Machine Learning and Data Mining

Recapitulation

Maxim Borisyak

National Research University Higher School of Economics (HSE)

September 11, 2018

Statistical estimations

Setup

Given:

- data: $X = \{x_i\}_{i=1}^{N}$;
- parameterized family of distributions $P(x \mid \theta)$.

Problem:

- estimate θ.

Maximum likelihood estimation

$$\begin{aligned} L(\theta) &= P(X \mid \theta); \\ \hat{\theta} &= \arg\max_{\theta} L(\theta). \end{aligned}$$

$$\mathcal{L}(\theta) = -\log \prod_i P(x_i \mid \theta) = -\sum_i \log P(x_i \mid \theta)$$

- consistent estimation: $\hat{\theta} \to \theta$ as $N \to \infty$;
- *might be biased*;
- equal to MAP estimation with uniform prior.

MLE: example

Given samples $\{x_i\}_{i=1}^N$ from a normal distribution estimate its mean.

$$\mu = \arg\min_{\mu} \mathcal{L}(X) =$$

$$\arg\min_{m} u - \sum_i \log\left(\frac{1}{Z}\exp\left[-\frac{(x_i-\mu)^2}{2\sigma^2}\right]\right) =$$

$$\arg\min_{\mu} \sum_i (x_i-\mu)^2 = \frac{1}{N}\sum_i x_i$$

Bayesian inference

$$P(\theta \mid X) = \frac{1}{Z} P(X \mid \theta) P(\theta);$$

- often, posterior distribution of predictions is of the main interest:

$$P(f(x) = y \mid X) = \int \mathbb{I}\left[f(x, \theta) = y\right] P(\theta \mid X) \; d\theta$$

- with a few exceptions posterior is intractable;
- often, approximate inference is utilized instead.

BI: example

Given samples $\{x_i\}_{i=1}^N$ from a normal distribution estimate mean under a normal prior.

$$P(\mu \mid X) = \frac{1}{Z} P(X \mid \mu) P(\mu) =$$

$$\frac{1}{Z} \exp\left[-\frac{\mu^2}{2c^2}\right] \cdot \prod \exp\left[-\frac{(x_i - \mu)^2}{2\sigma^2}\right]$$

$$\log P(\mu \mid X) = -Z - \frac{\mu^2}{2c^2} - \sum_i \frac{(x_i - \mu)^2}{2\sigma^2}$$

Maximum a posteriori estimation

$$\hat{\theta} = \arg\max_{\theta} P(\theta \mid X) = \arg\max_{\theta} P(X \mid \theta) P(\theta) =$$

$$\arg\min_{\theta} \left[- \log P(X \mid \theta) - \log P(\theta)\right] =$$

$$\arg\min_{\theta} \left[\text{neg log likelihood} + \text{penalty}\right]$$

$$\hat{\theta} = \arg\min_{\theta} \left[- \log P(\theta) - \sum_i \log P(x_i \mid \theta)\right]$$

- sometimes called **structural loss**:
    - i.e. includes 'structure' of the predictor into the loss.

MAP: example

Given samples $\{x_i\}_{i=1}^N$ from a normal distribution estimate mean under a normal prior.

$$\hat{\mu} = \arg\max_{\mu} \log P(\mu \mid X) =$$

$$\arg\max_{\mu} \left[-Z - \frac{\mu^2}{2c^2} - \sum_i \frac{(x_i - \mu)^2}{2\sigma^2} \right] =$$

$$\arg\min_{\mu} \left[\lambda\mu^2 + \sum_i (x_i - \mu)^2 \right] = \frac{1}{N + \lambda} \sum_i x_i$$

Machine Learning

Structure of a Machine Learning problem

Given:

- description of the problem:
  - prior knowledge;
- data:
  - input space: $\mathcal{X}$;
  - output space: $\mathcal{Y}$;
- metric M.

Problem:

- find a learning algorithm: $A : \mathcal{D} \to (\mathcal{X} \to \mathcal{Y})$ such that:

$$M(A(\mathrm{data})) \to \max$$

Differences from statistics

Machine Learning:

- usually, probability densities are intractable;
- high-dimensionality/small sample sizes;
- hence, no p-values etc;
- less formal assumptions.

Supervised learning

Regression

Output: $y \in \mathbb{R}$.

Assumptions:

- $y = f(x) + \varepsilon(x)$;
- $\varepsilon(x)$ - noise:
    - $\forall x_1, x_2 : x_1 \neq x_2 \Rightarrow \varepsilon(x_1)$ independent from $\varepsilon(x_2)$;
    - $\forall x : \mathbb{E}\,\varepsilon(x) = 0$.
- often, $\varepsilon(x)$ is assumed not to be dependent on x.

Regression loss

$$\mathcal{L}(f) = -\sum_i \log P((x_i, y_i) \mid f) =$$

$$-\sum_i \log P_\varepsilon(y_i - f(x_i) \mid f, x_i) =$$

$$-\sum_i \log P_\varepsilon(y_i - f(x_i) \mid x_i)$$

Regression: MSE

- $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$;
- $\sigma_\varepsilon^2 = \mathrm{const}$ (unknown);

$$\mathcal{L}(f) = -\sum_i \log P_\varepsilon(y_i - f(x_i) \mid x_i) =$$

$$\sum_i \left[Z(\sigma_\varepsilon^2) - \frac{(y_i - f(x_i))^2}{2\sigma_\varepsilon^2} \right] \sim$$

$$\sum_i (y_i - f(x_i))^2 \to \min$$

$$f^*(x) = \mathbb{E}\left[y \mid x\right]$$

Regression: MAE

- $\varepsilon \sim \mathrm{Laplace}(0, b_\varepsilon)$;
- $b_\varepsilon = \mathrm{const}$ (unknown);

$$\mathcal{L}(f) = -\sum_i \log P_\varepsilon(y_i - f(x_i) \mid x_i) =$$

$$\sum_i \left[Z(b_\varepsilon) - \frac{|y_i - f(x_i)|}{2b_\varepsilon} \right] \sim$$

$$\sum_i |y_i - f(x_i)| \to \min$$

$$f^*(x) = \mathrm{median}\left[y \mid x \right]$$

Linear regression

$$f(x) = w \cdot x$$

Linear regression + MSE + MLE

$$
\begin{aligned}
\mathcal{L}(w) &= \sum_i (w \cdot x_i - y_i)^2 = \|Xw - y\|^2 \to \min; \\
\frac{\partial}{\partial w}\mathcal{L}(w) &= 2X^T(Xw - y) = 0; \\
w &= (X^T X)^{-1} X^T y.
\end{aligned}
$$

Linear regression + MSE + MAP

$$
\begin{aligned}
\mathcal{L}(w) &= \sum_i (w \cdot x_i - y_i)^2 + \lambda \|w\|^2 = \\
& \|Xw - y\|^2 + \lambda \|w\|^2 \to \min; \\
\frac{\partial}{\partial w}\mathcal{L}(w) &= 2X^T(Xw - y) + 2\lambda w = 0; \\
w &= (X^T X + \lambda I)^{-1} X^T y.
\end{aligned}
$$

Linear regression + MSE + Bayesian Inference

- prior:

$$w \sim \mathcal{N}(0, \Sigma_w);$$

- data model:

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

Linear regression + MSE + Bayesian Inference

$$P(w \mid y, X) \propto P(y \mid w, X)P(w) \propto$$

$$\exp\left[-\frac{1}{2\sigma_\varepsilon^2}(y - Xw)^T(y - Xw)\right] \cdot \exp\left[-\frac{1}{2}w^T\Sigma_w^{-1}w\right] =$$

$$\exp\left[-\frac{1}{2}(w - w^*)^T A_w (w - w^*)\right]$$

where:

- $A_w = \frac{1}{\sigma_\varepsilon^2}XX^T + \Sigma_w^{-1}$;
- $w^* = \frac{1}{\sigma_\varepsilon^2}A_w^{-1}Xy$.

Linear regression + MSE + Bayesian Inference

To make prediction y' in point x':

$$P(y' \mid y, X, x') = \int P(y' \mid w, x') P(w \mid X, y) = \mathcal{N}\left(\frac{1}{\sigma_\varepsilon^2} x'^T A^{-1} X y, x'^T A^{-1} x'\right)$$

Basis expansion

To capture more complex dependencies basis functions can be introduced:

$$f(x) = \sum_i w \cdot \phi(x)$$

where:

- $\phi(x) \in \mathbb{R}^K$ - expanded basis.
- ϕ is fixed.

Basis expansion: example

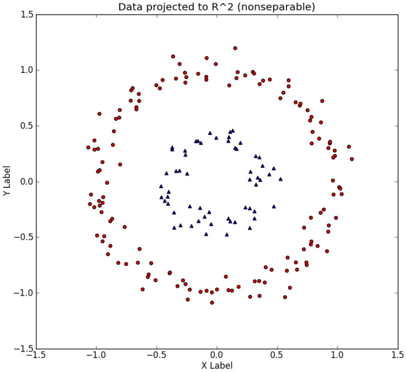
Regression with polynomials:

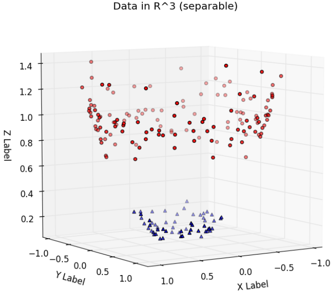
$$\phi(x) = \{1, x_1, \ldots, x_n, x_1^2, x_1 x_2, \ldots, x_n^2, \ldots\}$$

Periodic functions:

$$\phi(x) = \{1, \sin(x), \cos(x), \sin(2x), \cos(2x), \ldots\}$$

Basis expansion: example





Source: `eric-kim.net`

Classification

- classes: $y \in \{1, 2, \ldots, m\}$;
- classifier:

$$f : \mathcal{X} \to \mathbb{R}^m;$$
$$\sum_{k=1}^{m} f^k(x) = 1.$$

$$\mathcal{L}(f) = -\sum_i \sum_{k=1}^{m} \mathbb{I}[y_i = k] \log f^k(x_i);$$
$$\text{cross-entopy}(f) = \sum_i y_i' \cdot f(x_i).$$

Softmax

- often employed trick to make $f(x)$ a proper distribution:

$$\begin{aligned} f(x) &= \mathrm{softmax}(g(x)); \\ f^i(x) &= \frac{\exp(g^i(x))}{\sum_k \exp(g^k(x))}. \end{aligned}$$

Logistic regression

$$\begin{aligned} g(x) &= Wx + b; \\ f(x) &= \mathrm{softmax}(g(x)). \end{aligned}$$

Another form:

$$\frac{\log P(y = i \mid x)}{\log P(y = j \mid x)} = \frac{w_i \cdot x + b_i}{w_j \cdot x + b_j}$$

Logistic regression: 2 classes

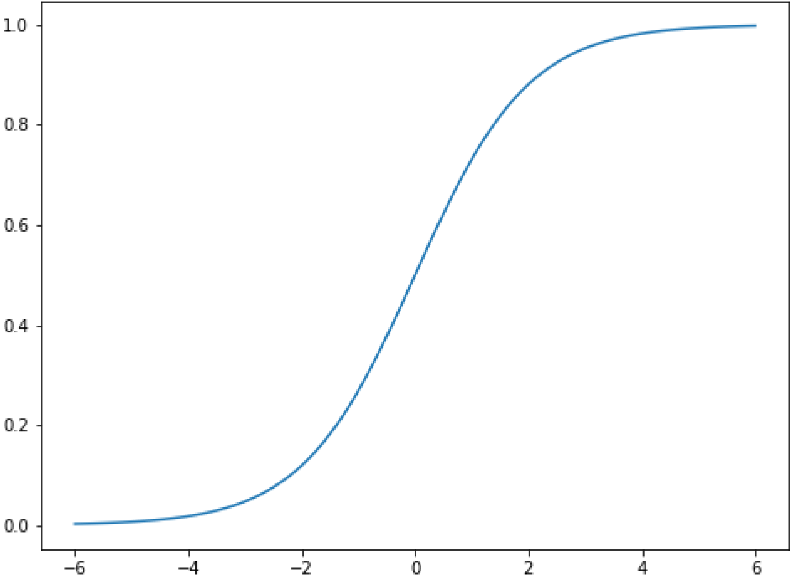
$$f_1(x) = \frac{\exp(w_1 \cdot x + b_1)}{\exp(w_1 \cdot x + b_1) + \exp(w_2 \cdot x + b_2)} =$$

$$\frac{1}{1 + \exp((w_2 - w_1) \cdot x + b_2 - b_1)} =$$

$$\frac{1}{1 + \exp(w' \cdot x + b')} =$$

$$\mathrm{sigmoid}(w' \cdot x + b').$$

Logistic regression: 2 classes



Training logistic regression

$$\mathcal{L}(w) =$$
$$\sum_i \mathbb{I}[y_i = 1] \log(1+\exp(wx_i+b)) + \mathbb{I}[y_i = 0] \log(1+\exp(-wx_i-b))$$

- has no analytical solution;
- smooth and convex.

Gradient Descent

$$f(\theta) \to \min;$$
$$\theta^* = \arg\min_{\theta} f(\theta).$$

$$\begin{aligned}
\theta^{t+1} &= \theta^t - \alpha \nabla f(\theta^t); \\
\theta^t &\to \theta^*, t \to \infty;
\end{aligned}$$

Gradient Descent

$\theta :=$ initialization

for $t := 1, \ldots$ **do**

  $\theta := \theta - \alpha \nabla f(\theta^t)$

end for

Stochastic Gradient Descent

$$f(\theta) = \sum_{i=1}^{N} f_i(\theta)$$

$\theta :=$ initialization

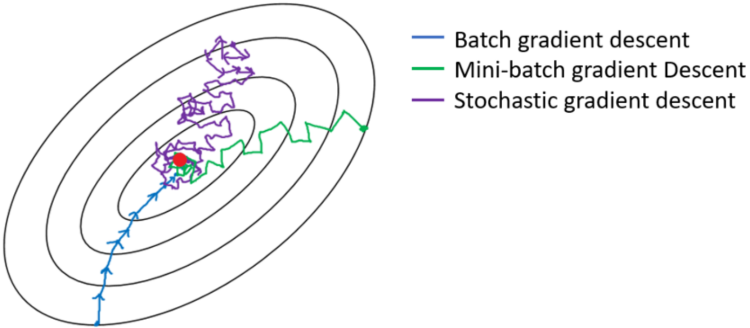
for $t := 1, \ldots$ **do**

$\quad i := \mathrm{random}(1, \ldots, N)$

$\quad \theta := \theta - \alpha \nabla f_i(\theta^t)$

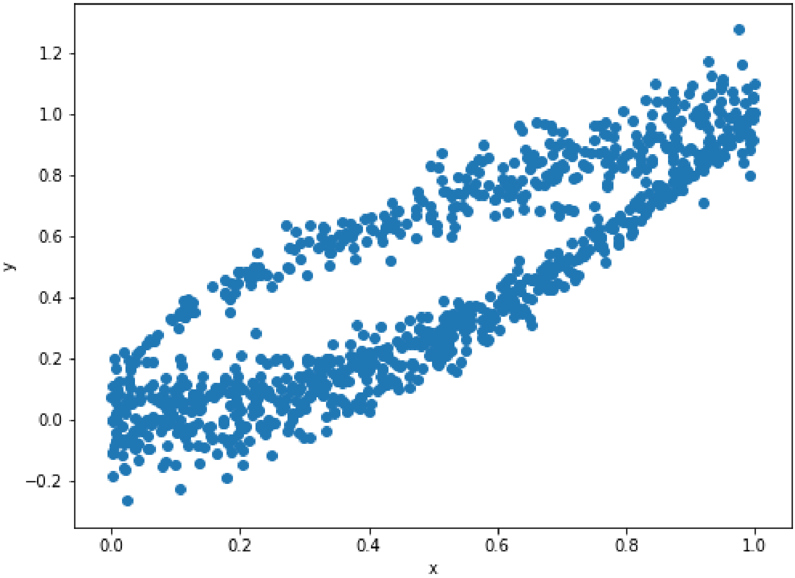
end for

Illustration

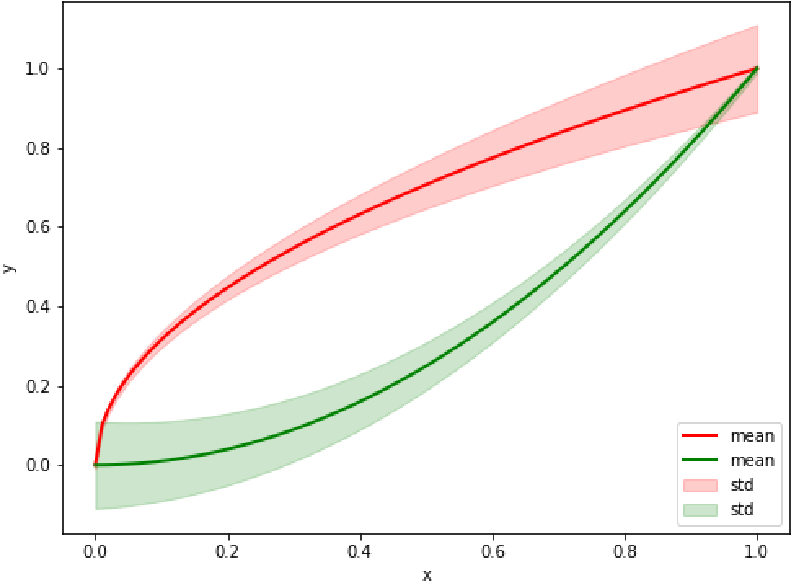


Source: `towardsdatascience.com`

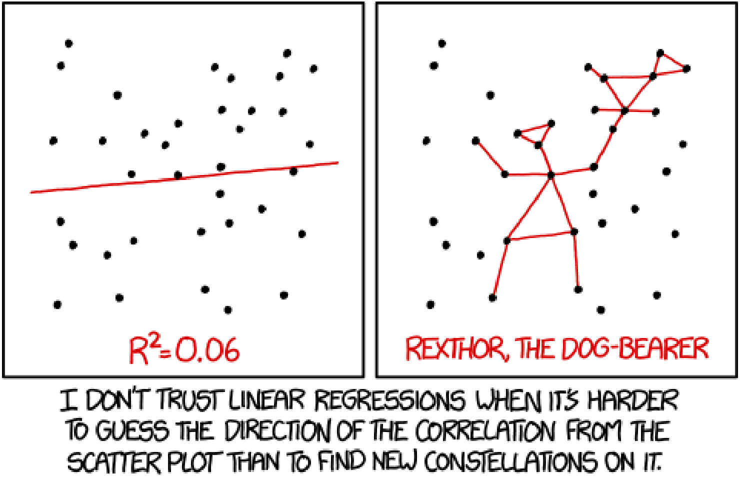
Tricky example



Tricky example



...



Source: `xkcd.com`

My first neural network

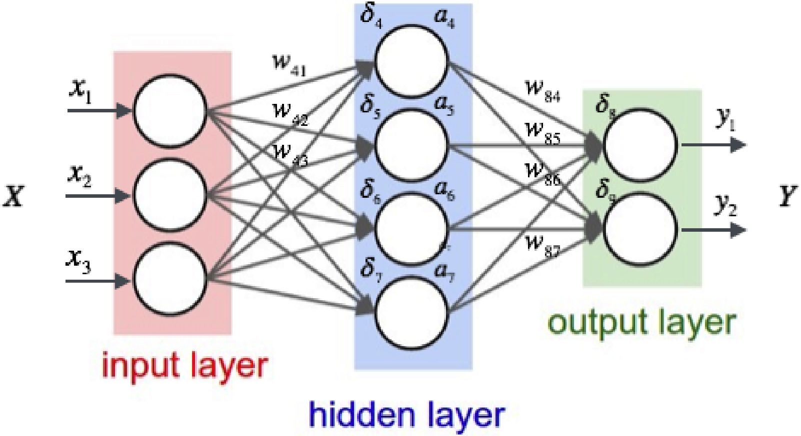
Universal Approximation Theorem

If ϕ is a non-constant, continuous, bounded, monotonic function, then every continuous function f on a compact set from $\mathbb{R}^n$ can be approximated with any precision $\varepsilon > 0$ by:

$$g(x) = c + \sum_{i=1}^{N} \alpha_i \phi(w_i \cdot x + b_i)$$

given large enough N.

Universal Approximators



Stochastic Gradient Descent and Co.

How to train a neural network

Stochastic Gradient Descent and Co.

- how to initialize?
- how to choose an appropriate learning rate?
- how many units?
- which activation function to choose?