

Introduction to Machine Learning

Lecture 4: Linear Regression

Ali Bereyhi

`ali.bereyhi@utoronto.ca`

Department of Electrical and Computer Engineering
University of Toronto

Winter 2026

Quick Recap: *ML General Recipe*

We defined ML as

the set of data-driven approaches that help us understand the environment and its behavior, and generalize it!

Any learning task is accomplished by ML through *three major steps*

- Collect data
- Specify a model *that captures the pattern*
- Develop a learning algorithm

Quick Recap: Unsupervised vs Supervised Learning

In Unsupervised Learning, samples are unlabeled

- *Data $\rightsquigarrow$ Collection of samples $\mathbb{D} = \{x_n : n = 1, \dots, N\}$*
- *Model $\rightsquigarrow$ Captures a pattern observed in data, e.g., fitting into clusters*
- *Learning algorithm $\rightsquigarrow$ It takes $\mathbb{D}$ and returns a **good** model*

*In Supervised Learning, samples are **labeled***

- *Data $\rightsquigarrow$ Collection of samples $\mathbb{D} = \{(x_n, \mathbf{v}_n) : n = 1, \dots, N\}$*
- *Model $\rightsquigarrow$ Captures relation between data samples and their **labels***
- *Learning algorithm $\rightsquigarrow$ It takes $\mathbb{D}$ and returns a **good** model*

Quick Recap: *Unsupervised Learning*

We studied three major unsupervised learning tasks

- *Clustering*
 - ↳ *Data*
 - ↳ *Model: K -centroid*
 - ↳ *Learning algorithm: K -means clustering*
- *Distribution Learning*
 - ↳ *Data*
 - ↳ *Model: a distribution with unknowns*
 - ↳ *Learning algorithm: maximum likelihood*
- *Dimensionality Reduction*
 - ↳ *Data*
 - ↳ *Model: projection*
 - ↳ *Learning algorithm: PCA*

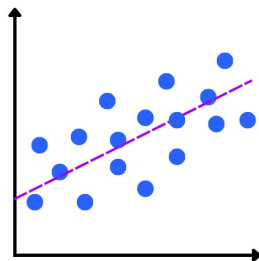
Today's Agenda: *Supervised Learning via Linear Models*

In today's lecture, we start with **supervised learning** and look into
linear models

through the following steps

- *Formulating supervised learning*
- *Linear Regression*
 - ↳ *We review some notions in functional analysis*
- *Maximum Likelihood Learning*

Supervised Learning: *Regression*



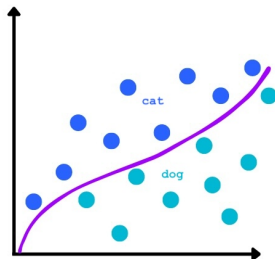
We see

$$\mathbb{D} = \{(\mathbf{x}_n, \mathbf{v}_n) : n = 1, \dots, N\}$$

and look for

$$\mathbf{v}_n = f(\mathbf{x}_n)$$

Supervised Learning: *Classification*



We see

$$\mathbb{D} = \{(\mathbf{x}_n, \text{class}_n) : n = 1, \dots, N\}$$

and look for

$$\text{class}_n = f(\mathbf{x}_n)$$

Supervised Learning: *Generic Formulation*

We again have three components

- *labeled dataset*

$$\mathbb{D} = \{(x_n \in \mathbb{X}, v_n \in \mathbb{V}) : n = 1, \dots, N\}$$

- *Model that relates data samples and their labels*

$$f : \mathbb{X} \mapsto \mathbb{V}$$

↳ We assume $f \in \mathbb{H}$ with $\mathbb{H}$ being the *hypothesis*

↳ **Example:** $\mathbb{H}$ contains all linear function, i.e.,

$$\mathbb{H} = \{f(x) = wx \text{ for all } w \in \mathbb{R}\}$$

- *Learning algorithm finds optimal model within hypothesis set*

$$\mathcal{A} : \mathbb{D} \mapsto f^* \in \mathbb{H}$$

Supervised Learning: *Generic Formulation*

In regression **labels** are continuous predictions

Example

(x_n, v_n) represent weight and **height** of people: $\mathbb{V} = [54, 272] \subset \mathbb{R}$

In classification **labels** are distinct classes

$$\mathbb{V} = \{\text{class}_1, \dots, \text{class}_K\}$$

for some integer number of classes K

Example

(x_n, v_n) represent weight and **type** of pets: $\mathbb{V} = \{\text{cat}, \text{dog}\}$

Supervised Learning: *Learning Algorithm*

? *How can we find the optimal model?*

Let for a sample $(\mathbf{x}, \mathbf{v})$

$$\mathbf{y} = f(\mathbf{x})$$

Then, the loss function determines how $\mathbf{y}$ and $\mathbf{v}$ are different

Loss Function

Loss function computes the difference between model output and true label

$$\mathcal{L}(\mathbf{y}, \mathbf{v}) \in \mathbb{R}$$

Example: *Euclidean distance between $\mathbf{y}$ and $\mathbf{v}$, i.e.,*

$$\mathcal{L}(\mathbf{y}, \mathbf{v}) = \|\mathbf{y} - \mathbf{v}\|^2$$

Supervised Learning: Risk Minimization

? *How can we find the optimal model?*

! *We search for minimal expected loss*

Risk $\equiv$ Expected Loss

Risk of a model is defined as the expectation of loss w.r.t. data distribution

$$\mathbb{E}_{\mathbf{x}, \mathbf{v}} \{ \mathcal{L}(\mathbf{y}, \mathbf{v}) \} = \mathbb{E}_{\mathbf{x}, \mathbf{v}} \{ \mathcal{L}(f(\mathbf{x}), \mathbf{v}) \} = R(f)$$

Optimal model is the one which minimizes the risk

$$f^* = \operatorname{argmin}_{f \in \mathbb{H}} R(f)$$

Supervised Learning: *Empirical Risk*

? *We don't know data distribution in general!*

! *We approximate it with the arithmetic average $\equiv$ law of large numbers*

Empirical Risk $\equiv$ Estimate of Risk

Say $y_n = f(x_n)$ for every $(x_n, v_n) \in \mathbb{D}$; then, the empirical risk is

$$\hat{R}(f) = \frac{1}{N} \sum_{n=1}^N \mathcal{L}(y_n, v_n)$$

If N is very large and samples are i.i.d.

$$\hat{R}(f) = \frac{1}{N} \sum_{n=1}^N \mathcal{L}(y_n, v_n) \approx \mathbb{E}_{\mathbf{x}, \mathbf{v}} \{ \mathcal{L}(\mathbf{y}, \mathbf{v}) \} = R(f)$$

Supervised Learning: *General Learning Algorithm*

? *How can we find the optimal model?*

! *We search for minimal empirical risk*

Optimal model is the one which minimizes the empirical risk

$$\begin{aligned} f^* &= \operatorname{argmin}_{f \in \mathbb{H}} \hat{R}(f) \\ &= \operatorname{argmin}_{f \in \mathbb{H}} \frac{1}{N} \sum_{n=1}^N \mathcal{L}(\mathbf{y}_n, \mathbf{v}_n) \end{aligned}$$

Regression Example: *Fitting Polynomial*

- *Labeled dataset*

$$\mathbb{D} = \{(\text{weight}, \text{price}) = (4, 2), (5, 1.5), (4.8, 3), (6, 4)\}$$

- *Model is a function taking weight and returning price*

$$\mathbb{H} = \{f(x) = w_0 + w_1x + \dots + w_Px^P : w_0, \dots, w_P \in \mathbb{R}\}$$

↳ *Hypothesis: v and x are related via a polynomial function*

$$v \approx y = f(x) = \sum_{i=1}^P w_i x^i$$

- *Learning algorithm $\equiv$ empirical risk minimization*

Regression Example: *Fitting Polynomial*

Optimal model is the one which minimizes the empirical risk

$$f^{\star} = \operatorname{argmin}_{f \in \mathbb{H}} \sum_{n=1}^N \frac{1}{N} \mathcal{L}(f(x_n), \mathbf{v}_n)$$

Say $\mathcal{L}(y, v) = (y - v)^2$: the optimal model is

$$f^{\star}(x) = \sum_{i=1}^P w_i^{\star} x^i$$

for $w_0^{\star}, \dots, w_P^{\star}$ that are

$$w_0^{\star}, \dots, w_P^{\star} = \operatorname{argmin}_{w_0, \dots, w_P} \frac{1}{N} \sum_{n=1}^N \mathcal{L}(w_0 + w_1 x_n + \dots + w_P x_n^P - \mathbf{v}_n)^2$$

Regression Example: *Fitting Line*

Let's restrict the **hypothesis** to a line

$$f(x) = wx$$

The empirical risk computed on

$$\mathbb{D} = \{(\text{weight}, \text{price}) = (4, 2), (5, 1.5), (4.8, 3), (6, 4)\}$$

is written as

$$\hat{R}(f) = \frac{1}{4} \left[(4w - 2)^2 + (5w - 1.5)^2 + (4.8w - 3)^2 + (6w - 4)^2 \right]$$

So, we have

$$w^* = \underset{w}{\operatorname{argmin}} \hat{R}(f) \rightsquigarrow f^*(x) = w^*x = 0.53x$$

Regression Example: *Fitting Line*

A Nice Thought Practice

Assume you did know that

$$(x, v) \sim P(x, v)$$

*Think about the **exact risk** and how you could determine it!*

Recall that

$$R(f) = \mathbb{E}_{x, \textcolor{red}{v}} \left\{ (wx - \textcolor{red}{v})^2 \right\}$$

Classification Example: *Separation by Weight*

- *Labeled dataset*

$$\mathbb{D} = \{(\text{weight}, \text{type of pet}) = (5.5, \text{cat}), (4.5, \text{cat}), \\ (12.5, \text{dog}), (9.5, \text{dog}), (7.5, \text{cat})\}$$

- *Model is a function taking weight and returning type*

$$\mathbb{H} = \{f(x) = \text{sign}(x - b) \in \{-1, 1\}\} \rightsquigarrow y = \begin{cases} \text{cat} & f(x) = -1 \\ \text{dog} & f(x) = 1 \end{cases}$$

↳ *Hypothesis*: cat and dog are separated in weight by *thresholding*

- *Learning algorithm* $\equiv$ *empirical risk minimization*

Classification Example: *Separation by Weight*

Optimal model is the one which minimizes the empirical risk

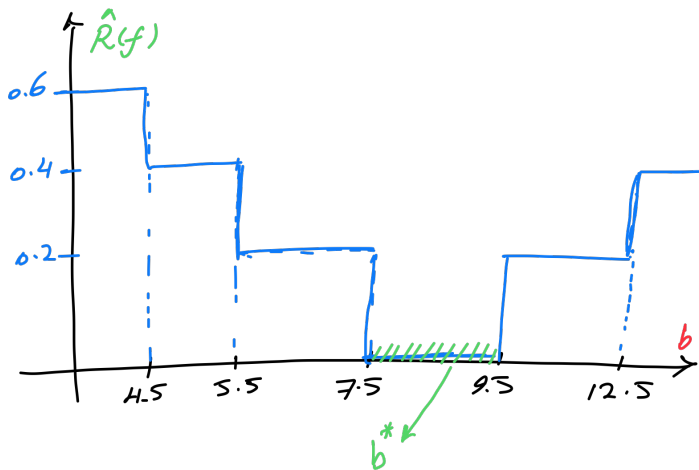
$$f^{\star} = \operatorname{argmin}_{f \in \mathbb{H}} \sum_{n=1}^N \frac{1}{N} \mathcal{L}(f(x_n), \mathbf{v}_n)$$

Say the loss is

$$\mathcal{L}(y, v) = \begin{cases} 1 & y \neq v \\ 0 & y = v \end{cases}$$

Let's find $\hat{R}(f)$

Classification Example: Separation by Weight



Classification Example: *Separation by Weight*

A Nice Thought Practice

Assume you did know that

$$(x, v) \sim P(x, v)$$

*Think about the **exact risk** and how you could determine it!*

Note that in this case

$$R(f) = \mathbb{E}_{x,v} \{ \mathcal{L}(\text{sign}(x - b), v) \} = \Pr \{ \text{sign}(x - b) \neq v \}$$

Supervised Learning: *Maximum Likelihood Learning*

? *Can't we use maximum likelihood?*

Let for a sample $(\mathbf{x}, \mathbf{v})$

$$\mathbf{v} \sim P_{\boldsymbol{\theta}}(\mathbf{v}|\mathbf{x})$$

The average log-likelihood is then given by

$$\mathcal{S}(\boldsymbol{\theta}) = \frac{1}{N} \sum_{n=1}^N \log P_{\boldsymbol{\theta}}(\mathbf{v}_n|\mathbf{x}_n)$$

We find the optimal model as

$$\boldsymbol{\theta}^{\star} = \operatorname{argmax}_{\boldsymbol{\theta}} \mathcal{S}(\boldsymbol{\theta})$$

! *We will see that this is equivalent to risk minimization*

Regression via Linear Models

Let's think of

$$\mathbb{D} = \left\{ \left(\mathbf{x}_n \in \mathbb{R}^d, v_n \in \mathbb{R} \right) : n = 1, \dots, N \right\}$$

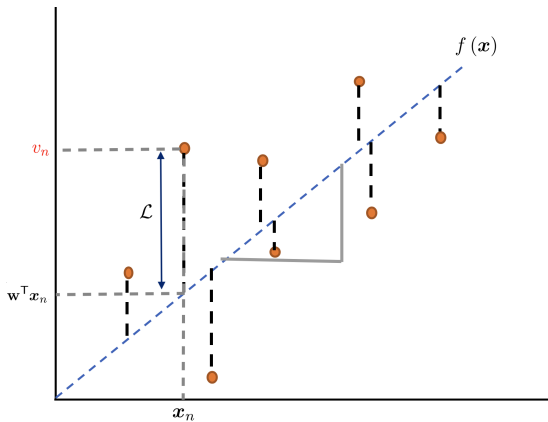
We focus on *linear models*

$$\mathbb{H} = \left\{ f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} \text{ for all } \mathbf{w} \in \mathbb{R}^d \right\}$$

Optimal linear model is

$$\mathbf{w}^\star = \operatorname{argmin}_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{N} \sum_{n=1}^N \mathcal{L} \left(\mathbf{w}^\top \mathbf{x}_n, v_n \right)$$

Regression via Linear Models: *Visualization*



Linear Regression: *Empirical Risk*

Typical choice of the loss function

$$\mathcal{L}(y_n, v_n) = (y_n - v_n)^2$$

So the empirical risk is

$$\hat{R}(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^N \left(\mathbf{w}^\top \mathbf{x}_n - v_n \right)^2$$

Attention

All vectors in these slides are column vectors. We use transpose to make them row vectors: column $\equiv \mathbf{x}_n \rightsquigarrow \mathbf{x}_n^\top \equiv$ row

Linear Regression: *Vectorized Empirical Risk*

We can collect the dataset into a matrix

$$\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]$$

So we have

$$\begin{aligned}\mathbf{w}^\top \mathbf{X} &= \mathbf{w}^\top [\mathbf{x}_1, \dots, \mathbf{x}_N] \\ &= [\mathbf{w}^\top \mathbf{x}_1, \dots, \mathbf{w}^\top \mathbf{x}_N] \\ &= [y_1, \dots, y_N] = \mathbf{y}^\top\end{aligned}$$

So, we have

$$\begin{aligned}\hat{R}(\mathbf{w}) &= \frac{1}{N} \sum_{n=1}^N (y_n - v_n)^2 = \frac{1}{N} \|\mathbf{y} - \mathbf{v}\|^2 \\ &= \frac{1}{N} \|\mathbf{X}^\top \mathbf{w} - \mathbf{v}\|^2\end{aligned}$$

Linear Regression: *Empirical Risk Minimization*

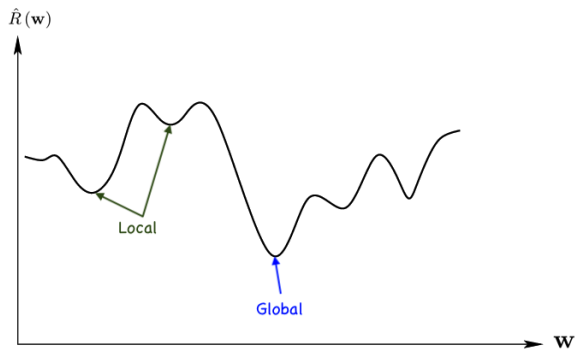
Optimal linear model is the $\mathbf{w}$ that minimized the empirical risk

$$\mathbf{w}^* = \operatorname{argmin}_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{N} \|\mathbf{X}^T \mathbf{w} - \mathbf{v}\|^2$$

? *How to optimize it?*

Global and Local Minimum

Empirical risk can have local and global minima



Note!

In this figure, we think of w to be a scalar!

Stationary Points

? *How can we find those points?*

! *They are again **stationary points**!*

Stationary Points

The points at which derivative of the function is zero

$$\frac{d\hat{R}}{d\mathbf{w}} = 0$$

Stationary point is where the slope is zero

- *Minimum*
- *Maximum*
- *Inflection*

Multivariate Functions: *Gradient*

? *But we have a multivariate function!*

$$\nabla \hat{R} = \begin{bmatrix} \frac{\partial \hat{R}}{\partial w_1} \\ \vdots \\ \frac{\partial \hat{R}}{\partial w_d} \end{bmatrix}$$

Stationary Points

At stationary points the gradient of the function is vector of zero $\nabla \hat{R} = \mathbf{0}$

- *Minimum*
- *Maximum*
- *Saddle Point*

Stationary Points

? *How can we find the global minimum?*

Naive Algorithm $\equiv$ Exhaustive Search

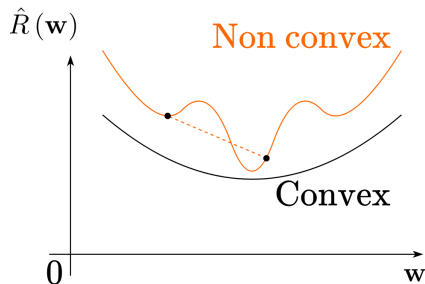
- ① *Find all stationary points*
 - ② *Select those that are minimum*
 - ③ *Take the smallest one*
-
- ! *Not really easy to find stationary points with a very complex function*
 - ↳ *We will see this in next lectures!*
 - ! *The number of such points can grow **exponentially** with dimension*
 - ↳ *This is usually computationally infeasible!*

Convex Optimization

In linear regression, we are lucky since

$$\hat{R}(\mathbf{w}) = \frac{1}{N} \|\mathbf{X}^T \mathbf{w} - \mathbf{v}\|^2$$

is **convex**: it has only one minimum which is both **local** and **global**



Linear Regression: *Empirical Risk Minimization*

Since we know that it's convex, we can find the optimal model by

$$\nabla \hat{R} = \frac{1}{N} \nabla \|\mathbf{X}^T \mathbf{w} - \mathbf{v}\|^2 = \mathbf{0}$$

With a bit of computation, we can see

$$\nabla \|\mathbf{X}^T \mathbf{w} - \mathbf{v}\|^2 = 2 \left(\mathbf{X} \mathbf{X}^T \mathbf{w} - \mathbf{X} \mathbf{v} \right)$$

So, we should find

$$\mathbf{X} \mathbf{X}^T \mathbf{w}^* = \mathbf{X} \mathbf{v}$$

Note!

Replace in the empirical risk and you see it makes sense in special cases!

Linear Regression: *Empirical Risk Minimization*

This is a linear system of d equations with d unknowns

$$\mathbf{X}\mathbf{X}^T \mathbf{w}^\star = \mathbf{X}\mathbf{v}$$

- We can solve it if equations are linearly independent, i.e., $\det \mathbf{X}\mathbf{X}^T \neq 0$
 - ↳ Usually the case when $N \geq d$
- If multiple equations are linearly dependent: we have no unique solution
 - ↳ *Always* the case when $N < d$

When $\det \mathbf{X}\mathbf{X}^T \neq 0$, we can write

$$\mathbf{w}^\star = \left(\mathbf{X}\mathbf{X}^T\right)^{-1} \mathbf{X}\mathbf{v} = \mathbf{X}^\dagger \mathbf{v}$$

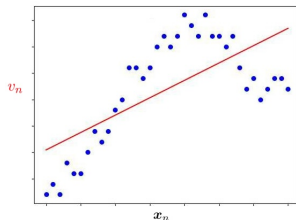
$\mathbf{X}^\dagger$ is the *pseudo-inverse* of $\mathbf{X}$

Linear Regression: *Optimal Model*

So, the optimal model is given by

$$f^*(x) = \underbrace{x^\top}_{\text{new sample out of } \mathbb{D}} \underbrace{\mathbf{X}^\dagger \mathbf{v}}_{\text{computed from } \mathbb{D}}$$

- We can check its generalization by testing it on a test set $\mathbb{T}$
 - ↳ More about it later
- It's only a *linear model* and is not guaranteed to generalize well
 - ↳ If data shows *extreme* nonlinearity, it doesn't work well!



Linear Regression with *Affine Models*

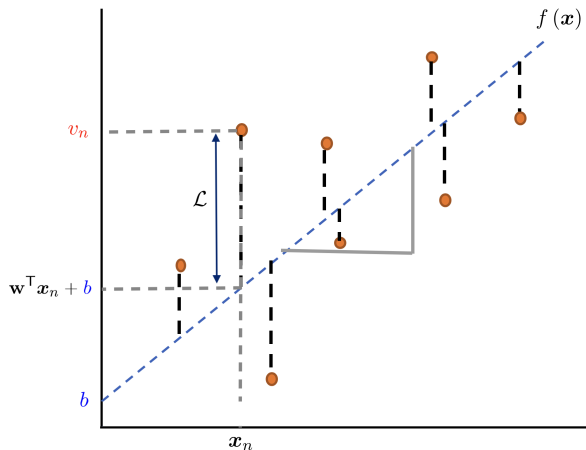
We may also include a **bias** in the linear model

$$\mathbb{H} = \left\{ f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b \text{ for all } \mathbf{w} \in \mathbb{R}^d \text{ and } b \in \mathbb{R} \right\}$$

Optimal linear model is

$$\mathbf{w}^\star, b^\star = \operatorname{argmin}_{\mathbf{w} \in \mathbb{R}^d, b \in \mathbb{R}} \frac{1}{N} \sum_{n=1}^N \left(\mathbf{w}^\top \mathbf{x}_n + b - v_n \right)^2$$

Regression via Affine: *Visualization*



Linear Regression with *Affine Models*

We can interpret it as a linear model again

$$\mathbf{w}^\top \mathbf{x} + b = \begin{bmatrix} b & \mathbf{w}^\top \end{bmatrix} \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix}$$

So, we can make a new dataset matrix as

$$\mathbf{X} = \begin{bmatrix} 1 & \dots & 1 \\ \mathbf{x}_1 & \dots & \mathbf{x}_N \end{bmatrix}$$

and everything goes as before

$$\begin{bmatrix} b^\star \\ \mathbf{w}^\star \end{bmatrix} = \left(\mathbf{X} \mathbf{X}^\top \right)^{-1} \mathbf{X} \mathbf{v} = \mathbf{X}^\dagger \mathbf{v}$$

Linear Regression with *Vector Labels*

We could also have

$$\mathbb{D} = \left\{ \left(\mathbf{x}_n \in \mathbb{R}^d, \mathbf{v}_n \in \mathbb{R}^\ell \right) : n = 1, \dots, N \right\}$$

We could extend everything to higher dimensions

$$\mathbb{H} = \left\{ f(\mathbf{x}) = \mathbf{W}^T \mathbf{x} + \mathbf{b} \text{ for all } \mathbf{W} \in \mathbb{R}^{d \times \ell} \text{ and } \mathbf{b} \in \mathbb{R}^\ell \right\}$$

Everything in this case is again as in scalar case

A Nice Practice

Write the optimal model in this case. You just need to adjust dimensions!

Complexity of Computing Optimal Model

Finding optimal model can be computationally expensive with big datasets

- Finding $\mathbf{X}\mathbf{X}^\top$ needs $\mathcal{O}(d^2 N)$ computations
- Finding $(\mathbf{X}\mathbf{X}^\top)^{-1}$ needs between $\mathcal{O}(d^{2.4})$ and $\mathcal{O}(d^3)$ computations
- Finding $(\mathbf{X}\mathbf{X}^\top)^{-1} \mathbf{X}$ needs $\mathcal{O}(d^2 N)$ computations
- We need at least in order of d data samples

So we have around $\mathcal{O}(d^3)$ complexity!

Not Good with Large-dimensional Data!

If data is very big, i.e., d extremely large, it's not a good approach!

Maximum Likelihood Learning

We can instead think of a model for $P(v|x)$: *let's assume*

$$v = \mathbf{w}^T \mathbf{x} + z$$

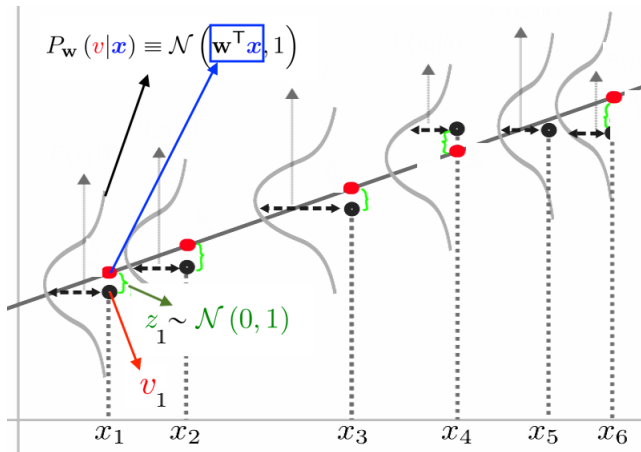
where z is a Gaussian *innovation* or *perturbation* term

$$z \sim \mathcal{N}(0, 1)$$

Then, we have

$$P_{\mathbf{w}}(v|x) \equiv \mathcal{N}(\mathbf{w}^T \mathbf{x}, 1)$$

Maximum Likelihood Learning



Maximum Likelihood Learning

We can write the conditional distribution as

$$P_{\mathbf{w}}(\mathbf{v}|\mathbf{x}) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(\mathbf{v} - \mathbf{w}^T \mathbf{x})^2}{2}}$$

and the log-likelihood as

$$\log P_{\mathbf{w}}(\mathbf{v}|\mathbf{x}) = -\frac{1}{2} \log 2\pi - \frac{(\mathbf{v} - \mathbf{w}^T \mathbf{x})^2}{2}$$

Maximum likelihood learning will then be

$$\begin{aligned} \max_{\mathbf{w}} \mathcal{S}(\mathbf{w}) &= \max_{\mathbf{w}} \frac{1}{N} \sum_{n=1}^N \log P_{\mathbf{w}}(\mathbf{v}_n|\mathbf{x}_n) \\ &\equiv \min_{\mathbf{w}} \frac{1}{N} \sum_{n=1}^N (\mathbf{v}_n - \mathbf{w}^T \mathbf{x}_n)^2 = \min_{f \in \mathcal{H}} \hat{R}(f) \end{aligned}$$

Maximum Likelihood vs Risk Minimization

Maximum likelihood learning does

$$\min_{\mathbf{w}} \frac{1}{N} \sum_{n=1}^N \left(\textcolor{red}{v} - \mathbf{w}^T \textcolor{blue}{x} \right)^2$$

In this expression, we have

- $\left(\textcolor{red}{v} - \mathbf{w}^T \textcolor{blue}{x} \right)^2$ because we assumed *Gaussian perturbation*
 - ↳ This is imposed by *quadratic loss* in risk minimization
- $\mathbf{w}^T \textcolor{blue}{x}$ because we assumed linear model for mean
 - ↳ This is imposed by *hypothesis* in risk minimization

Moral of Story

Maximum Likelihood with $P_{\mathbf{w}} \equiv$ Risk Minimization with $\mathbb{H}_{\mathbf{w}}$ and $\mathcal{L}$

Further Read

- Bishop
 - ↳ Chapter 3: *Sections 3.1 – 3.3*
- ESL
 - ↳ Chapter 3: *Section 3.1 – 3.2*
- Goodfellow
 - ↳ Chapter 5: *Section 5.7*

Linear Regression

Linear Regression

Supervised Learning