

Logistic Regression

Concepts:

Probabilistic predictions and decision theory

Maximum likelihood estimation application to LR

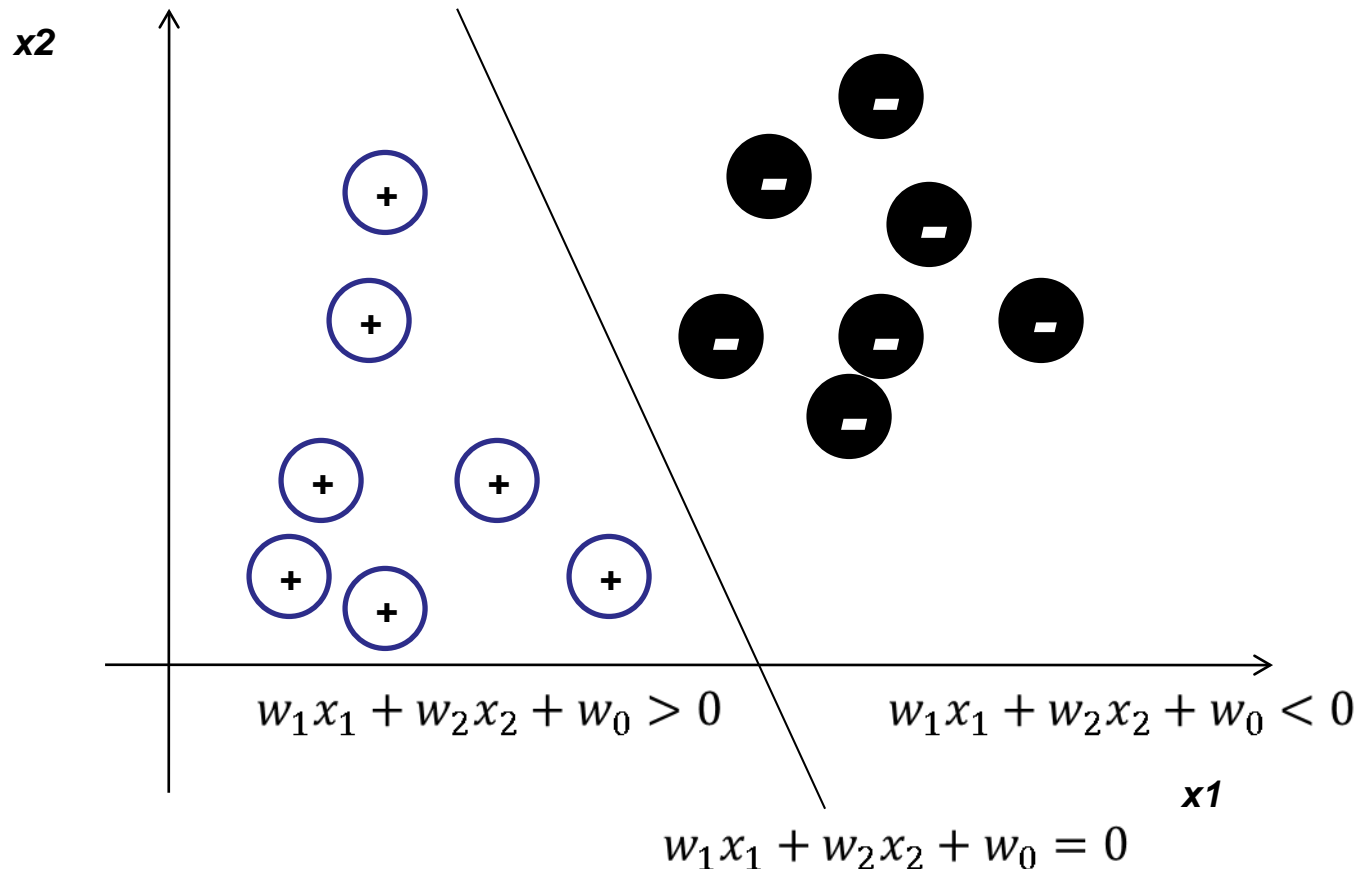
Maximum a posterior (MAP) estimation and connection to regularization

Classification problem

- Given input x , the goal is to predict y , which is a categorical variable
 - x : the feature vector
 - y : the class label
- Example:
 - x : monthly income and bank saving amount;
 y : risky or not risky
 - x : review text for a product
 y : sentiment positive, negative or neutral

Binary Linear Classifier

- We will begin with the simplest choice: linear classifiers for binary classification problems ($y \in \{0,1\}$)

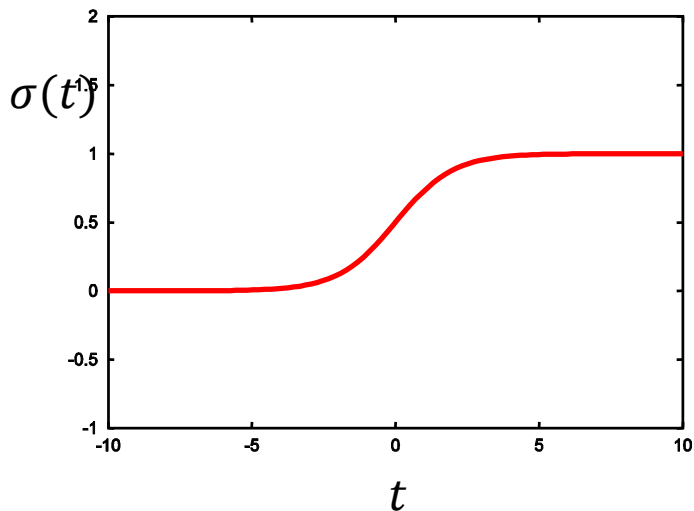


Logistic Regression

- Input $\mathbf{x} = [1, x_1, x_2, \dots, x_d]^T$, target output $y \in \{0, 1\}$
- Logistic regression is a probabilistic classifier:

$$P(y = 1 | \mathbf{x}; \mathbf{w}) = \sigma(\mathbf{w}^T \mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{w}^T \mathbf{x})}$$

$$P(y = 0 | \mathbf{x}; \mathbf{w}) = 1 - \sigma(\mathbf{w}^T \mathbf{x})$$



- linear function $\mathbf{w}^T \mathbf{x}$ has range $(-\infty, \infty)$
- Sigmoid function σ warps the value of $\mathbf{w}^T \mathbf{x}$ to a value between 0 and 1

Logistic regression: linear classifier

- Maximum A Posteriori (MAP) prediction of y :

$$y_{map} = \arg \max_{v \in \{0,1\}} P(y = v | \mathbf{x}; \mathbf{w})$$

- We will predict $y = 1$ if

$$P(y = 1 | \mathbf{x}; \mathbf{w}) \geq P(y = 0 | \mathbf{x}; \mathbf{w}) \Rightarrow$$

$$\frac{1}{1 + \exp(-\mathbf{w}^T \mathbf{x})} \geq \frac{\exp(-\mathbf{w}^T \mathbf{x})}{1 + \exp(-\mathbf{w}^T \mathbf{x})} \Rightarrow$$

$$1 \geq \exp(-\mathbf{w}^T \mathbf{x}) \Rightarrow$$

$$0 \geq -\mathbf{w}^T \mathbf{x} \Rightarrow \mathbf{w}^T \mathbf{x} \geq 0$$

- We refer to $\mathbf{w}^T \mathbf{x} = 0$ as our decision boundary

A more general decision rule

If we have some knowledge about the cost of different types of mistakes, given $P(y|\mathbf{x})$, we can choose the prediction that ***minimizes the expected cost***:

$$y^* = \arg \min_y \sum_{y'} L(y, y') P(y'|\mathbf{x})$$

True label → Predicted ↓	Spam	Non-spam
Spam	0	10
Non-spam	1	0

For example: $P(y = \text{spam}|\mathbf{x}) = 0.6$

- The expected cost if predict spam?
- What if we predict non-spam?
- Which prediction minimizes the expected cost?

With this more general decision rule, it can be shown that Logistic regression still leads to a linear classifier, just with a different threshold 6

Learning for Logistic Regression

- Given a set of training examples:
- We assume examples are identically, independently distributed (I.I.D.) following:

$$P(y = 1|\mathbf{x}; \mathbf{w}) = \frac{1}{1 + \exp(-\mathbf{w}^T \mathbf{x})}$$

- Learn \mathbf{w} from the training data using Maximum Likelihood Estimation

Maximum (conditional) Likelihood Estimation

- Data log-likelihood:

$$\begin{aligned}\log \prod_i P(\mathbf{x}^i, y^i; \mathbf{w}) &= \sum_i \log P(\mathbf{x}^i, y^i; \mathbf{w}) \\ &= \sum_i \log P(y^i | \mathbf{x}^i; \mathbf{w}) P(\mathbf{x}^i; \mathbf{w}) \\ &= \sum_i \log P(y^i | \mathbf{x}^i; \mathbf{w}) + C\end{aligned}$$

- Maximum (conditional) likelihood estimation of \mathbf{w} :

$$\mathbf{w}_{MLE} = \operatorname{argmax}_{\mathbf{w}} \sum_i \log P(y^i | \mathbf{x}^i; \mathbf{w})$$

Computing Log-likelihood

$$\begin{aligned}l(\mathbf{w}) &= \sum_i \log P(y^i | \mathbf{x}^i; \mathbf{w}) \\&= \sum_{y^i=1} \log P(y = 1 | \mathbf{x}^i; \mathbf{w}) + \sum_{y^i=0} \log P(y = 0 | \mathbf{x}^i; \mathbf{w}) \\&= \sum_i y^i \log P(y = 1 | \mathbf{x}^i; \mathbf{w}) + (1 - y^i) \log P(y = 0 | \mathbf{x}^i; \mathbf{w}) \\&= \sum_i y^i \log \frac{P(y = 1 | \mathbf{x}^i; \mathbf{w})}{P(y = 0 | \mathbf{x}^i; \mathbf{w})} + \log P(y = 0 | \mathbf{x}^i; \mathbf{w}) \\&= \sum_i y^i \mathbf{w}^T \mathbf{x}^i + \log \left(\frac{1}{1 + \exp(\mathbf{w}^T \mathbf{x}^i)} \right) \\&= \sum_i y^i \mathbf{w}^T \mathbf{x}^i - \log(1 + \exp(\mathbf{w}^T \mathbf{x}^i))\end{aligned}$$

Gradient

$$l(\mathbf{w}) = \sum_i [y^i \mathbf{w}^T \mathbf{x}^i - \log(1 + \exp(\mathbf{w}^T \mathbf{x}^i))]$$

$$\nabla l = \sum_i \left[y^i \mathbf{x}^i - \frac{\exp(\mathbf{w}^T \mathbf{x}^i) \mathbf{x}^i}{1 + \exp(\mathbf{w}^T \mathbf{x}^i)} \right]$$

$$= \sum_i \left[y^i \mathbf{x}^i - \frac{\mathbf{x}^i}{1 + \exp(-\mathbf{w}^T \mathbf{x}^i)} \right]$$

$$= \sum_i [y^i - P(y = 1 | \mathbf{x}^i; \mathbf{w})] \mathbf{x}^i$$

Batch Gradient Ascent for LR

Given : training examples (\mathbf{x}^i, y^i) , $i = 1, \dots, N$

Let $\mathbf{w} \leftarrow \mathbf{w}_0$ // e.g., $(0,0,0,\dots,0)$

Repeat until convergence

$\mathbf{d} \leftarrow (0,0,0,\dots,0)$

For $i = 1$ to N do

$$\hat{y}^i \leftarrow \frac{1}{1 + e^{-\mathbf{w}^T \mathbf{x}^i}}$$

$$error = y^i - \hat{y}^i$$

$$\mathbf{d} = \mathbf{d} + error \cdot \mathbf{x}^i$$

$$\mathbf{w} \leftarrow \mathbf{w} + \eta \mathbf{d}$$

Stochastic Gradient Ascent for LR

Given : training examples (\mathbf{x}^i, y^i) , $i = 1, \dots, N$

Let $\mathbf{w} \leftarrow \mathbf{w}_0$ // e.g., $(0,0,0,\dots,0)$

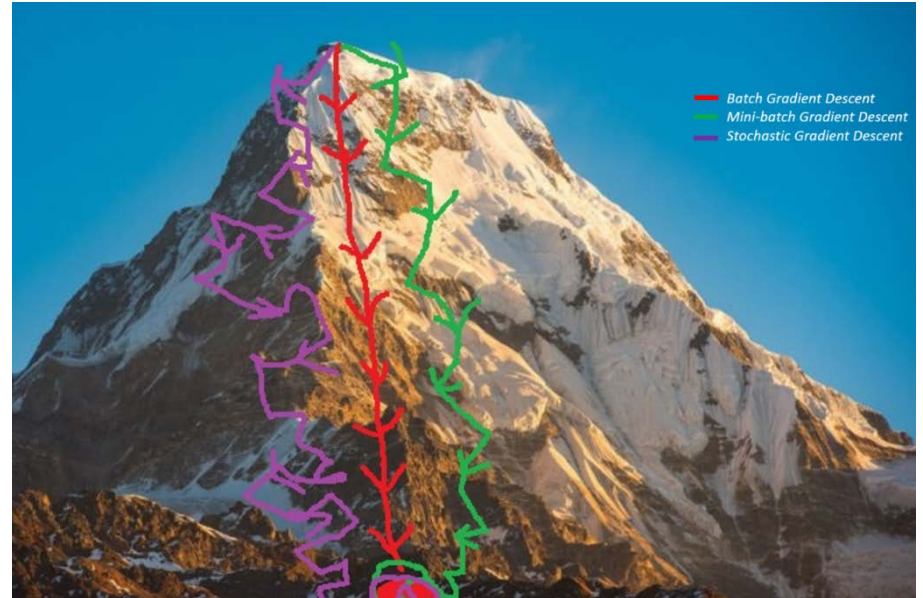
Repeat until convergence

Randomly shuffle examples

For $i = 1$ to N do

$$\hat{y}^i \leftarrow \frac{1}{1 + e^{-\mathbf{w}^T \mathbf{x}^i}}$$

$$\mathbf{w} \leftarrow \mathbf{w} + \eta (y^i - \hat{y}^i) \mathbf{x}^i$$



- Stochastic gradient ascent performs updates for each example
- Learning shows more fluctuations than batch gradient ascent
- Shuffling the examples in each round (epoch) helps to make it more robust

Picking learning rate

- Use grid search in log-space over small values
 - 0.01, 0.001, 0.0001, ...
 - Plot the objective to gauge convergence
 - If no global optimum, use separate tuning/validation set to select learning rate
- Sometimes, employ a schedule to gradually reduce the learning rate
 - $\frac{1}{1+kt}$ where k is a hyperparameter and t is the iteration number,
 - $\frac{1}{t^2}$
- More advanced techniques
 - Adaptive gradient that uses different rate for different dimensions (ADAM, AdaGrad ...)

Logistic regression overfits

- Consider the gradient:

$$\sum_i [y^i - P(y = 1 | \mathbf{x}^i; \mathbf{w})] \mathbf{x}^i$$

If we have a binary feature x_p that only takes value 1 for positive examples, i.e., this feature perfectly classifies the examples

What will happen to $\frac{\partial l}{\partial x_p}$? Will it ever be zero?

What will happen to w_p ?

In general, when data is linearly separable, LR overfits

Soft-max Logistic Regression for $k > 2$ classes

- For $k > 2$ classes, we can define the posterior probability using the soft-max function

$$p(y = k|\mathbf{x}) = \hat{y}_k = \frac{\exp(\mathbf{w}_k^T \mathbf{x})}{\sum_{j=1}^K \exp(\mathbf{w}_j^T \mathbf{x})}$$

- Going through the same MLE derivations, we arrive at the following gradient:

$$\nabla_{\mathbf{w}_k} L = \sum_{i=1}^N (y_k^i - \hat{y}_k^i) \mathbf{x}^i$$

where $y_k^i = 1$ if $y^i = k$, and 0 otherwise for $k = 1, \dots, K$

- So far we have introduced MLE for logistic regression
- We will now introduce another paradigm for estimating model parameters
 - The Bayesian paradigm

Bayesian vs. Frequentist

- Two different views for parameter estimation
- Frequentist: a parameter is a deterministic unknown value
- Bayesian: a parameter is a random variable with a distribution
 - Use priors to express our belief/preference about the parameter before observing any data
 - After observing the data, update our belief by computing the posterior distribution of the parameter

$$p(\theta|D) = \frac{p(\theta)p(D|\theta)}{p(D)} = \frac{p(\theta)p(D|\theta)}{\int p(D|\theta)p(\theta)d\theta}$$

Posterior distribution of θ

Prior distribution of θ

Maximum A Posteriori (MAP) estimation as a penalty method

$$\begin{aligned}\hat{\theta}_{MAP} &= \operatorname{argmax}_{\theta} p(\theta|D) \\ &= \operatorname{argmax}_{\theta} \frac{p(D|\theta)p(\theta)}{p(D)} \\ &= \operatorname{argmax}_{\theta} p(D|\theta)p(\theta) \\ &= \operatorname{argmax}_{\theta} \log p(D|\theta) + \log p(\theta)\end{aligned}$$



***Penalty term
/regularization***

MAP for Logistic Regression

$$\begin{aligned}\operatorname{argmax}_{\mathbf{w}} P(\mathbf{w}|\mathbf{D}) &= \operatorname{argmax}_{\mathbf{w}} P(\mathbf{D}|\mathbf{w}) P(\mathbf{w}) \\ &= \operatorname{argmax}_{\mathbf{w}} \log P(\mathbf{D}|\mathbf{w}) + \log P(\mathbf{w})\end{aligned}$$

- $\log P(\mathbf{D}|\mathbf{w})$: the log-likelihood of \mathbf{w}

$$\sum_i \log P(y^i | \mathbf{x}^i, \mathbf{w})$$

- $P(\mathbf{w})$: a prior distribution, e.g.

$$\mathbf{w} \sim N(0, \sigma^2 \mathbf{I})$$

Large weights correspond to more complex models, this prior prefer simpler hypothesis (zero mean)

Logistic Regression: MAP

$$\begin{aligned} & \underset{\mathbf{w}}{\operatorname{argmax}} \log P(\mathbf{D}|\mathbf{w}) + \log P(\mathbf{w}) \\ &= \underset{\mathbf{w}}{\operatorname{argmax}} l(\mathbf{w}) + \log N(\mathbf{w}; 0, \sigma^2 \mathbf{I}) \\ &= \underset{\mathbf{w}}{\operatorname{argmax}} l(\mathbf{w}) + \sum_j \log\left(\frac{1}{\sqrt{2\pi}\sigma} \exp \frac{-w_j^2}{2\sigma^2}\right) \\ &= \underset{\mathbf{w}}{\operatorname{argmax}} l(\mathbf{w}) + \sum_j \frac{-w_j^2}{2\sigma^2} \\ &= \underset{\mathbf{w}}{\operatorname{argmax}} l(\mathbf{w}) - \underbrace{\frac{\lambda}{2} \sum_j w_j^2}_{\text{Regularization}} \end{aligned}$$

$$\lambda = \frac{1}{\sigma^2}$$

Old delta:

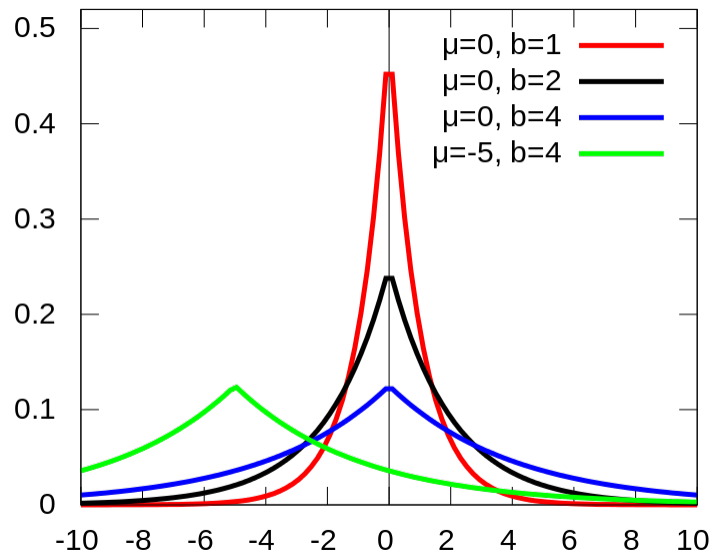
$$\nabla L(\mathbf{w}) = \sum_{i=1}^N (y^i - \hat{y}^i) \mathbf{x}^i$$



$$\nabla L(\mathbf{w}) = \sum_{i=1}^N (y^i - \hat{y}^i) \mathbf{x}^i - \lambda \mathbf{w}$$

Impact of prior

- Instead of using Gaussian prior, one can also consider other priors
- Laplace prior: $w_i \sim \text{Laplace}(0, b)$

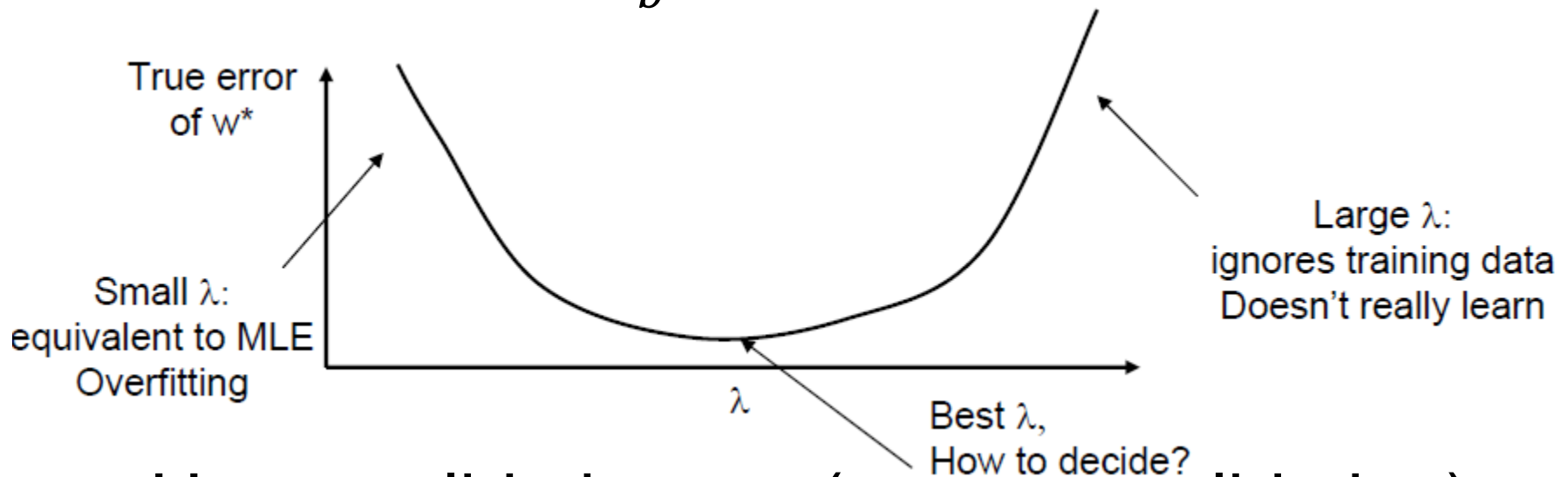


$$p(w_i) = \frac{1}{2b} \exp - \frac{|w_i|}{b}$$

- Lead to L1 regularization: $-\frac{1}{b} \sum_i |w_i|$

Impact of λ

- λ is inversely proportional to the variance of our prior belief
 - Gaussian prior: $\lambda = \frac{1}{\sigma^2}$
 - Laplace prior: $\frac{1}{b}$



- Use a validation set (or cross-validation) to choose λ

Summary of Logistic Regression

- A popular discriminative classifier
- Learns conditional probability distribution $P(y | \mathbf{x})$
 - Defined by a logistic function
 - Produces a **linear** decision boundary
 - Nonlinear classifier by using basis functions
- Maximum likelihood estimation (MLE)
 - Gradient ascent bears interesting similarity with perceptron
 - Overfits for linearly separable case, regularization can help
 - Multi-class logistic regression: use the soft-max function
- Maximum posterior estimation (MAP)
 - Gaussian prior on the weights = L_2 regularization
 - Laplace prior = L_1 regularization
 - Overfitting controlled by the variance on the prior