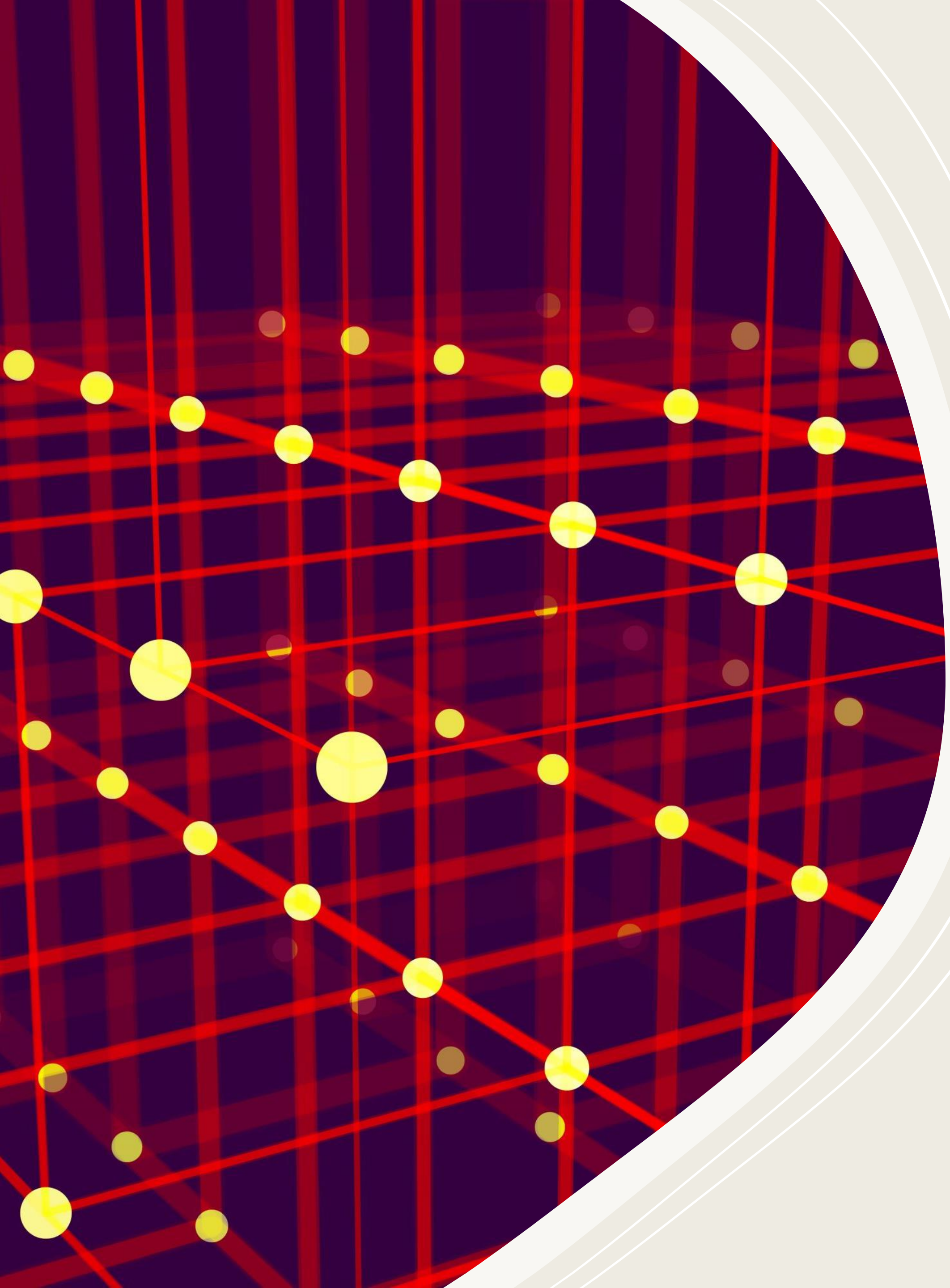


CPSC 340 – Tutorial 7

Lironne Kurzman
lironnek@cs.ubc.ca

Slides courtesy of Nam Hee Kim

University of British Columbia

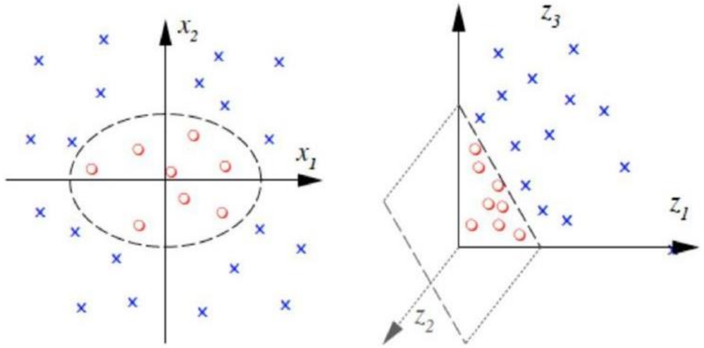
November 1st, 2021

Agenda

1. Kernel Trick
2. MAP/MLE
3. Gradient Descent?

Why Kernel Trick?

- Change of basis is useful.
- We have data that is not linearly structured and we want to use a linear model on this data.
- If we first map our data into a higher-dimensional space, a linear model operating in this space will behave non-linearly in the original input space.



https://math.stackexchange.com/questions/353607/how-do-inner-product-space-determine-half-planes

Why Kernel Trick?

- However, mapping and storing the data could be very costly.
- For instance, with data of dimension (n, d) and a polynomial basis of degree p, there are $O(d^p)$ terms/ features.
- For large d and p, intractable to store.
- Kernel tricks helps efficient use of all such features.

The Kernel "trick"

- Assume L2-regularized least squares objective with basis Z:

$$f(v) = \frac{1}{2}||Zv - Y||^2 + \frac{\lambda}{2}||v||^2$$

- Normal equations to find minimum v:

$$v = (Z^T Z + \lambda I)^{-1} Z^T Y$$

- Other normal equations to find minimum v:

$$v = Z^T (Z Z^T + \lambda I)^{-1} Y$$

The Kernel "trick"

- If we want to make predictions $\hat{y}$ for test data $\tilde{X}$ by forming $\tilde{Z}$ using "other normal" equations:

$$
\begin{aligned}
\hat{y} &= \tilde{Z}v \\
&= \overset{\tilde{K}}{\boxed{\tilde{Z}Z^T}}(\overset{K}{\boxed{ZZ^T}} + \lambda I)^{-1}Y \\
&= \tilde{K}(K + \lambda I)^{-1}Y
\end{aligned}
$$

- Efficiently compute K and $\tilde{K}$ even though forming Z and $\tilde{Z}$ is intractable.

Kernel Function

- Kernel function does not calculate Z explicitly. Returns similarity between transformed points Z_i, Z_j :

$$K(X_i, X_j) = Z_i^T Z_j$$

using only untransformed points X_i, X_j.

Kernel Types

- Linear Kernel:

$$K(X_i, X_j) = Z_i^T Z_j = X_i^T X_j$$

- Degree P polynomial Kernel:

$$K(X_i, X_j) = Z_i^T Z_j = (1 + X_i^T X_j)^P$$

- Gaussian RBF Kernel:

$$K(X_i, X_j) = Z_i^T Z_j = \exp(-\frac{||X_i - X_j||^2}{2\sigma^2})$$

MLE/MAP

MLE/MAP Intuition

- Maximum Likelihood (MLE)

$$\arg\max_{w} p(D|w)$$

  - Find w that maximizes the probability of D given w
- Maximum A Posteriori (MAP)

$$\arg\max_{w} p(w|D) \propto p(D|w)p(w)$$

  - Find w that maximizes its probability given D

MLE/MAP Intuition

- Supervised Learning assumptions:
  1. Dataset is a tuple $D \doteq (X, y)$
  2. y depends on w but X does not
  3. The samples are independent and identically distributed (i.i.d.)

- Negative Log Likelihood (NLL)

$$\arg\max_{\theta}\{f(\theta)\} \equiv \arg\min_{\theta}\{-\log f(\theta)\}$$

MLE/MAP Intuition

- Supervised Learning assumptions:
  1. Dataset is a tuple $D \doteq (X, y)$
  2. y depends on w but X does not
  3. The samples are independent and identically distributed (i.i.d.)

- Negative Log Likelihood (NLL)

$$\arg\max_{\theta}\{f(\theta)\} \equiv \arg\min_{\theta}\{-\log f(\theta)\}$$

Maximum a Posteriori (MAP) Estimation

- MAP estimate definition:

$$\hat{w} = \underset{w}{\operatorname{argmax}}\{P(w|D)\}$$

  - With Bayes rule, we can see that MAP maximizes likelihood times the prior:

$$\hat{w} = \underset{w}{\operatorname{argmax}}\{P(w|D)\} = \underset{w}{\operatorname{argmax}}\{\frac{P(D|w)P(w)}{P(D)}\} \equiv \underset{w}{\operatorname{argmax}}\{P(D|w)P(w)\}$$

  - With IID examples, MAP becomes:

$$\hat{w} = \underset{w}{\operatorname{argmax}}\{P(w|D)\} = \underset{w}{\operatorname{argmax}}\{\prod_{i=1}^{n} P(D_i|w)P(w)\}$$

  - By taking the negative of the logarithm:

$$\hat{w} = \underset{w}{\operatorname{argmin}}\{-\sum_{i=1}^{n}[\log P(D_i|w)] - \log P(w)\}$$

Example

2 MAP Estimation

Rubric: {reasoning:8}

In class, we considered MAP estimation in a regression model where we assumed that:

- The likelihood $p(y_i \mid x_i, w)$ is a normal distribution with a mean of $w^T x_i$ and a variance of 1.
- The prior for each variable j, $p(w_j)$, is a normal distribution with a mean of zero and a variance of λ^{-1}.

Under these assumptions, we showed that this leads to the standard L2-regularized least squares objective function,

$$f(w) = \frac{1}{2}\|Xw - y\|^2 + \frac{\lambda}{2}\|w\|^2,$$

Example

- The normal distribution notation is $\mathcal{N}(\mu, \sigma^2)$
- Given $y_i | x_i, w \sim \mathcal{N}(w^T x_i, 1)$, which means:

$$p(y_i | x_i, w) = \frac{1}{\sqrt{2 \cdot 1 \cdot \pi}} \exp(-\frac{(w^T x_i - y_i)^2}{2 \cdot 1})$$

- Therefore, maximizing $p(y | X, w)$ w.r.t w is equivalent to minimizing the unregularized least squares problem.

Example

- Given

$$p(y|X,w) = \prod_{i=1}^{N} \frac{1}{\sqrt{2 \cdot 1 \cdot \pi}} \exp\left(-\frac{(w^T x_i - y_i)^2}{2 \cdot 1}\right)$$

$$\begin{aligned}
\arg\max_w p(y|X,w) &= \arg\max_w \prod_{i=1}^{N} \frac{1}{\sqrt{2 \cdot 1 \cdot \pi}} \exp\left(-\frac{(w^T x_i - y_i)^2}{2 \cdot 1}\right) \\
&= \arg\min_w \Bigg\{ - N \log(\frac{1}{\sqrt{2\pi}}) \\
&\quad - \sum_{i=1}^{N} \log\left(\exp\left(-\frac{(w^T x_i - y_i)^2}{2}\right)\right) \Bigg\} \quad \text{(log is monotonic)} \\
&= \arg\min_w \sum_{i=1}^{N} \frac{(w^T x_i - y_i)^2}{2} \quad \text{(ignoring the constant term)} \\
&= \arg\min_w \frac{1}{2} \cdot ||Xw - y||_2^2 \quad \text{(negate both sides)} \\
&= \arg\min_w ||Xw - y||_2^2 \quad \text{(does not change solution)}
\end{aligned}$$

Example

$$p(w|D) = \frac{p(D|w)p(w)}{p(D)} \propto p(D|w)p(w)$$

The rest is the same as MLE with the addition of $p(w)$:

$$\begin{aligned} p(D|w)p(w) &= p(X, y|w)p(w) \\ &= p(X|w)p(y|X, w)p(w) \\ &= p(X)p(y|X, w)p(w) \\ &\propto p(y|X, w)p(w) \end{aligned}$$

Therefore we get:

$$\begin{aligned} \arg\max_w p(w|D) &\equiv \arg\max_w p(y|X, w)p(w) \\ &\equiv \arg\min_w \{-\big(\log p(y|X, w) + \log p(w)\big)\} \\ &\equiv \arg\min_w \{-\big(\sum_{i=1}^{N} \log p(y_i|x_i, w) + \log p(w)\big)\} \end{aligned}$$

Example

$$p(w|D) = \frac{p(D|w)p(w)}{p(D)} \propto p(D|w)p(w)$$

The rest is the same as MLE with the addition of $p(w)$:

$$\begin{aligned} p(D|w)p(w) &= p(X, y|w)p(w) \\ &= p(X|w)p(y|X, w)p(w) \\ &= p(X)p(y|X, w)p(w) \\ &\propto p(y|X, w)p(w) \end{aligned}$$

Therefore we get:

$$\begin{aligned} \arg\max_w p(w|D) &\equiv \arg\max_w p(y|X, w)p(w) \\ &\equiv \arg\min_w \{-(\log p(y|X, w) + \log p(w))\} \\ &\equiv \arg\min_w \{-(\sum_{i=1}^{N} \log p(y_i|x_i, w) + \log p(w))\} \end{aligned}$$

Example

- Same definition of y as before:

$$y_i | x_i, w \sim \mathcal{N}(w^T x_i, 1)$$

- With the addition of a prior:

$$w_j \sim \mathcal{N}(0, \lambda^{-1})$$

Example

$$p(y|X,w) = \prod_{i=1}^{N} \frac{1}{\sqrt{2 \cdot 1 \cdot \pi}} \exp(-\frac{(w^T x_i - y_i)^2}{2 \cdot 1}) \quad y_i|x_i, w \sim N(w^T x_i, 1)$$

$$p(w) = \prod_{j=1}^{d} \frac{1}{\sqrt{2 \cdot \lambda^{-1} \cdot \pi}} \exp(-\frac{(w_j - 0)^2}{2 \cdot \lambda^{-1}}) \quad w_j \sim N(0, \lambda^{-1})$$

$$\begin{aligned}
\arg\max_w p(w|X,y) &= \arg\max_w p(y|X,w) \cdot p(w) \\
&= \arg\max_w \log(p(y|X,w)) + \log\left(\prod_{j=1}^{d} \frac{1}{\sqrt{2 \cdot \lambda^{-1} \cdot \pi}} \exp\left(-\frac{(w_j - 0)^2}{2 \cdot \lambda^{-1}}\right)\right) \\
&= \arg\max_w \log(p(y|X,w)) + \sum_{i=1}^{d} \log\left(\exp\left(-\frac{\lambda}{2} w_j^2\right)\right) \\
&= \arg\min_w -\log(p(y|X,w)) + \sum_{i=1}^{d} \frac{\lambda}{2} w_j^2 \quad \text{(negate both sides)} \\
&= \arg\min_w \frac{1}{2}||Xw - y||^2 + \frac{\lambda}{2}||w||^2
\end{aligned}$$

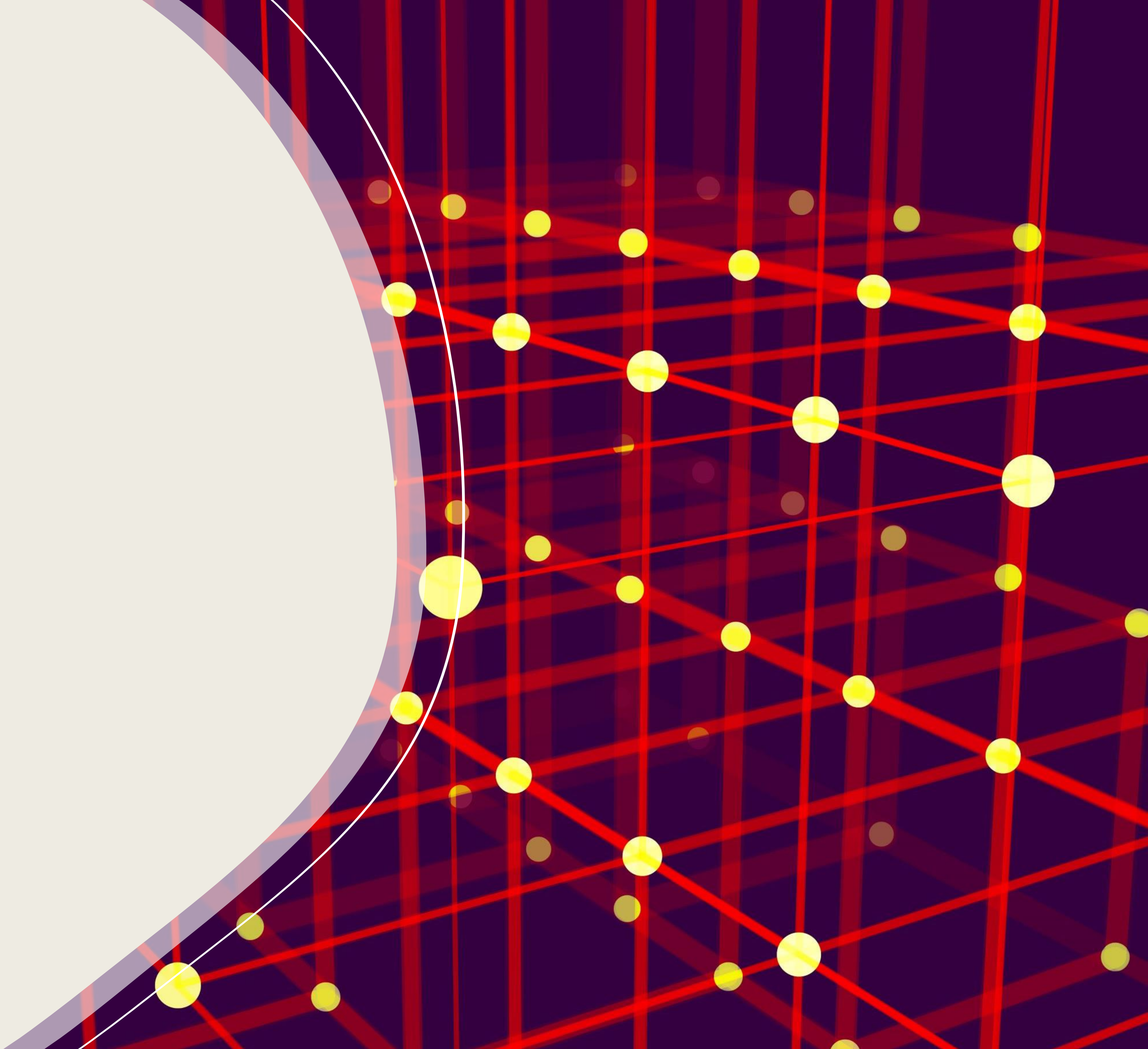
Example

$$p(y|X,w) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\cdot 1\cdot \pi}} \exp(-\frac{(w^T x_i - y_i)^2}{2\cdot 1}) \quad y_i|x_i, w \sim N(w^T x_i, 1)$$

$$p(w) = \prod_{j=1}^{d} \frac{1}{\sqrt{2\cdot \lambda^{-1}\cdot \pi}} \exp(-\frac{(w_j - 0)^2}{2\cdot \lambda^{-1}}) \quad w_j \sim N(0, \lambda^{-1})$$

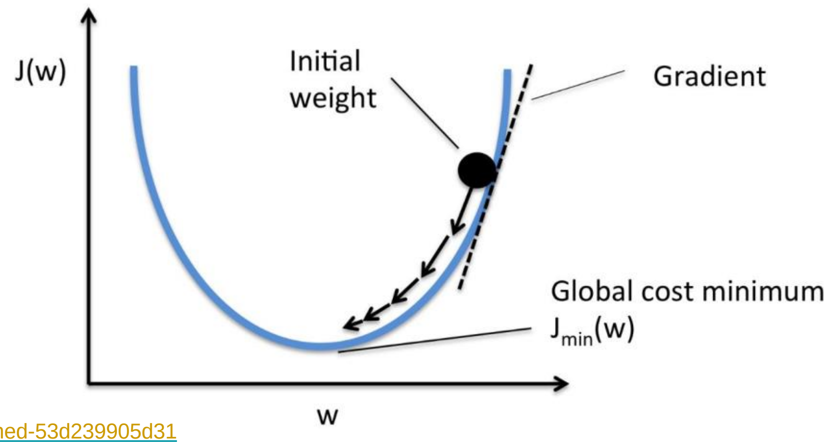
$$\begin{aligned}
\arg\max_w p(w|X,y) &= \arg\max_w p(y|X,w)\cdot p(w) \\
&= \arg\max_w \log(p(y|X,w)) + \log\left(\prod_{j=1}^{d} \frac{1}{\sqrt{2\cdot\lambda^{-1}\cdot\pi}} \exp\left(-\frac{(w_j-0)^2}{2\cdot\lambda^{-1}}\right)\right) \\
&= \arg\max_w \log(p(y|X,w)) + \sum_{i=1}^{d} \log\left(\exp\left(-\frac{\lambda}{2}w_j^2\right)\right) \\
&= \arg\min_w -\log(p(y|X,w)) + \sum_{i=1}^{d} \frac{\lambda}{2} w_j^2 \quad \text{(negate both sides)} \\
&= \arg\min_w \frac{1}{2}||Xw - y||^2 + \frac{\lambda}{2}||w||^2
\end{aligned}$$

MLE/MAP



Gradient Descent

"Gradient descent is an iterative algorithm, that starts from a random point on a function and travels down its slope in steps until it reaches the lowest point of that function."



Aishwarya V Srinivasan : https://towardsdatascience.com/stochastic-gradient-descent-clearly-explained-53d239905d31
Sarthak Gupta: https://hackernoon.com/dl03-gradient-descent-719aff91c7d6

Gradient Descent

The steps of the algorithm are

1. Find the slope of the objective function **with respect to each parameter/feature**. In other words, compute the gradient of the function.

$$\nabla_w f(w) = [\frac{\partial f(w)}{\partial w_1}, \frac{\partial f(w)}{\partial w_2}, \ldots, \frac{\partial f(w)}{\partial w_d}]^T$$

Aishwarya V Srinivasan : https://towardsdatascience.com/stochastic-gradient-descent-clearly-explained-53d239905d31
Sarthak Gupta: https://hackernoon.com/dl03-gradient-descent-719aff91c7d6

Gradient Descent

2. Pick a random initial value for the parameters.

$$w^{0}_{init} = [w^{0}_{1}, w^{0}_{2}, \dots, w^{0}_{d}]^T$$

3. Update the gradient function by plugging in the parameter values.

$$\nabla_w f(w_{init}) = [\frac{\partial f(w_{init})}{\partial w_1}, \frac{\partial f(w_{init})}{\partial w_2}, \dots, \frac{\partial f(w_{init})}{\partial w_d}]^T$$

Aishwarya V Srinivasan : https://towardsdatascience.com/stochastic-gradient-descent-clearly-explained-53d239905d31
Sarthak Gupta: https://hackernoon.com/dl03-gradient-descent-719aff91c7d6

Gradient Descent

4. Calculate the step sizes for each feature as : **step size = gradient * learning rate.**

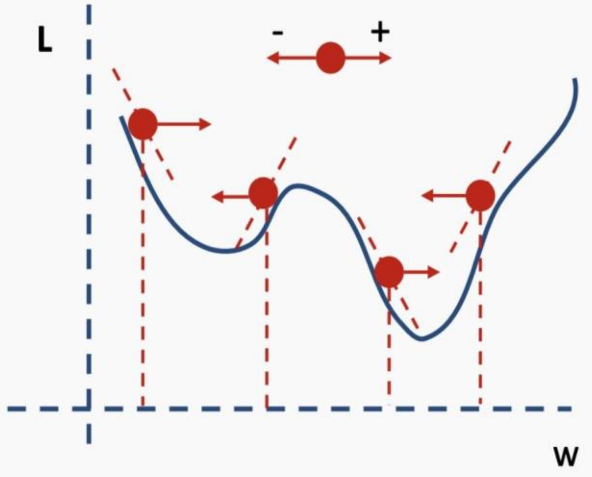
5. Calculate the new parameters as : **new params = old params -step size**

6. Repeat steps 3 to 5 until gradient is almost 0.

Aishwarya V Srinivasan : https://towardsdatascience.com/stochastic-gradient-descent-clearly-explained-53d239905d31
Sarthak Gupta: https://hackernoon.com/dl03-gradient-descent-719aff91c7d6

Gradient Descent

$$w^{(i+1)} = w^{(i)} - \lambda \frac{d\mathcal{L}}{dw}$$



Aishwarya V Srinivasan : https://towardsdatascience.com/stochastic-gradient-descent-clearly-explained-53d239905d31
Sarthak Gupta: https://hackernoon.com/dl03-gradient-descent-719aff91c7d6