Lecture 4 ML & Optimization

- Independence ⇒ Laws of Large Numbers (LLN)
- LLN justifies ERM aka learning from data

Independent random variables & their sums

$X \perp\!\!\!\perp Y \iff p(x,y) = p(x)p(y)$

Fact:

$$\mathbb{E}[XY] = \mathbb{E}[X]\,\mathbb{E}[Y]$$

Pf

$$\mathbb{E}[XY] = \int_{\mathbb{R}^2} xy\, p(x,y)\,dx\,dy = \int_{\mathbb{R}^2} x\,p(x)\; y\,p(y)\,dx\,dy$$

$$= \int x\,p(x)\,dx \int y\,p(y)\,dy = \mathbb{E}[X]\,\mathbb{E}[Y]$$

Actually,

$$\forall f,g:\ \mathbb{E}[f(X)g(Y)] = \mathbb{E}[f(X)]\,\mathbb{E}[g(Y)]$$

$$\iff X \perp\!\!\!\perp Y$$

Fact if $X_1, \dots, X_K$ are r.v.s, then

$$\mathbb{E}[X_1 + \cdots + X_K] = \mathbb{E}[X_1] + \cdots + \mathbb{E}[X_K]$$

in fact

$$\mathbb{E}[a_1 X_1 + \cdots + a_K X_K] = a_1 \mathbb{E}[X_1] + \cdots + a_K \mathbb{E}[X_K]$$

Notation:

We say $X_1, \dots, X_K$ are i.i.d. (independent and identically distributed) if:

- $\forall i$, $X_i \sim \mathbb{P}$: identically distributed
- $p(x_1, \dots, x_K) = p(x_1) \cdots p(x_K)$: independence

Supervised ML learns models f using i.i.d. samples $(x_i, y_i) \sim \mathbb{P}$

Variance of sums of r.v.s

① **Fact:** $X_1, \dots, X_K$ are independent

$$\Rightarrow \quad \mathrm{Var}(X_1 + \cdots + X_K) = \mathrm{Var}(X_1) + \cdots + \mathrm{Var}(X_K)$$

② **Fact:**

$$\mathrm{Var}(aX) = a^2 \mathrm{Var}(X)$$

Prf 2

$$\mathrm{Var}(aX) = \mathbb{E}(aX - a\mathbb{E}X)^2 = \mathbb{E}[a^2(X - \mathbb{E}X)^2]$$

$$= a^2 \mathbb{E}(X - \mathbb{E}X)^2 = a^2 \mathrm{Var}(X)$$

Prf(1)

$$\mathrm{Var}(X_1+\cdots+X_K) = \mathbb{E}(X_1+\cdots+X_K)^2 - \left[\mathbb{E}(X_1+\cdots+X_K)\right]^2$$

$$= \mathbb{E}\left[\sum_{i=1}^{K}\sum_{j=1}^{K} X_i X_j\right] - \left[\sum_{i=1}^{K}\mathbb{E}[X_i]\right]^2$$

$$= \sum_{i=1}^{K}\mathbb{E}[X_i^2] + \sum_{i=1}^{K}\sum_{\substack{j=1\\ j\neq i}}^{K}\mathbb{E}[X_i]\mathbb{E}[X_j]$$

$$- \sum_{i=1}^{K}\sum_{j=1}^{K}\mathbb{E}[X_i]\mathbb{E}[X_j]$$

$$= \sum_{i=1}^{K}\mathbb{E}[X_i^2] - \sum_{i=1}^{K}\left(\mathbb{E}[X_i]\right)^2 + \sum_{i=1}^{K}\sum_{\substack{j=1\\ j\neq i}}^{K}\mathbb{E}[X_i]\mathbb{E}[X_j]$$

$$- \sum_{i=1}^{K}\sum_{\substack{j=1\\ j\neq i}}^{K}\mathbb{E}[X_i]\mathbb{E}[X_j] = \sum_{i=1}^{K}\mathrm{Var}(X_i)$$

∎

Behavior of Sample Mean

Sample $X_1, \dots, X_n$ i.i.d. from $\mathbb{P}$

and take $$\mu_n = \frac{1}{n} \sum_{i=1}^{n} X_i$$

Fact 1

$$\mathbb{E}\mu_n = \mathbb{E}X_1 = \cdots = \mathbb{E}X_n = \mu \quad \left(\text{where } \mu = \mathbb{E}_{X \sim \mathbb{P}} X\right)$$

Pf

$$\mathbb{E}\mu_n = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}X_i = \frac{1}{n} \sum_{i=1}^{n} \mu = \mu$$

Fact 2

$$\mathrm{Var}(\mu_n) = \frac{1}{n}\sigma^2 \quad \left(\text{where } \sigma^2 = \mathrm{Var}(X) \text{ when } X \sim \mathbb{P}\right)$$

↳

(cont'd)

$$\mathrm{Var}(\mu_n) = \mathrm{Var}\left(\sum_{i=1}^{n} \frac{1}{n} X_i\right) = \sum_{i=1}^{n} \mathrm{Var}\left(\frac{1}{n} X_i\right)$$

$$= \sum_{i=1}^{n} \frac{1}{n^2} \mathrm{Var}(X_i) = \sum_{i=1}^{n} \frac{\sigma^2}{n^2}$$

$$= \frac{\sigma^2}{n}$$

Cor LLN (see Chebyshev's ineq)

If $X_1, \dots, X_n \sim \mathbb{P}$ are i.i.d. and $\varepsilon > 0$ is arbitrary

$$\mathbb{P}(|\mu_n - \mu| > \varepsilon) \to 0 \quad \text{as } n \to \infty$$

LLN justifies ERM

In ML we want to minimize $\mu(f) = \mathbb{E}_{(x,y)\sim \mathbb{P}} \ell(f(x), y)$

but we can't because we don't know

$\mathbb{P}$ or we can't solve that optim problem.

Instead:

- sample $(x_i, y_i) \sim \mathbb{P}$ i.i.d. for $i = 1, \dots, n$
- form $\mu_n(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i)$
  and minimize μ_n

We know $\mu_n(f) \to \mu(f)$ as $n \to \infty$ by LLN, so

ERM gives same minimizer as population risk

Parameterized ML models

In ML we want to learn

$\mathbb{E}[y|x] = f(x)$ — regression function

$p(y|x) = g(x, y)$ — conditional distributive (predictive)

In general, these are very complicated, so we use simplified but useful parameterized ML models

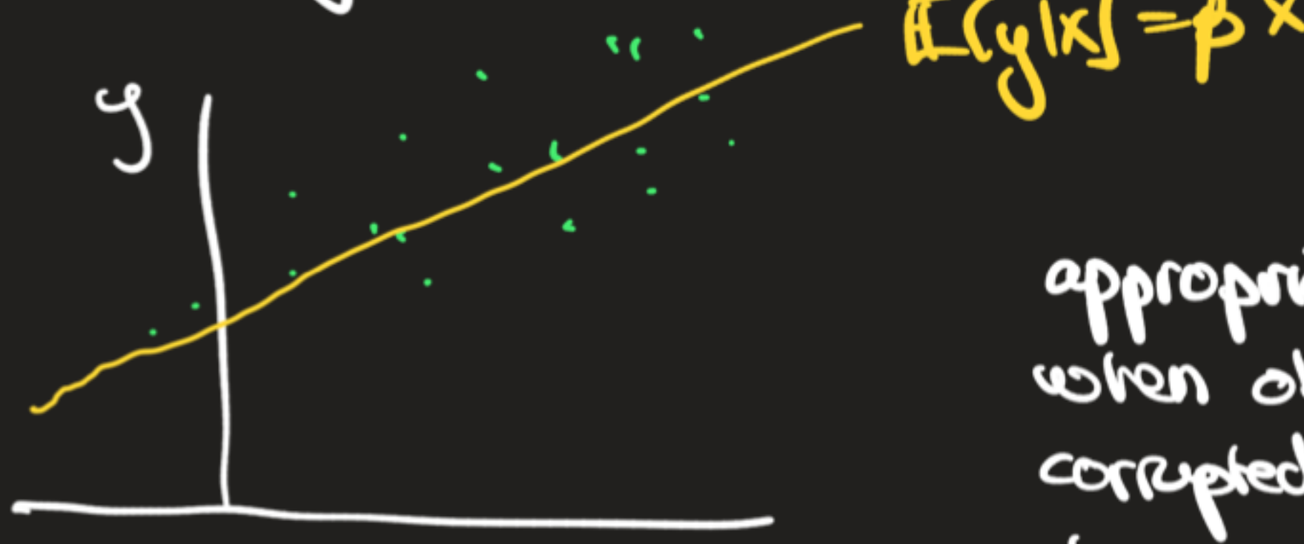
e.g. to learn

$\mathbb{E}[y|x] = f(x; \theta)$ where $\theta \in \mathbb{R}^p$ is a set of parameters we optimize over

Gaussian noise model for regression

Assume: $y|x \sim \mathcal{N}(\beta^T x, \sigma^2)$

$\Rightarrow \mathbb{E}[y|x] = \beta^T x$



Fitting: learn β using OLS

Predicting: given x_{new}, take $y_{pred} = \beta^T x_{new}$

appropriate for regression when observations look corrupted by Gaussian noise whose variance does not depend on x, and there is a linear trend in x

Classification w/ Bernoulli noise model

Assumption: $y|x \sim \text{Bern}(p(x))$

useful for binary classification (e.g. spam detection)

$$\mathbb{E}[y|x] = p(x)$$

where we require $p(x) \in [0,1]$. To ensure this, we take

$$p(x) = \sigma(\underbrace{\theta^T x}_{\text{logit}})$$ where σ is the logistic sigmoid

$$\sigma(v) = \frac{1}{1+e^{-v}}$$

Very used for binary classification (maybe w/ a different logit function)

Fitting: logistic regression (next class)

Predicting: if $p(x_{new}) >$ threshold, return 1 else 0

Categorical regression (multiclass classification) model

A r.v. $z \sim \text{Categorical}(p_1, \dots, p_K)$ if

$$\mathbb{P}(z = i) = p_i \quad \text{for } i = 1, \dots, K$$

where $p_i \geq 0$ and $\sum p_i = 1$

We use this dist in MC for classification

Assumption: $y \mid x \sim \text{Categorical}(p_1(x), \dots, p_K(x))$

s.t. $p_i(x) \geq 0$ and $\sum p_i(x) = 1$

To ensure this, take

$$p_i(x) = \frac{e^{\theta_i^T x}}{\sum_{j=1}^{K} e^{\theta_j^T x}} = \frac{e^{\theta_i^T x}}{Z_\theta(x)}$$

$\theta_i^T x$ ← logit of class i

$Z_\theta(x) = \sum_{j=1}^{K} e^{\theta_j^T x}$ is called the partition function or normalizing constant

Fitting: multinomial logistic regression

Predicting: given x_{new}, predict $y_{pred} = \underset{i}{\operatorname{argmax}}\ p(x_{new})_i$

Some notation:

- Instead of letting $y = 1, \dots, K$, encode the classes with one-hot encoding $y = e_1, \dots, e_K$ where $e_i = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^K$ (ith position)
- Given a vector x, define $e^x = \begin{bmatrix} e^{x_1} \\ \vdots \\ e^{x_K} \end{bmatrix}$
- Given $\theta_i \in \mathbb{R}^d$ for $i = 1, \dots, K$, define
  $\Theta = \begin{bmatrix} \theta_1^T \\ \vdots \\ \theta_K^T \end{bmatrix} \in \mathbb{R}^{K \times d}$ as our parameter matrix
- Let $\mathbf{1} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^K$ denote the vector of all ones

Now we can write $p(x) = p_\theta(x)$

$$p_\theta(x) = \begin{bmatrix} p_1(x) \\ \vdots \\ p_k(x) \end{bmatrix} = \begin{bmatrix} e^{\theta_1^T x}/Z_\theta(x) \\ \vdots \\ e^{\theta_k^T x}/Z_\theta(x) \end{bmatrix} = \frac{1}{Z_\theta(x)} \begin{bmatrix} e^{\theta_1^T x} \\ \vdots \\ e^{\theta_k^T x} \end{bmatrix}$$

$$= \frac{e^{\theta x}}{Z_\theta(x)} = \frac{e^{\theta x}}{1^T e^{\theta x}} = \text{softmax}(\theta x)$$

$$\left(\text{softmax}: v \to \frac{e^v}{1^T e^v}\right)$$

Ex

$$\begin{bmatrix} 30 \\ 1 \\ 0 \end{bmatrix} \xrightarrow{\text{softmax}} \begin{bmatrix} 1 \\ 2.54 \times 10^{-13} \\ 9.36 \times 10^{-14} \end{bmatrix} \qquad \begin{bmatrix} 0.9 \\ 0.05 \\ 0.05 \end{bmatrix} \xrightarrow{\text{softmax}} \begin{bmatrix} 0.539 \\ 0.23 \\ 0.23 \end{bmatrix}$$

Prediction: $y_{pred} = \underset{i}{\text{argmax}} \ (\theta x)_i$

Notice

$$\mathbb{E}[y|x] = \sum_{i=1}^{k} e_i [p_\theta(x)]_i = p_\theta(x)$$