = (w, \sigma^2)$ with $w \in \mathbb{R}^d$ $\sigma^2 > 0$



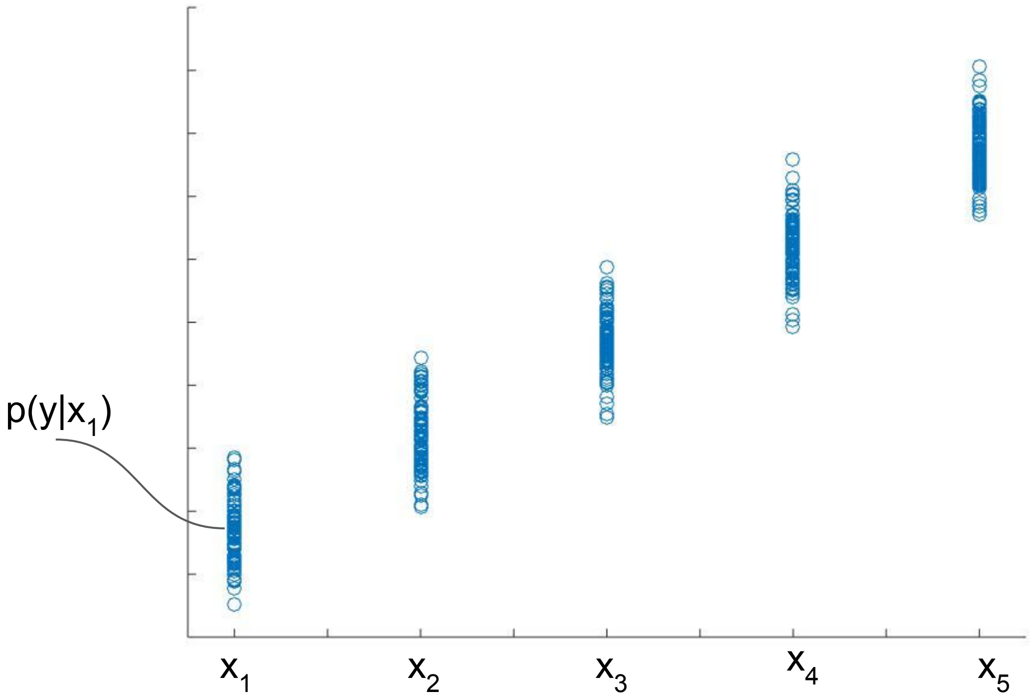
Gaussian linear regression:

$$p_\theta(y|x) = N(y|\mu(x), \sigma^2(x)),$$

$$\theta = (w, \sigma^2) \text{ with } w \in \mathbb{R}^d \; \sigma^2 > 0$$

$$\mu(x) = w^T x, \sigma^2(x) = \sigma^2$$

Gaussian linear regression:



$$p_\theta(y|x) = N(y|\mu(x), \sigma^2(x)),$$

$\theta = (w, \sigma^2)$ with $w \in \mathbb{R}^d$ $\sigma^2 > 0$

$$\mu(x) = w^T x, \sigma^2(x) = \sigma^2$$

$$p_\theta(y|x) = N(y|w^T x, \sigma^2)$$

MAXIMUM LIKELIHOOD ESTIMATION FOR GAUSSIAN LINEAR REGRESSION

(1) DATA

$$D = \left((x_1, y_1), (x_2, y_2) \dots (x_n, y_n) \right)$$

$$x_i \in \mathbb{R}^d, \quad y_i \in \mathbb{R}$$

(2) MODEL

$$Y \sim N(w^T x, \sigma^2)$$

ASSUME σ^2 KNOWN.

$\boxed{\text{(3) OBJECTIVE (LIKELIHOOD)}}$

$$\theta \in \Theta, \quad \theta_{MLE} \in \underset{\theta \in \Theta}{\text{ARGMAX}}\ P(D \mid \theta)$$

$$P(D \mid \theta) = P(y_1, y_2 \cdots y_n \mid x_1, x_2, x_3 \dots x_n, \theta)$$

$$= \prod_{i=1}^{n} p(y_i \mid x_i, \theta)$$

$$= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}\left(y_i - w^T x_i\right)^2\right)$$

$$= \left(\frac{1}{2\pi\sigma^2}\right)^n \exp\left(\frac{-1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i - w^T x_i\right)^2\right)$$

$$\begin{pmatrix} y_1 - w^T x_1 \\ \vdots \\ y_n - w^T x_n \end{pmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} - \begin{pmatrix} x_1^T w \\ \vdots \\ x_n^T w \end{pmatrix} = \vec{y} - \begin{pmatrix} x_1^T \\ \vdots \\ x_n^T \end{pmatrix} \vec{w} = \vec{y} - A\vec{w}$$

$$\boxed{w^T x_i = x_i^T w}$$

$$A = \begin{pmatrix} \text{---} & x_1^T & \text{---} \\ & \vdots & \\ \text{---} & x_n^T & \text{---} \end{pmatrix}$$

"DESIGN MATRIX"

(2)

$$\sum_{i=1}^{n} (y_i - w^T x_i)^2 = (y - Aw)^T (y - Aw)$$

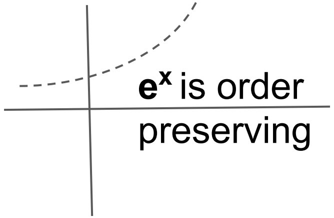
$$= \|y - Aw\|^2$$

EUCLIDIAN NORM.

$$p(D|\theta) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) \exp\left(-\frac{1}{2\sigma^2} (y - Aw)^T (y - Aw)\right)$$

→ MAXIMIZING $p(D|\theta)$ = MINIMIZE $(y - Aw)^T (y - Aw)$

why is this true?



$\boxed{\text{(4) OPTIMIZATION}}$

$$\mathcal{L} = (y - Aw)^T (y - Aw) = \overbrace{y^T y}^{\text{NO DEPENDENCE ON } w} - \underbrace{2y^T A w}_{= 2w^T A^T y} + \overbrace{w^T A^T A w}^{\text{QUADRATIC FORM.}}$$

$$\nabla_w \mathcal{L} = 0 - 2A^T y + 2A^T A w$$

$$A^T A w = A^T y$$

$$\underline{w^* = (A^T A)^{-1} A^T y.}$$

$$A^+ = (A^T A)^{-1} A^T$$

"MOORE PENROSE INVERSE".

→ w^* IS CRITICAL POINT, IS IT A MINIMUM?

COMPUTE HESSIAN, $H = \nabla^2 \mathcal{L}$

$$H_{i,j} = \left(\frac{\partial}{\partial x_i}\frac{\partial}{\partial x_j}\mathcal{L}\right)_{i,j}$$

$$H = \nabla(\nabla \mathcal{L}) = \nabla\left(-A^T y + A^T A w\right)$$

$$= \underline{A^T A}$$, w^* IS MINIMUM IF $A^T A$ IS POSITIVE SEMI-DEFINITE.

$$\underline{w_{MLE} = (A^T A)^{-1} A^T y.}$$

If the Hessian at a given point has all positive eigenvalues, it is a positive-semidefinite. This Suggests that the underlying function is convex.

we can also get an MLE estimate of σ_{MLE} using a similar approach

(5) EXTENSION TO CASE WHEN $\phi(x)$ IS NOT IDENTITY

$f(x) = w^T \phi(x)$ $\phi : \mathbb{R}^d \to \mathbb{R}^m$, WHERE m = # OF BASIS FUNCTIONS.

DESIGN MATRIX FORMERLY (i.e FOR $\phi(x) = x$)

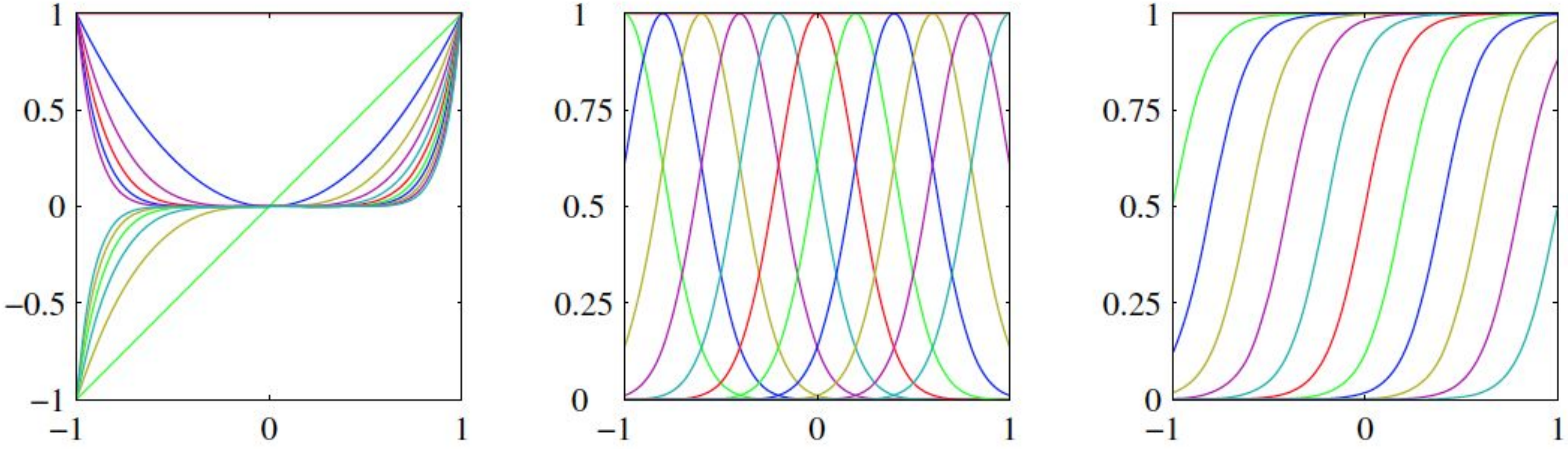
$$A = \begin{pmatrix} \text{---} & x_1^T & \text{---} \\ & \vdots & \\ \text{---} & x_n^T & \text{---} \end{pmatrix}$$

THEN FOR ARBITRARY $\phi(x)$

$$\Phi = \begin{pmatrix} \text{---} & \phi(x_1)^T & \text{---} \\ & \vdots & \\ \text{---} & \phi(x_n)^T & \text{---} \end{pmatrix}$$

$$w_{MLE} = (\Phi^T \Phi)^{-1} \Phi y = \Phi^+ y$$

Nonlinearity via basis functions $\phi(\mathbf{x})$

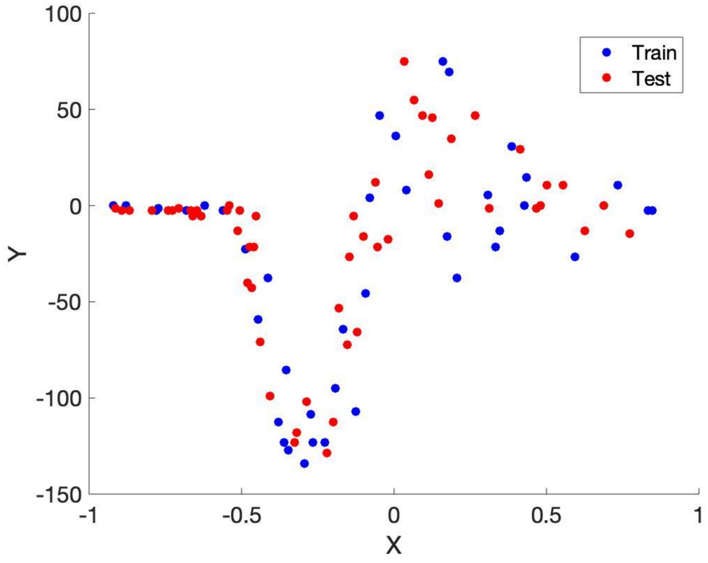


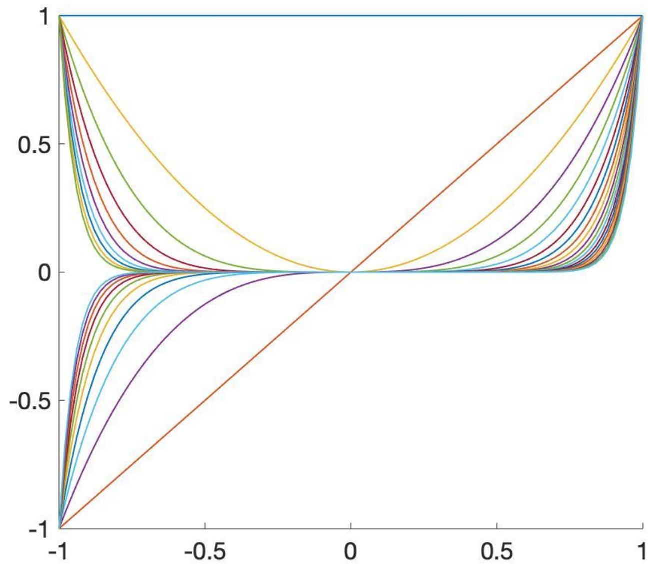
Polynomials

Gaussians

Sigmoids

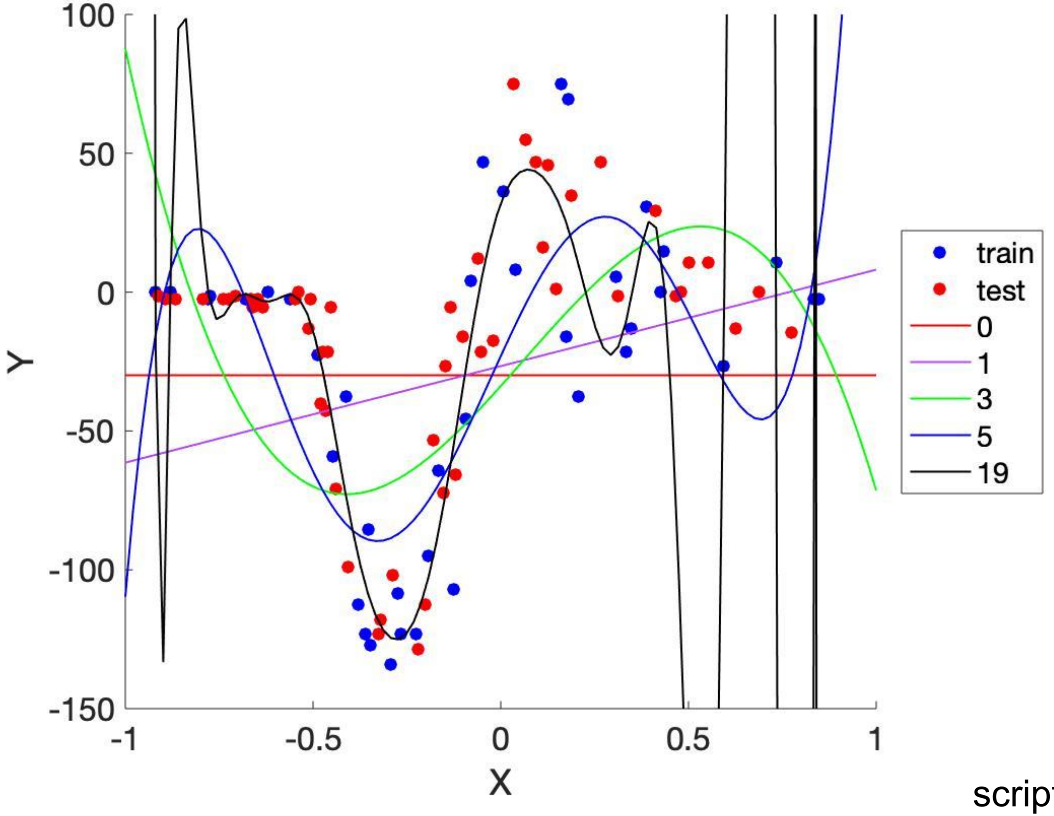
Nonlinearity via basis functions $\phi(\mathbf{x})$





scripts will be uploaded
on class website

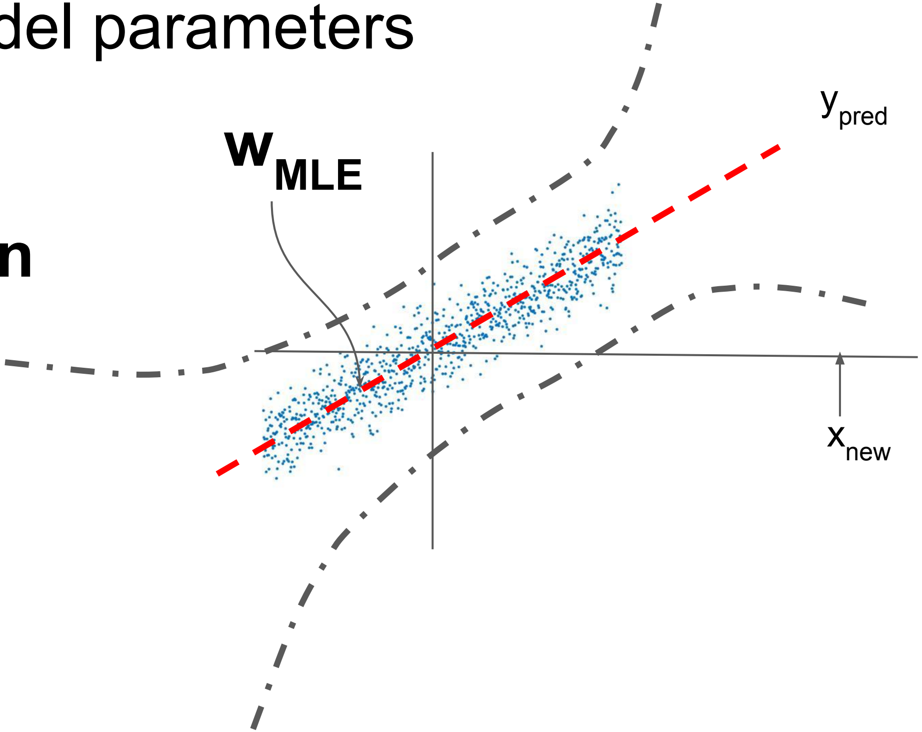
Nonlinearity via basis functions $\phi(\mathbf{x})$



scripts will be uploaded
on class website

So what's the problem with $\mathbf{w}_{\mathbf{MLE}}$?

1. Easily leads to overfitting
2. No measure of uncertainty
3. Add conjugate prior on model parameters and compute posterior
    a. Bayesian approach
       a type of regularization



COMPUTING THE POSTERIOR FOR GAUSSIAN LINEAR REGRESSION

RECALL THAT THE LIKELIHOOD:

$$P(D \mid w) \propto \exp\left(-\frac{a}{2}(y - Aw)^T (y - Aw)\right)$$

where $a = \frac{1}{\sigma^2}$

$$A = \begin{pmatrix} \text{---} & x_1^T & \text{---} \\ & \vdots & \\ \text{---} & x_n^T & \text{---} \end{pmatrix}$$

"DESIGN MATRIX"

(1) POSTERIOR

$$P(w|D) \propto P(D|w)\,P(w)$$

$$w \sim \mathcal{N}(0, b^{-1} I)$$

MULTIVARIATE GAUSSIAN.

$$P(w|D) \propto \exp\left(-\frac{a}{2}(y-Aw)^T(y-Aw)\right) \cdot \exp\left(-\frac{b}{2} w^T w\right)$$

$$= \exp\left(-\frac{a}{2}(y-Aw)^T(y-Aw) - \boxed{\frac{b}{2} w^T w}\right)$$

RIDGE REGRESSION TYPE REGULARIZATION.

NOTICE THAT:

1. $P(w|D)$ IS QUADRATIC IN "w"
2. ALSO A GAUSSIAN.

$$P(w|D) = \mathcal{N}(w \mid \mu, \Lambda^{-1})$$

$$\begin{cases} \Lambda = a A^T A + bI \\ \mu = a \Lambda^{-1} A^T y \end{cases}$$

Multivariate gaussian

$$f_{\mathbf{X}}(x_1, \ldots, x_k) = \frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)}{\sqrt{(2\pi)^k|\boldsymbol{\Sigma}|}}$$

$\boldsymbol{\mu} = 0$
$\boldsymbol{\Sigma}$ = diagonal with entries 1/b

A direct consequence of having the posterior is that we can easily get the $\mathbf{w}_{MAP}$ estimate

$$\mathbf{w}_{MAP} = (\mathbf{A}^T\mathbf{A} + \frac{b}{a}\mathbf{I})^{-1}\mathbf{A}^T\mathbf{y}$$

compare with

$$\mathbf{w}_{MLE} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{y}$$

2. PREDICTIVE DISTRIBUTION

(5)

WHAT WE REALLY WANT...

GIVEN x_{new}, $P(y \mid x, D)$

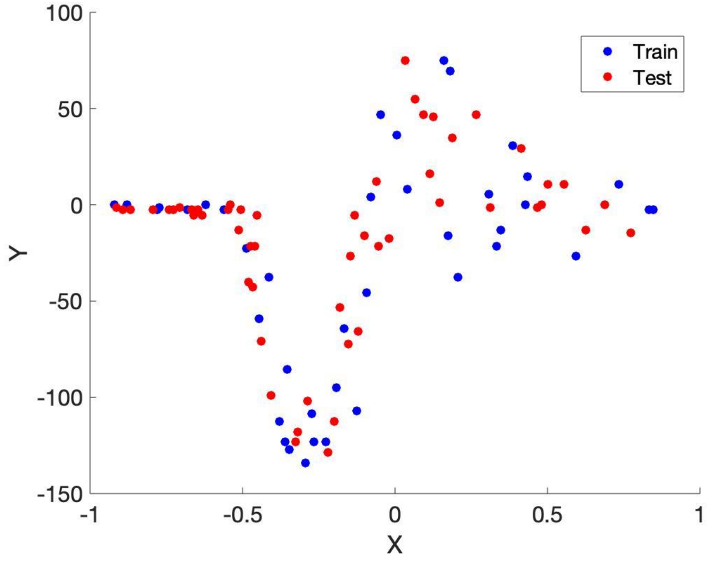
$$P(y \mid x, D) = N(y \mid u, 1/\lambda)$$

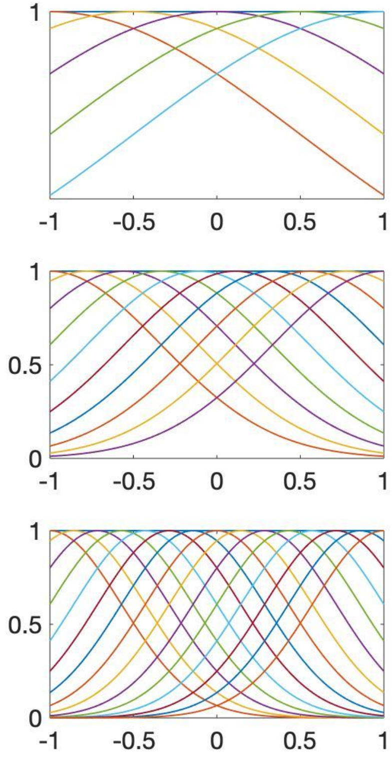
WHERE:

$$\boxed{\begin{aligned} u &= \mu^T x \\ \frac{1}{\lambda} &= \frac{1}{a} + x^T \Lambda^{-1} x \end{aligned}}$$

A thorough and complete derivation, can be found here: https://www.youtube.com/watch?v=LCISTY9S6SQ&list=PLED7YdXrsctWQf1K9BIUJQrsmGNX0PdQH&index=64

Nonlinearity via basis functions $\phi(\mathbf{x})$

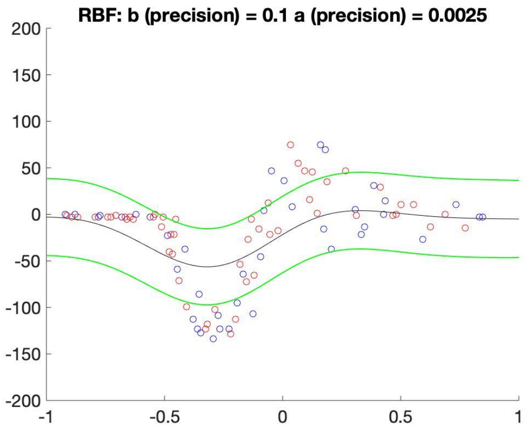


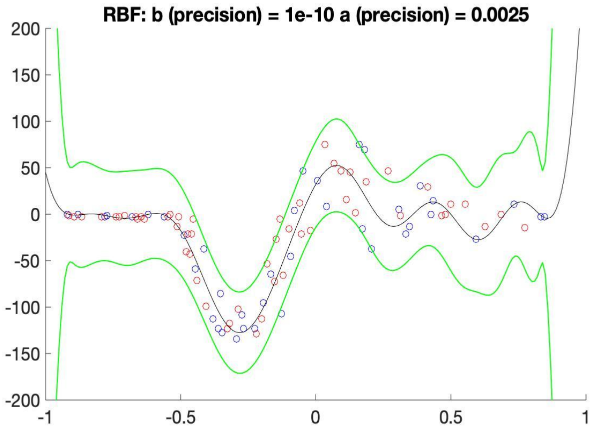


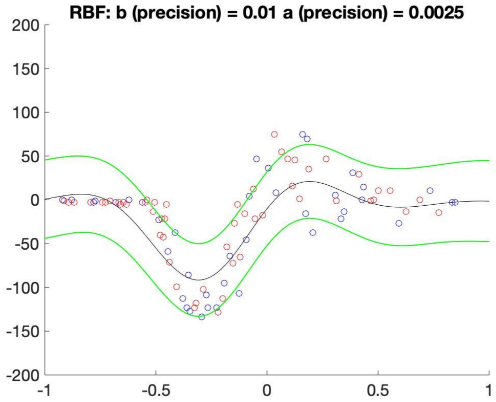
M = 30 basis functions

scripts will be uploaded
on class website

Varying the prior on the model parameters



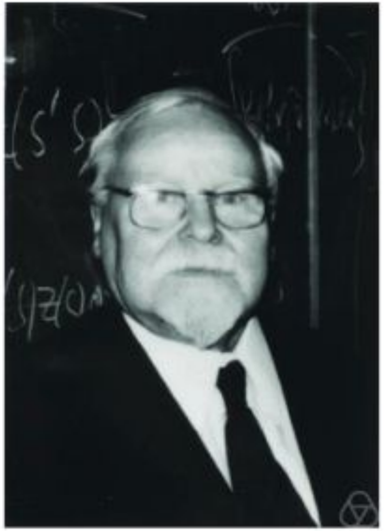




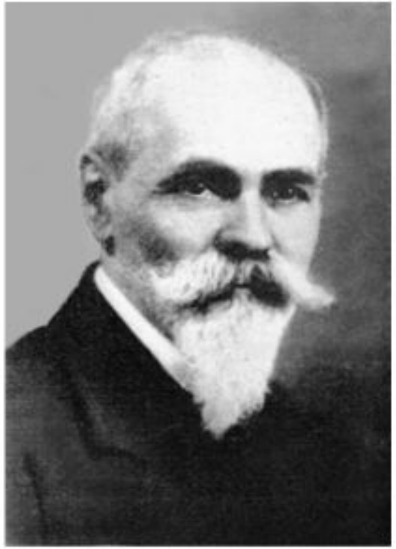
For very small precision (i.e. prior is infinitely broad), we converge on $\mathbf{w}_{MLE}$

The precision parameter is a hyperparameter that can be learned from the data

Different interpretations of the effect of regularization



Tikhonov, smoothing an ill-posed problem



Zaremba, model complexity minimization

Bayes: priors over parameters

Different kinds of regularization

- L0 Norm
  - # of non-zero entries

$$\|w\|_0 = \sum_d 1_{[w_d \neq 0]}$$

- L1 Norm
  - Sum of absolute values

$$|w| = \|w\|_1 = \sum_d |w_d|$$

- L2 Norm & Squared L2 Norm
  - Sum of squares
  - Sqrt(sum of squares)

$$\|w\| = \sqrt{\sum_d w_d^2} \equiv \sqrt{w^T w}$$

$$\|w\|^2 = \sum_d w_d^2 \equiv w^T w$$

- L-infinity Norm
  - Max absolute value

$$\|w\|_\infty = \lim_{p \to \infty} \sqrt[p]{\sum_d |w_d|^p} = \max_d |w_d|$$

Different kinds of regularization

- L0 Norm
  - # of non-zero entries

$$\|w\|_0 = \sum_d 1_{[w_d \neq 0]}$$

- L1 Norm
  - Sum of absolute values

$$|w| = \|w\|_1 = \sum_d |w_d|$$

- L2 Norm & Squared L2 Norm
  - Sum of squares
  - Sqrt(sum of squares)

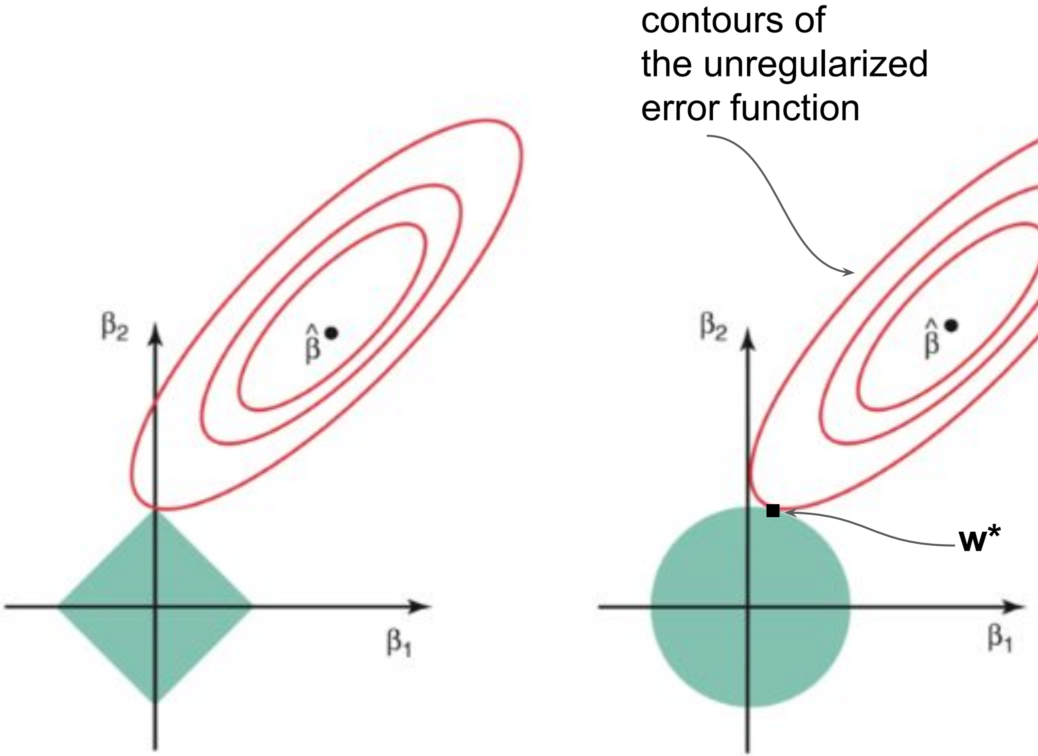
$$\|w\| = \sqrt{\sum_d w_d^2} \equiv \sqrt{w^T w}$$

$$\|w\|^2 = \sum_d w_d^2 \equiv w^T w$$

- L-infinity Norm
  - Max absolute value

$$\|w\|_\infty = \lim_{p \to \infty} \sqrt[p]{\sum_d |w_d|^p} = \max_d |w_d|$$

A geometrical interpretation of regularization (L_1, L_2)



Ridge constraint:

$$\beta_1^2 + \beta_2^2 = 1$$

Lasso constraint:

$$|\beta_1| + |\beta_2| = 1$$

Model selection

- **"True" distribution:** P(x,y)
  - Unknown to us
- **Train:** f(x) = y
  - Using training data: $S = \{(x_i, y_i)\}_{i=1}^{N}$
  - Sampled identically and independently from P(x,y)
- **Test Error:**

$$L_P(f) = E_{(x,y)\sim P(x,y)}\left[L(y, f(x))\right]$$

- **Overfitting:** Test Error >> Training Error

Model selection

- **Test Error:**

$$L_P(f) = E_{(x,y)\sim P(x,y)}\left[L(y, f(x))\right]$$

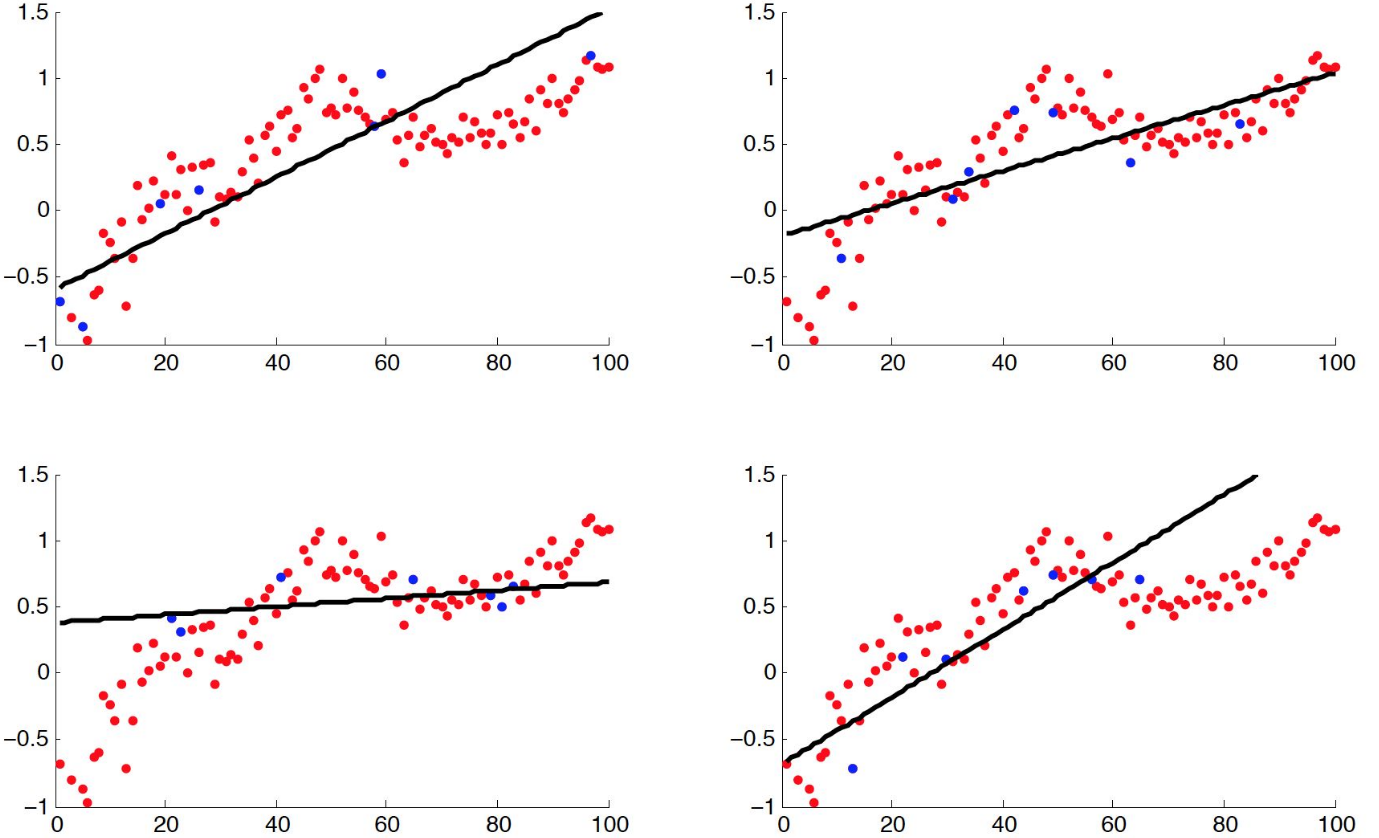
- **Treat f_S as random variable:** (randomness over *S*)

$$f_S = \underset{w,b}{\operatorname{argmin}} \sum_{(x_i,y_i)\in S} L\left(y_i, f(x_i \mid w, b)\right)$$

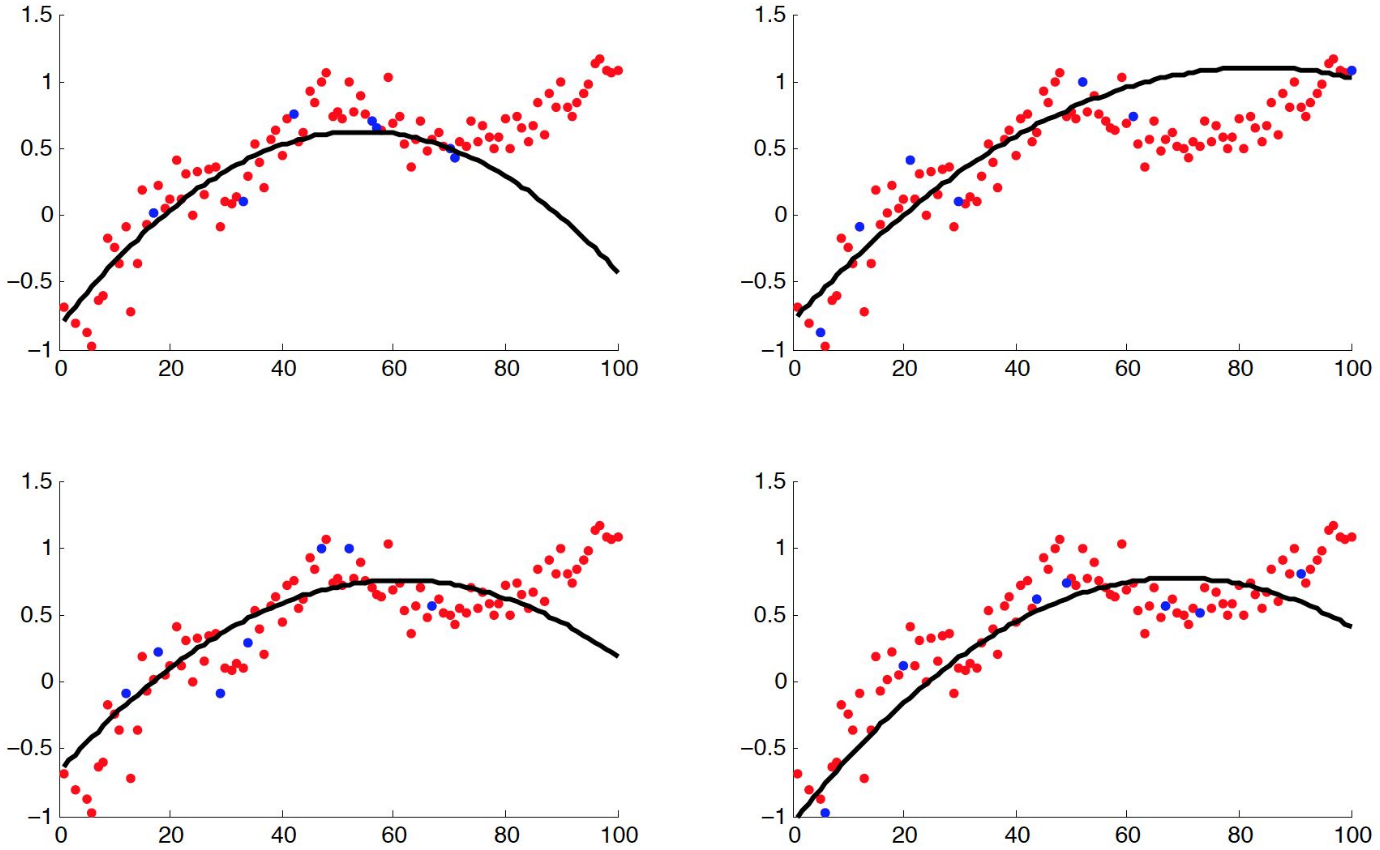
- **Expected Test Error:**

$$E_S\left[L_P(f_S)\right] = E_S\left[E_{(x,y)\sim P(x,y)}\left[L(y, f_S(x))\right]\right]$$

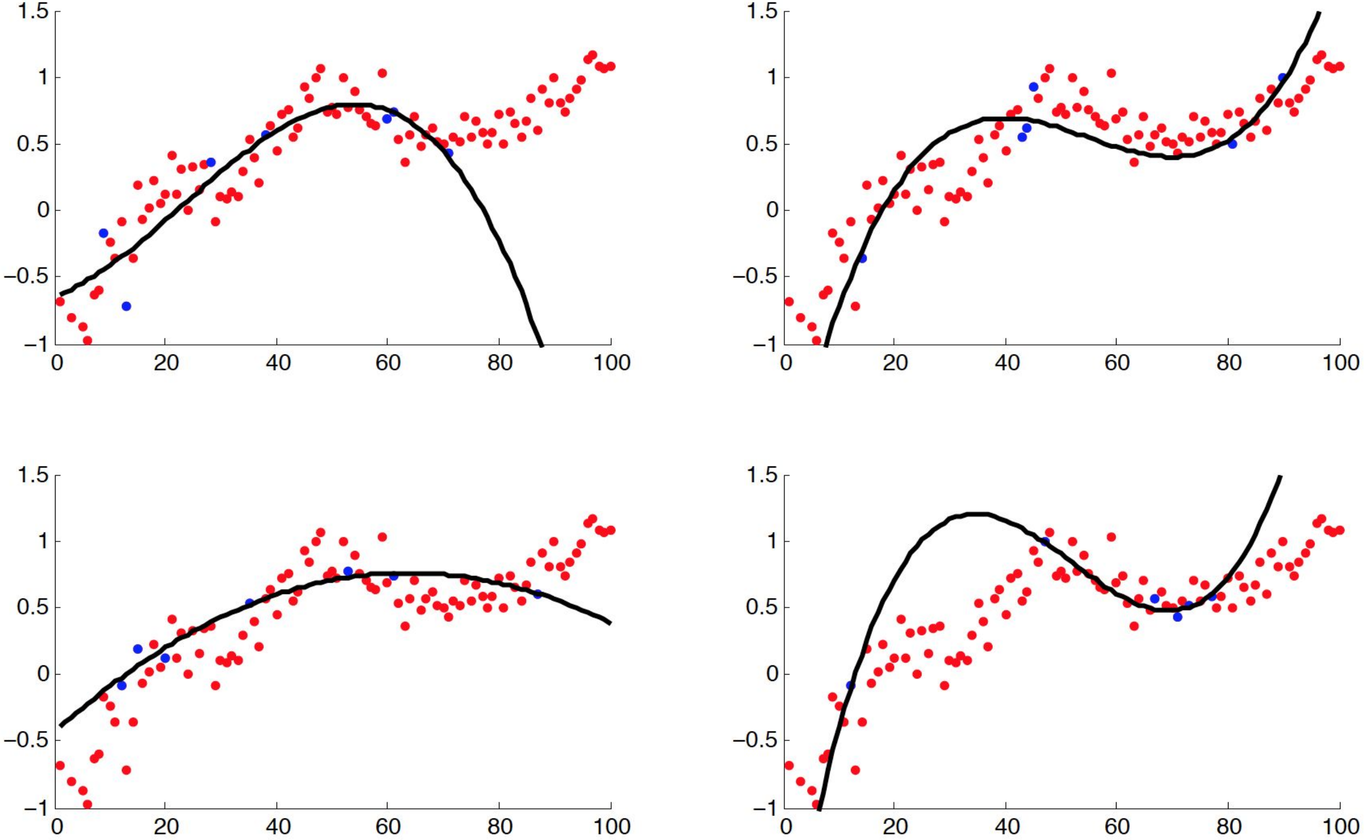
$f_S(x)$ Linear



$f_S(x)$ Quadratic



$f_S(x)$ Cubic



Bias-variance tradeoff (for squared loss)

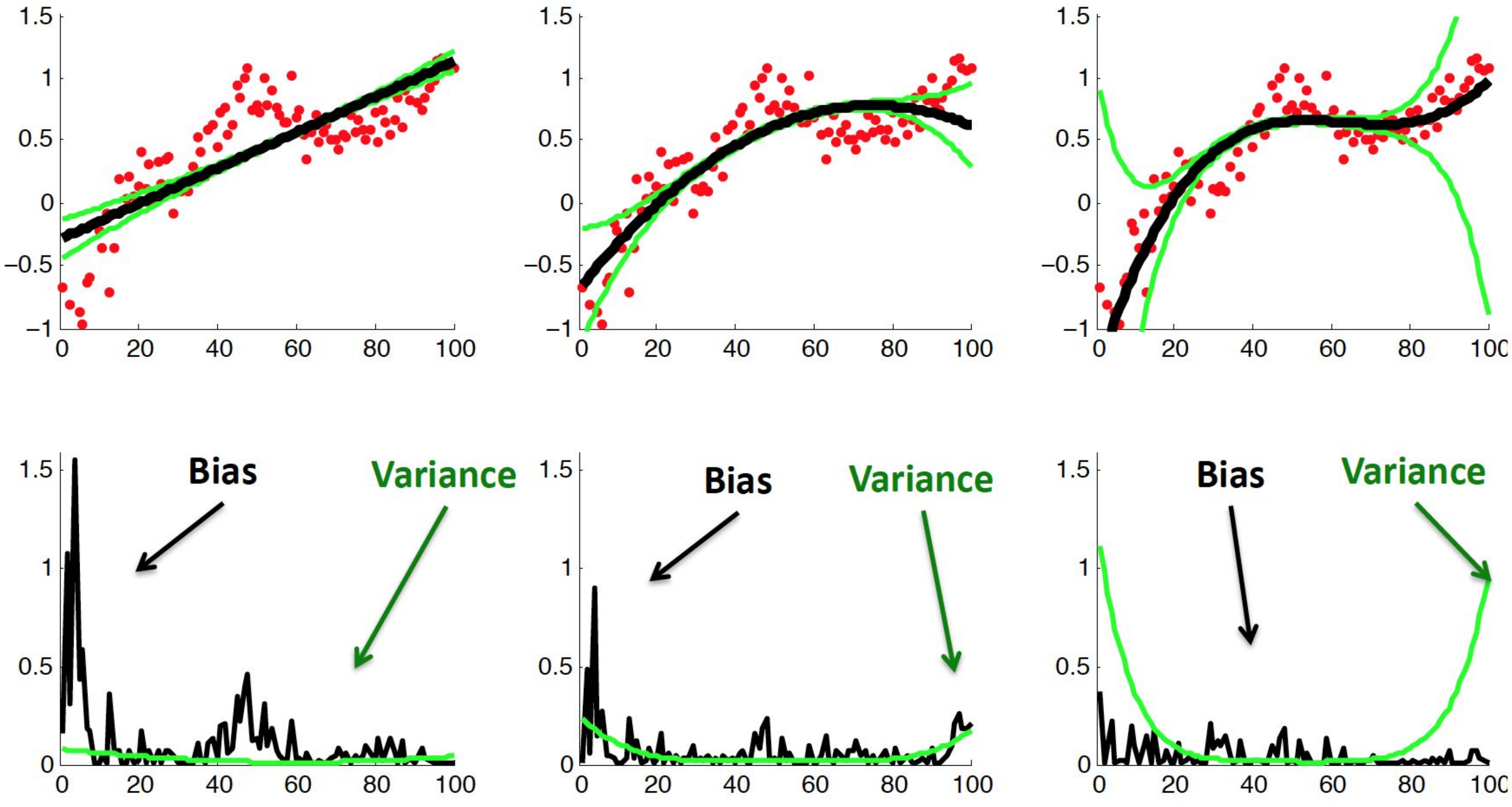
$$E_S\left[L_P(f_S)\right] = E_S\left[E_{(x,y)\sim P(x,y)}\left[L(y, f_S(x))\right]\right]$$

- For squared error:

$$E_S\left[L_P(f_S)\right] = E_{(x,y)\sim P(x,y)}\left[\underbrace{E_S\left[\left(f_S(x) - F(x)\right)^2\right]}_{\text{Variance Term}} + \underbrace{\left(F(x) - y\right)^2}_{\text{Bias Term}}\right]$$

$$F(x) = E_S\left[f_S(x)\right]$$

"Average prediction"



Bias
Variance
Bias
Variance
Bias
Variance

The end