Lecture 8 Linear Model for Regression

Lecture 8 Linear Model for Regression

- Regression and linear models
- Batch methods
  - Ordinary least squares (OLS)
  - Maximum likelihood estimates
- Sequential methods
  - Least mean squares (LMS)
  - Recursive (sequential) least squares (RLS)

What is regression analysis?

Problem Setup

Given a set of *N* labeled examples, $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$ $(\mathbf{x}_n \in \mathcal{X} \subset \mathbb{R}^D$ and $y_n \in \mathcal{Y} \subset \mathbb{R})$, the goal is to learn a mapping

$$f(\mathbf{x}) : \mathcal{X} \mapsto \mathcal{Y},$$

which associates x with *y*, such that we can make prediction about y_* when a new input $\mathbf{x}_* \notin \mathcal{D}$ is provided.

- **Parametric regression:** Assume a functional form for f(x) (e.g. linear models).
- **Nonparametric regression:** Do not assume functional form for f(x).

In this lecture we focus on parametric regression.

Regression

- Regression aims at modeling the dependence of a response Y on a covariate *X*. In other words, the goal of regression is to predict the value of one or more continuous target variables *y* given the value of input vector x.
- The regression model is described by

$$y = f(\mathbf{x}) + \epsilon.$$

- Terminology:
  - x: input, independent variable, predictor, regressor, covariate
  - y: output, dependent variable, response
- The dependence of a response on a covariate is captured via a conditional probability distribution, $p(y|\mathbf{x})$
- Depending on *f(x)*,
  - Linear regression with **basis functions:** $f(\mathbf{x}) = \sum_{j=1}^{M} w_j \phi_j(\mathbf{x}) + w_0.$
  - Linear regression with **kernels:** $f(\mathbf{x}) = \sum_{i=1}^{N} w_i k(\mathbf{x}, \mathbf{x}_i) + w_0.$

Regression Function: Conditional Mean

We consider the mean squared error and find the MMSE estimate:

$$\begin{aligned}
\mathcal{E}(f) &= \mathbb{E}\,(y - f(\mathbf{x}))^2 \\
&= \int\int\cdots\int (y - f(\mathbf{x}))^2 p(\mathbf{x}, y)\, d\mathbf{x}dy \\
&= \int\int\cdots\int (y - f(\mathbf{x}))^2 p(\mathbf{x})p(y|\mathbf{x})\, d\mathbf{x}dy \\
&= \int\cdots\int p(\mathbf{x}) \left[\underbrace{\int (y - f(\mathbf{x}))^2 p(y|\mathbf{x})\, dy}_{\text{to be minimized}} \right] d\mathbf{x}
\end{aligned}$$

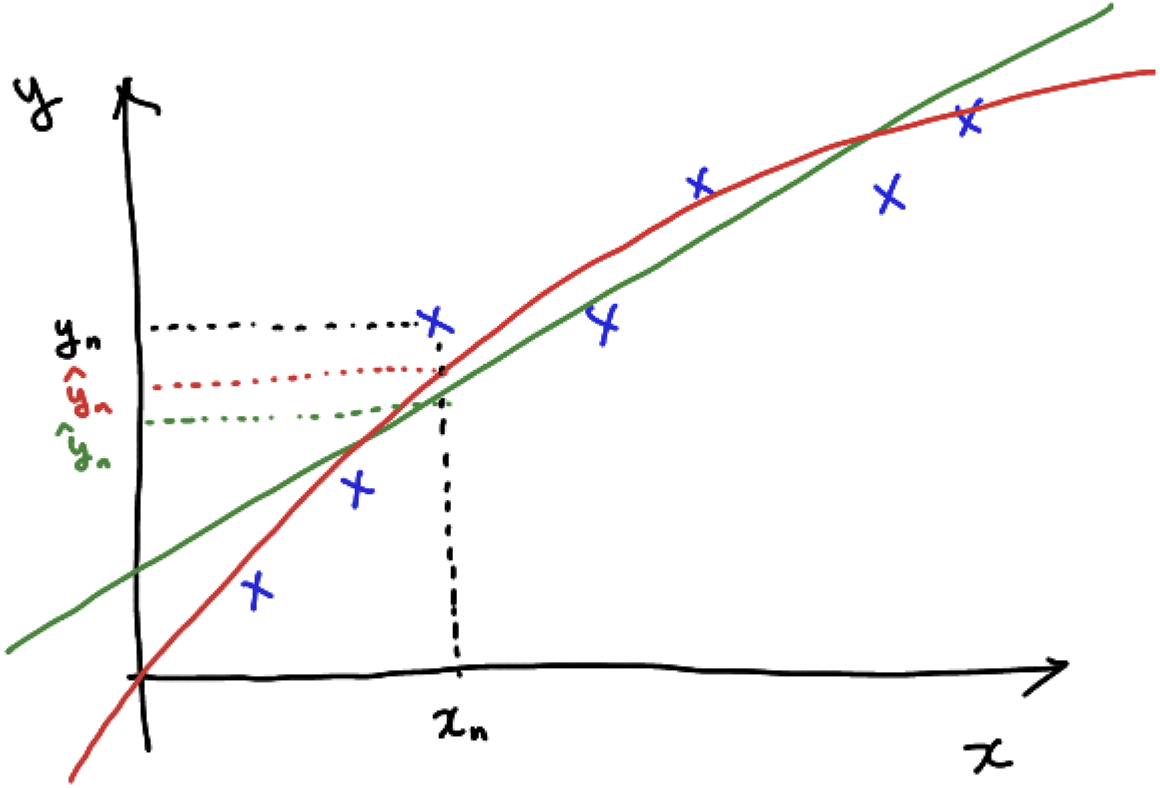
$\frac{\partial}{\partial f(\mathbf{x})}\left[\int (y - f(\mathbf{x}))^2 p(y|\mathbf{x})\, dy\right] = 0$ leads to

$$\boxed{f(\mathbf{x}) = \int y p(y|\mathbf{x})\, dy = \mathbb{E}\left[y|\mathbf{x}\right]}.$$

Linear Regression

Why Linear Models?

$$\boxed{y_n = w_1 x_{n,1} + w_2 x_{n,2} + \cdots + w_M x_{n,M} + w_0 + \epsilon_n,} \quad \forall n = 1, \ldots, N.$$



- ► Built on well-developed **linear transformation.**
- ► Can be solved **analytically.**
- ► Yield some **interpretability** (in contrast to deep learning).

Linear Regression

Linear regression refers to a model in which the conditional mean of *y* given the value of x is an affine function of $\phi(\mathbf{x})$

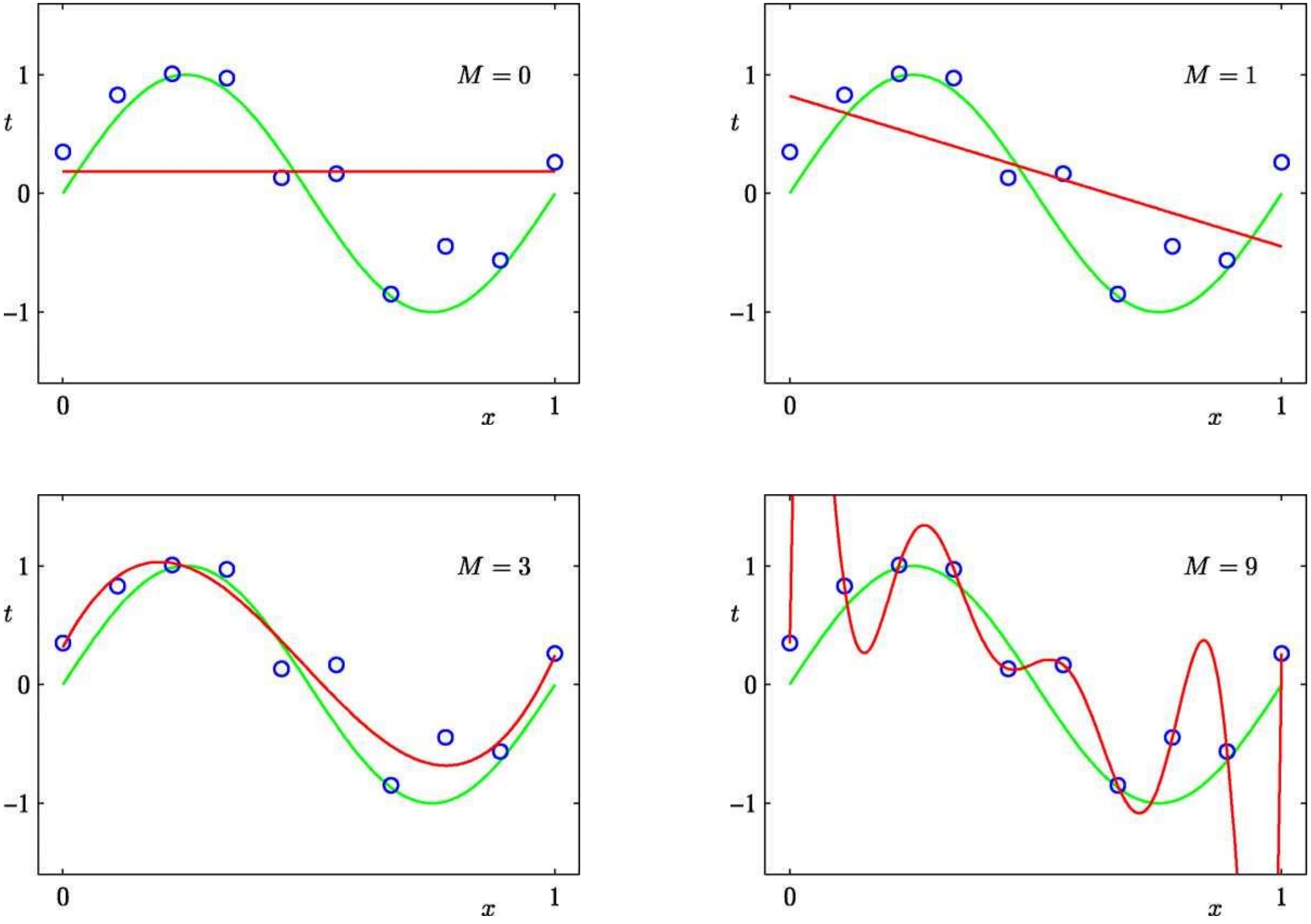
$$f(\mathbf{x}) = \sum_{j=1}^{M} w_j \phi_j(\mathbf{x}) + w_0 \phi_0(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}),$$

where $\phi_j(\mathbf{x})$ are known as basis functions and

$$\begin{aligned} \mathbf{w} &= [w_0, w_1, \ldots, w_M]^\top, \\ \boldsymbol{\phi} &= [\phi_0, \phi_1, \ldots, \phi_M]^\top. \end{aligned}$$

By using nonlinear basis functions, we allow the function f(x) to be a nonlinear function of the input vector x (but a linear function of $\boldsymbol{\phi}(\mathbf{x})$).

Polynomial Regression: $y_n = \sum_{j=0}^{M} w_j \phi_j(x_n) = \sum_{j=0}^{M} w_j x_n^j$



[Figure source: Bishop's PRML]

Basis Functions

- Polynomial regression: $\phi_j(x) = x^j$.
- Gaussian basis functions: $\phi_j(\mathbf{x}) = \exp\left\{-\frac{\|\mathbf{x}-\boldsymbol{\mu}_j\|^2}{2\sigma^2}\right\}$.
- Spline basis functions: Piecewise polynomials (divide the input space up into regions and fit a different polynomial in each region).
- Many other possible basis functions: sigmoidal basis functions, hyperbolic tangent basis functions, Fourier basis, wavelet basis, and so on.

Ordinary Least Squares

Loss function view

Least Squares Method

Given a set of training data $\{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$, we determine the weight vector $\mathbf{w} \in \mathbb{R}^{M+1}$ which minimizes

$$\mathcal{J}_{LS}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left(y_n - \mathbf{w}^\top \phi(\mathbf{x}_n)\right)^2 = \frac{1}{2}\|\mathbf{y} - \mathbf{\Phi}^\top \mathbf{w}\|_2^2,$$

where $\mathbf{y} = [y_1, \dots, y_N]^\top$ and $\mathbf{\Phi} \in \mathbb{R}^{(M+1)\times N}$ is known as the design matrix $\Phi_{tj} = \phi_j(\mathbf{x}_t)$, i.e.,

$$\mathbf{\Phi}^\top = \begin{bmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_M(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_M(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_M(\mathbf{x}_N) \end{bmatrix}.$$

Note that

$$\|\mathbf{y} - \mathbf{\Phi}^\top \mathbf{w}\|_2^2 = \left(\mathbf{y} - \mathbf{\Phi}^\top \mathbf{w}\right)^\top \left(\mathbf{y} - \mathbf{\Phi}^\top \mathbf{w}\right).$$

Find the estimate $\widehat{\mathbf{w}}_{LS}$ such that

$$\mathbf{w}_{LS} = \arg\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{y} - \mathbf{\Phi}^\top \mathbf{w}\|_2^2,$$

where both y and Φ are given.

How do you find the minimizer $\mathbf{w}_{LS}$?

Solve $\frac{\partial}{\partial \mathbf{w}} \left(\frac{1}{2}\|\mathbf{y} - \mathbf{\Phi}^\top \mathbf{w}\|_2^2 \right) = 0$ for $\mathbf{w}$.

Note that

$$\begin{aligned}\frac{1}{2}\|\mathbf{y}-\mathbf{\Phi}^\top\mathbf{w}\|_2^2 &= \frac{1}{2}\left(\mathbf{y}-\mathbf{\Phi}^\top\mathbf{w}\right)^\top\left(\mathbf{y}-\mathbf{\Phi}^\top\mathbf{w}\right)\\ &= \frac{1}{2}\left(\mathbf{y}^\top\mathbf{y}-\mathbf{w}^\top\mathbf{\Phi}\mathbf{y}-\mathbf{y}^\top\mathbf{\Phi}^\top\mathbf{w}+\mathbf{w}^\top\mathbf{\Phi}\mathbf{\Phi}^\top\mathbf{w}\right).\end{aligned}$$

Then, we have

$$\frac{\partial}{\partial\mathbf{w}}\left(\frac{1}{2}\|\mathbf{y}-\mathbf{\Phi}^\top\mathbf{w}\|_2^2\right)=\mathbf{\Phi}\mathbf{\Phi}^\top\mathbf{w}-\mathbf{\Phi}\mathbf{y}.$$

Therefore, $\frac{\partial}{\partial \mathbf{w}}\left(\frac{1}{2}\|\mathbf{y}-\mathbf{\Phi w}\|_2^2\right)=0$ leads to the **normal equation** that is of the form

$$\mathbf{\Phi\Phi}^\top \mathbf{w} = \mathbf{\Phi y}.$$

Thus, LS estimate of **w** is given by

$$\mathbf{w}_{LS} = \left(\mathbf{\Phi\Phi}^\top\right)^{-1}\mathbf{\Phi y} = \mathbf{\Phi}^\dagger \mathbf{y},$$

where $\mathbf{\Phi}^\dagger$ is known as the **Moore-Penrose pseudo-inverse.**

Least Squares

Probabilistic model view with MLE

Maximum Likelihood

We consider a linear model where the target variable y_n is assumed to be generated by a deterministic function $f(\mathbf{x}_n; \mathbf{w}) = \mathbf{w}^\top \phi(\mathbf{x}_n)$ with additive Gaussian noise:

$$y_n = \mathbf{w}^\top \phi(\mathbf{x}_n) + \epsilon_n,$$

for $n = 1, \ldots, N$ and $\epsilon_n \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$.

In a compact form, we have

$$\mathbf{y} = \mathbf{\Phi}^\top \mathbf{w} + \epsilon.$$

In other words, we model $p(\mathbf{y}|\mathbf{\Phi}, \mathbf{w})$ as

$$p(\mathbf{y}|\mathbf{\Phi}, \mathbf{w}) = \mathcal{N}(\mathbf{\Phi}^\top \mathbf{w}, \sigma^2 \mathbf{I}).$$

The log-likelihood is given by

$$\begin{aligned}\mathcal{L} &= \log p(\mathbf{y}|\mathbf{\Phi}, \mathbf{w}) = \sum_{n=1}^{N} \log p(y_n|\phi(\mathbf{x}_n), \mathbf{w}) \\ &= -\frac{N}{2}\log\sigma^2 - \frac{N}{2}\log 2\pi - \sigma^{-2}\mathcal{J}_{LS}.\end{aligned}$$

MLE is given by

$$\mathbf{w}_{ML} = \arg\max_{\mathbf{w}} \log p(\mathbf{y}|\mathbf{\Phi}, \mathbf{w}),$$

leading to

$$\mathbf{w}_{ML} = \mathbf{w}_{LS},$$

which we arrived at under **Gaussian noise assumption.**

Sequential Methods

LMS and RLS

Online Learning

A method of machine learning in which data becomes available in a sequential order and is used to update our best predictor for future data at each step, as opposed to batch learning techniques which generate the best predictor by learning on the entire training data set at once.

[Source: Wikipedia]

Mean Squared Error (MSE)

Interested in MMSE estimate:

$$\arg\min_{\mathbf{w}} \mathrm{MSE} = \mathbb{E}\left[\|y - \mathbf{w}^\top \phi(\mathbf{x})\|_2^2\right].$$

Sample average: $\mathrm{MSE} \approx \frac{1}{N} \sum_{n=1}^{N} \left(y_n - \mathbf{w}^\top \phi(\mathbf{x}_n)\right)^2.$

Instantaneous squared error: $\left(y_n - \mathbf{w}^\top \phi(\mathbf{x}_n)\right)^2.$

Least Mean Squares (LMS)

Approximate $\mathbb{E}\left[\left(y - \mathbf{w}^\top \phi(\mathbf{x})\right)^2\right] \approx \left(y_n - \mathbf{w}^\top \phi(\mathbf{x}_n)\right)^2.$

LMS is a gradient-descent method which minimizes the **instantaneous squared error**

$$\mathcal{J}_n = \left(y_n - \mathbf{w}^\top \phi(\mathbf{x}_n)\right)^2.$$

The gradient descent method leads to the updating rule for w that is of the form

$$\begin{aligned} \mathbf{w} &\leftarrow \mathbf{w} - \eta \nabla \mathcal{J}_n \\ &\leftarrow \mathbf{w} + \eta \left(y_n - \mathbf{w}^\top \phi(\mathbf{x}_n)\right) \phi(\mathbf{x}_n), \end{aligned}$$

where $\eta > 0$ *is* **learning rate.**

[Widrow and Hoff, 1960]

Recursive (Sequential) LS

We introduce the forgetting factor λ to de-emphasize old samples, leading to the following error function

$$\mathcal{J}_{RLS} = \frac{1}{2}\sum_{i=1}^{n} \lambda^{n-i}(y_i - \phi_i \mathbf{w}_n^\top)^2,$$

where $\phi_i = \phi(\mathbf{x}_i)$.

Solving $\frac{\partial \mathcal{J}_{RLS}}{\partial \mathbf{w}_n} = 0$ for $\mathbf{w}_n$ leads to

$$\left[\sum_{i=1}^{n} \lambda^{n-i} \phi_i \phi_i^\top\right] \mathbf{w}_n = \left[\sum_{i=1}^{n} \lambda^{n-i} y_i \phi_i\right].$$

We define

$$\mathbf{P}_n = \left[\sum_{i=1}^{n} \lambda^{n-i} \phi_i \phi_i^\top\right]^{-1},$$
$$\mathbf{r}_n = \left[\sum_{i=1}^{n} \lambda^{n-i} y_i \phi_i\right].$$

With these definitions, we have

$$\mathbf{w}_n = \mathbf{P}_n \mathbf{r}_n.$$

The core idea of RLS is to apply the matrix inversion lemma to develop

the sequential algorithm without matrix inversion.

$$(\mathbf{A} + \mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{C}^{-1} + \mathbf{DA}^{-1}\mathbf{B})^{-1}\mathbf{DA}^{-1}.$$

The recursion for $\mathbf{P}_n$ is given by

$$\begin{aligned}
\mathbf{P}_n &= \left[\sum_{i=1}^{n} \lambda^{n-i} \phi_i \phi_i^\top\right]^{-1} \\
&= \left[\lambda \sum_{i=1}^{n-1} \lambda^{n-1-i} \phi_i \phi_i^\top + \phi_n \phi_n^\top\right]^{-1} \\
&= \frac{1}{\lambda}\left[\mathbf{P}_{n-1} - \frac{\mathbf{P}_{n-1}\phi_n\phi_n^\top \mathbf{P}_{n-1}}{\lambda + \phi_n^\top \mathbf{P}_{n-1}\phi_n}\right]. \quad \text{(matrix inversion lemma)}
\end{aligned}$$

Thus, the updating rule for w is given by

$$
\begin{aligned}
\mathbf{w}_n &= \mathbf{P}_n \mathbf{r}_n \\
&= \frac{1}{\lambda}\left[\mathbf{P}_{n-1} - \frac{\mathbf{P}_{n-1}\phi_n\phi_n^\top \mathbf{P}_{n-1}}{\lambda + \phi_n^\top \mathbf{P}_{n-1}\phi_n}\right]\left[\lambda \mathbf{r}_{n-1} + y_n\phi_n\right] \\
&= \mathbf{w}_{n-1} + \underbrace{\frac{\mathbf{P}_{n-1}\phi_n}{\lambda + \phi_n^\top \mathbf{P}_{n-1}\phi_n}}_{\text{gain}} \underbrace{\left[y_n - \phi_n^\top \mathbf{w}_{n-1}\right]}_{\text{error}} .
\end{aligned}
$$