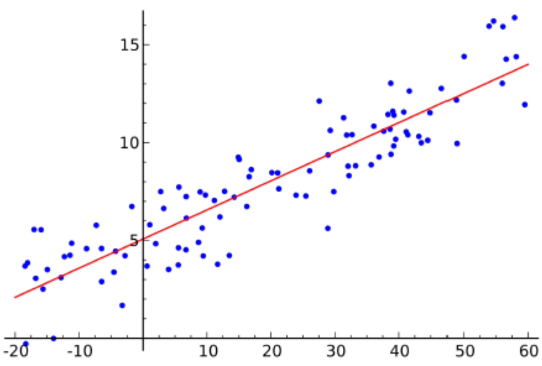
Probabilistic Machine Learning

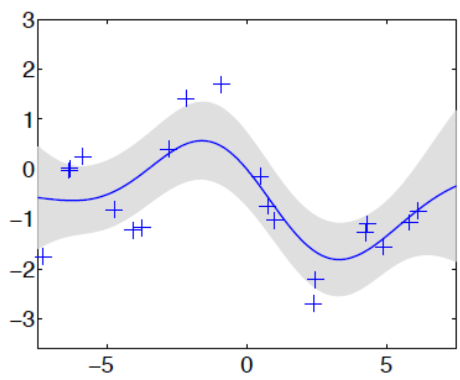
by Prof. Seungchul Lee
iSystems Design Lab
http://isystems.unist.ac.kr/
UNIST

Table of Contents

1. *Probabilistic* Linear Regression

$$P(X \mid \theta) = \text{Probability} \left[\text{data} \mid \text{pattern}\right]$$





- Inference idea

data = underlying pattern + independent noise

- each response generated by a linear model plus some Gaussian noise

$$y = \omega^T x + \varepsilon, \quad \varepsilon \sim \mathcal{N}\left(0, \sigma^2\right)$$

- each response y then becomes a draw from the following Gaussian:

$$y \sim \left(\omega^T x, \sigma^2\right)$$

- Probability of each response variable

$$P(y \mid x, \omega) = \mathcal{N}\left(\omega^T x, \sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}\left(y - \omega^T x\right)^2\right)$$

- Given observed data $D = \{(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)\}$, we want to estimate the weight vector ω

1.1. Maximum Likelihood Solution

- Log-likelihood:

$$\begin{aligned}
\ell(\omega) = \log L(\omega) &= \log P(D \mid \omega) \\
&= \log P(Y \mid X, \omega) \\
&= \log \prod_{n=1}^{m} P\left(y_n \mid x_n, \omega\right) \\
&= \sum_{n=1}^{m} \log P\left(y_n \mid x_n, \omega\right) \\
&= \sum_{n=1}^{m} \log \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\left(y_n - \omega^T x_n\right)^2}{2\sigma^2}\right) \\
&= \sum_{n=1}^{m} \left\{-\frac{1}{2}\log\left(2\pi\sigma^2\right) - \frac{\left(y_n - \omega^T x_n\right)^2}{2\sigma^2}\right\}
\end{aligned}$$

- Maximum Likelihood Solution:

$$\begin{aligned}
\hat{\omega}_{MLE} &= \arg\max_{\omega} \log P(D \mid \omega) \\
&= \arg\max_{\omega} \; -\frac{1}{2\sigma^2} \sum_{n=1}^{m} \left(y_n - \omega^T x_n\right)^2 \\
&= \arg\min_{\omega} \frac{1}{2\sigma^2} \sum_{n=1}^{m} \left(y_n - \omega^T x_n\right)^2 \\
&= \arg\min_{\omega} \sum_{n=1}^{m} \left(y_n - \omega^T x_n\right)^2
\end{aligned}$$

- It is equivalent to the least-squares objective for linear regression

1.2. Maximum-a-Posteriori Solution

- Let's assume a Gaussian prior distribution over the weight vector ω

$$P(\omega) \sim \mathcal{N}\left(\omega \mid 0, \lambda^{-1} I\right) = \frac{1}{(2\pi)^{D/2}} \exp\left(-\frac{\lambda}{2} \omega^T \omega\right)$$

- Log posterior probability:

$$\log P(\omega \mid D) = \log \frac{P(\omega) P(D \mid \omega)}{P(D)} = \log P(\omega) + \log P(D \mid \omega) - \underbrace{\log P(D)}_{\text{constant}}$$

- Maximum-a-Posteriori Solution:

$$\begin{aligned}
&\hat{\omega}_{MAP} \\
&= \arg\max_{\omega} \log P(\omega \mid D) \\
&= \arg\max_{\omega} \{\log P(\omega) + \log P(D \mid \omega)\} \\
&= \arg\max_{\omega} \left\{-\frac{D}{2} \log(2\pi) - \frac{\lambda}{2} \omega^T \omega + \sum_{n=1}^{m} \left\{-\frac{1}{2} \log\left(2\pi\sigma^2\right) - \frac{\left(y_n - \omega^T x_n\right)^2}{2\sigma^2}\right\}\right\} \\
&= \arg\min_{\omega} \frac{1}{2\sigma^2} \sum_{n=1}^{m} \left(y_n - \omega^T x_n\right)^2 + \frac{\lambda}{2} \omega^T \omega \\
&\text{(ignoring constants and changing max to min)}
\end{aligned}$$

- For $\sigma = 1$ (or some constant) for each input, it’s equivalent to the regularized least-squares objective

- BIG Lesson: MAP $= l_2$ norm regularization

1.3. Summary: MLE vs MAP

- MLE solution:

$$\hat{\omega}_{MLE} = \arg\min_{\omega} \frac{1}{2\sigma^2} \sum_{n=1}^{m} \left(y_n - \omega^T x_n\right)^2$$

- MAP solution:

$$\hat{\omega}_{MLE} = \arg\min_{\omega} \frac{1}{2\sigma^2} \sum_{n=1}^{m} \left(y_n - \omega^T x_n\right)^2 + \frac{\lambda}{2}\omega^T \omega$$

- Take-Home messages:
  - MLE estimation of a parameter leads to unregularized solutions
  - MAP estimation of a parameter leads to regularized solutions
  - The prior distribution acts as a regularizer in MAP estimation

- Note : for MAP, different prior distributions lead to different regularizers
  - Gaussian prior on ω regularizes the l_2 norm of ω
  - Laplace prior $\exp(-C\|\omega\|_1)$ on ω regularizes the l_1 norm of ω

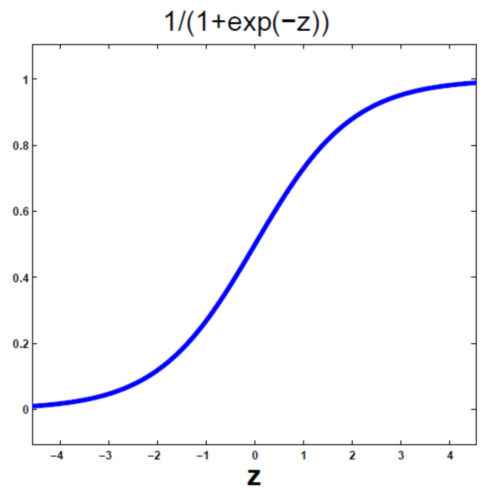
2. *Probabilistic* Linear Classification

- Often we do not just care about predicting the label y for an example
- Rather, we want to predict the label probabilities $P(y \mid x, \omega)$
  - E.g., $P(y = +1 \mid x, \omega)$: the probability that the label is $+1$
  - In a sense, it is our confidence in the predicted label

- Probabilistic classification models allow us do that

- Consider the following function in a compact expression $(y = -1/+1)$:

$$P(y \mid x, \omega) = \sigma\left(y\omega^T x\right) = \frac{1}{1 + \exp(-y\omega^T x)}$$



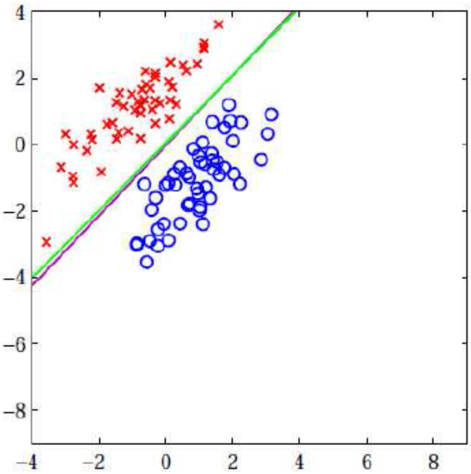
- σ is the logistic function which maps all real number into $(0, 1)$

2.1. Logistic Regression

- What does the decision boundary look like for logistic regression?
- At the decision boundary labels $-1/+1$ becomes equiprobable

$$
\begin{aligned}
P(y=+1 \mid x, \omega) &= P(y=-1 \mid x, \omega) \\
\frac{1}{1+\exp(-\omega^T x)} &= \frac{1}{1+\exp(\omega^T x)} \\
\exp\left(-\omega^T x\right) &= \exp\left(\omega^T x\right) \\
\omega^T x &= 0
\end{aligned}
$$

- The decision boundary is therefore linear $\implies$ logistic regression is a linear classifier



- note: it is possible to kernelize and make it nonlinear

2.2. Maximum Likelihood Solution

- Goal: want to estimate ω from the data $D = \{(x_1, y_1), \cdots, (x_m, y_m)\}$
- Log-likelihood:

$$
\begin{aligned}
\ell(\omega) = \log L(\omega) &= \log P(D \mid \omega) \\
&= \log P(Y \mid X, \omega) \\
&= \log \prod_{n=1}^{m} P(y_n \mid x_n, \omega) \\
&= \sum_{n=1}^{m} \log P(y_n \mid x_n, \omega) \\
&= \sum_{n=1}^{m} \log \frac{1}{1+\exp(-y_n \omega^T x_n)} \\
&= \sum_{n=1}^{m} -\log\left[1+\exp\left(-y_n \omega^T x_n\right)\right]
\end{aligned}
$$

- Maximum Likelihood Solution:

$$\hat{\omega}_{MLE} = \arg\max_{\omega} \log L(\omega) = \arg\min_{\omega} \sum_{n=1}^{m} \log\left[1 + \exp\left(-y_n \omega^T x_n\right)\right]$$

- No closed-form solution exists but we can do gradient descent on ω

$$\begin{aligned}\nabla_{\omega} \log L(\omega) &= \sum_{n=1}^{m} -\frac{1}{1+\exp(-y_n \omega^T x_n)} \exp\left(-y_n \omega^T x_n\right)\left(-y_n x_n\right) \\ &= \sum_{n=1}^{m} \frac{1}{1+\exp\left(y_n \omega^T x_n\right)} y_n x_n\end{aligned}$$

2.3. Maximum-a-Posteriori Solution

- Let's assume a Gaussian prior distribution over the weight vector ω

$$P(\omega) = \mathcal{N}\left(\omega \mid 0, \lambda^{-1} I\right) = \frac{1}{(2\pi)^{D/2}} \exp\left(-\frac{\lambda}{2} \omega^T \omega\right)$$

- Maximum-a-Posteriori Solution:

$$\begin{aligned}&\hat{\omega}_{MAP} \\ &= \arg\max_{\omega} \log P(\omega \mid D) \\ &= \arg\max_{\omega} \{\log P(\omega) + \log P(D \mid \omega) - \underbrace{\log P(D)}_{\text{constant}}\} \\ &= \arg\max_{\omega} \{\log P(\omega) + \log P(D \mid \omega)\} \\ &= \arg\max_{\omega} \left\{ -\frac{D}{2}\log(2\pi) - \frac{\lambda}{2}\omega^T \omega + \sum_{n=1}^{m} -\log\left[1 + \exp\left(-y_n \omega^T x_n\right)\right]\right\} \\ &= \arg\min_{\omega} \sum_{n=1}^{m} \log\left[1 + \exp\left(-y_n \omega^T x_n\right)\right] + \frac{\lambda}{2}\omega^T \omega \\ &\text{(ignoring constants and changing max to min)}\end{aligned}$$

- BIG Lesson: MAP $= l_2$ norm regularization

- No closed-form solution exists but we can do gradient descent on ω

- See "A comparison of numerical optimizers for logistic regression (http://research.microsoft.com/en-us/um/people/minka/papers/logreg/minka-logreg.pdf)" by Tom Minka on optimization techniques (gradient descent and others) for logistic regression (both MLE and MAP)

2.4. Summary: MLE vs MAP

- MLE solution:

$$\hat{\omega}_{MLE} = \arg\min_{\omega} \sum_{n=1}^{m} \log\left[1 + \exp\left(-y\omega^T x_n\right)\right]$$

- MAP solution:

$$\hat{\omega}_{MAP} = \arg\min_{\omega} \sum_{n=1}^{m} \log\left[1 + \exp\left(-y\omega^T x_n\right)\right] + \frac{\lambda}{2}\omega^T \omega$$

- Take-home messages (we already saw these before)
  - MLE estimation of a parameter leads to unregularized solutions
  - MAP estimation of a parameter leads to regularized solutions
  - The prior distribution acts as a regularizer in MAP estimation

- Note: For MAP, different prior distributions lead to different regularizers
  - Gaussian prior on ω regularizes the l_2 norm of ω
  - Laplace prior $\exp\left(-C\lVert\omega\rVert_1\right)$ on ω regularizes the l_1 norm of ω

3. *Probabilistic* Clustering

- will not cover in this course

4. *Probabilistic* Dimension Reduction

- will not cover in this course

In [1]:

```
%%javascript
$.getScript('https://kmahelona.github.io/ipython_notebook_goodies/ipython_notebook_toc.js')
```