Lecture 5
Maximum Likelihood and *Maximum-a-Posteriori* Estimate
&
Intro to Supervised Classification

Bayes' rule for evaluating plausibility of a scientific hypothesis, given data.

$$\boxed{P\left(H_i \mid D, I\right) = \frac{P\left(H_i \mid I\right) P\left(D \mid H_i, I\right)}{P(D \mid I)}}$$

H_i = proposition asserting the truth of a hypothesis of interest

I = proposition representing our prior information

D = proposition representing data

$P\left(D \mid H_i, I\right)$ = probability of obtaining data D; if H_i and I are true (Likelihood function)

$P\left(H_i \mid I\right)$ = Prior probability of hypothesis

$P\left(H_i \mid D, I\right)$ = Posterior probability of hypothesis

$P(D \mid I) = \sum_i P\left(H_i \mid I\right) P\left(D \mid H_i, I\right)$ is the normalization (tricky to evaluate)

Bayes' rule allows you to evaluate the probability that the hypothesis is true once new data arrives! (How your belief changes depending on the incoming data)

Bayesian Parameter Estimation for Linear Regression

Suppose we have a data (training) set $S = \{(\boldsymbol{x}^{(1)}, y_1), \ldots, (\boldsymbol{x}^{(m)}, y_m)\}$ generated from the god-given rule

$$y = \boldsymbol{w}_{true}^T \boldsymbol{x} + \epsilon \text{ with } \quad \epsilon \in \mathcal{N}(0, \sigma_\epsilon^2)$$

How to use Bayesian inference framework to estimate *both* the *noise* and the *signal*?

Bayesian Parameter Estimation for Linear Regression

Suppose we have a data (training) set $S = \{(\boldsymbol{x}^{(1)}, y_1), \ldots, (\boldsymbol{x}^{(m)}, y_m)\}$ generated from the god-given rule

$$y = \boldsymbol{w}_{true}^T \boldsymbol{x} + \epsilon \text{ with } \quad \epsilon \in \mathcal{N}(0, \sigma_\epsilon^2)$$

How to use Bayesian inference framework to estimate *both* the *noise* and the *signal*?

First: Assume a hypothesis. Hypothesise that this training set is generated from a Gaussian model, whose mean is $\mu(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x}$ and variance σ^2. Namely,

$$P(y \,|\, \boldsymbol{x}, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{w}^T \boldsymbol{x}, \sigma^2) \,.$$

Bayesian Parameter Estimation for Linear Regression

Suppose we have a data (training) set $S = \{(\boldsymbol{x}^{(1)}, y_1), \ldots, (\boldsymbol{x}^{(m)}, y_m)\}$ generated from the god-given rule

$$y = \boldsymbol{w}_{true}^T \boldsymbol{x} + \epsilon \text{ with } \quad \epsilon \in \mathcal{N}(0, \sigma_\epsilon^2)$$

How to use Bayesian inference framework to estimate *both* the *noise* and the *signal*?

First: Assume a hypothesis. Hypothesise that this training set is generated from a Gaussian model, whose mean is $\mu(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x}$ and variance σ^2. Namely,

$$P(y \,|\, \boldsymbol{x}, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{w}^T \boldsymbol{x}, \sigma^2)\,.$$

Second: Calculate the *likelihood* (that such hypothesis leads to observed data) $P\left(D \mid \boldsymbol{\theta}\right) \equiv P\left(D \mid H_\theta\right)$

$$P\left(D \mid \boldsymbol{\theta}\right) = \Pi_{i=1}^{m} \left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2} (y_i - \boldsymbol{w}^T \boldsymbol{x}^{(i)})^2 \right] \right)$$

Bayesian Parameter Estimation for Linear Regression

Suppose we have a data (training) set $S = \{(\boldsymbol{x}^{(1)}, y_1), \ldots, (\boldsymbol{x}^{(m)}, y_m)\}$ generated from the god-given rule

$$y = \boldsymbol{w}_{true}^T \boldsymbol{x} + \epsilon \text{ with } \quad \epsilon \in \mathcal{N}(0, \sigma_\epsilon^2)$$

How to use Bayesian inference framework to estimate *both* the *noise* and the *signal*?

First: Assume a hypothesis. Hypothesise that this training set is generated from a Gaussian model, whose mean is $\mu(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x}$ and variance σ^2. Namely,

$$P(y \,|\, \boldsymbol{x}, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{w}^T \boldsymbol{x}, \sigma^2) \,.$$

Second: Calculate the *likelihood* (that such hypothesis leads to observed data) $P\left(D \mid \boldsymbol{\theta}\right) \equiv P\left(D \mid H_\theta\right)$

$$P\left(D \mid \boldsymbol{\theta}\right) = \Pi_{i=1}^m \left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2}(y_i - \boldsymbol{w}^T \boldsymbol{x}^{(i)})^2 \right] \right)$$

Third: Pick the hypothesis that maximises the likelihood, or equivalently the *log-likelihood* in this case. This principle is called the Maximum Likelihood Estimate (MLE)

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$$

where the *log-likelihood* is defined as

$$L(\boldsymbol{\theta}) = \sum_{i=1}^m \log P(y_i \,|\, \boldsymbol{x}^{(i)}, \boldsymbol{\theta})$$

In our case, the log-likelihood reads

$$L(\boldsymbol{\theta}) = -\frac{1}{2\sigma^2}\sum_{i=1}^{m}(y_i - \boldsymbol{w}^T\boldsymbol{x}^{(i)})^2 - \frac{m}{2}\log(2\pi\sigma^2)$$

In our case, the log-likelihood reads

$$L(\boldsymbol{\theta}) = -\frac{1}{2\sigma^2}\sum_{i=1}^{m}(y_i - \boldsymbol{w}^T\boldsymbol{x}^{(i)})^2 - \frac{m}{2}\log(2\pi\sigma^2)$$

$$= -\frac{1}{2\sigma^2}\|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|_2^2 - \frac{m}{2}\log(2\pi\sigma^2)$$

Maximising the log-likelihood, with respect to $\boldsymbol{w}$, is identical to solving the least square problem!
In addition, the variance σ^2 of the noise can be learned here!

In our case, the log-likelihood reads

$$L(\boldsymbol{\theta}) = -\frac{1}{2\sigma^2}\sum_{i=1}^{m}(y_i - \boldsymbol{w}^T\boldsymbol{x}^{(i)})^2 - \frac{m}{2}\log(2\pi\sigma^2)$$

$$= -\frac{1}{2\sigma^2}\|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|_2^2 - \frac{m}{2}\log(2\pi\sigma^2)$$

Maximising the log-likelihood, with respect to $\boldsymbol{w}$, is identical to solving the least square problem!
In addition, the variance σ^2 of the noise can be learned here!

Recalling

$$P(\theta \mid D) = \frac{P(\theta)\,P(D \mid \theta)}{P(D)}$$

In MLE, we pick the most plausible hypothesis by maximising the likelihood, *with uninformative prior*.

In our case, the log-likelihood reads

$$L(\boldsymbol{\theta}) = -\frac{1}{2\sigma^2}\sum_{i=1}^{m}(y_i - \boldsymbol{w}^T\boldsymbol{x}^{(i)})^2 - \frac{m}{2}\log(2\pi\sigma^2)$$

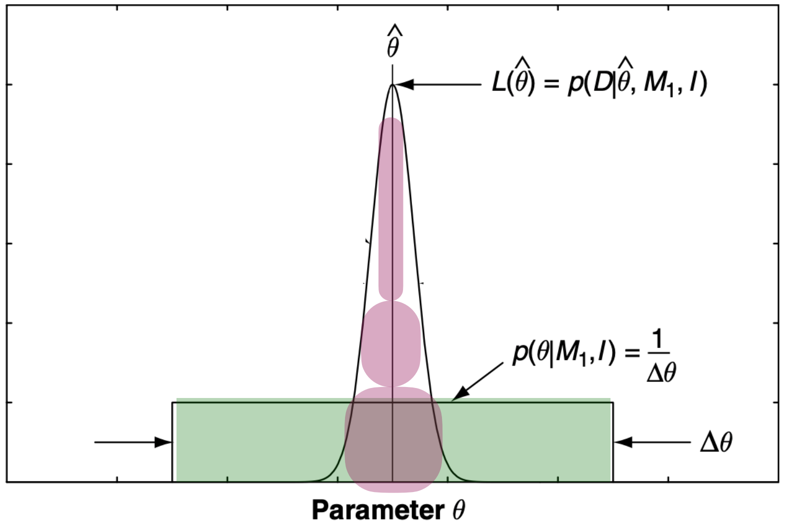
$$= -\frac{1}{2\sigma^2}\|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|_2^2 - \frac{m}{2}\log(2\pi\sigma^2)$$

Maximising the log-likelihood, with respect to $\boldsymbol{w}$, is identical to solving the least square problem!

In addition, the variance σ^2 of the noise can be learned here!

Recalling

$$P\left(\boldsymbol{\theta} \mid D\right) = \frac{P\left(\boldsymbol{\theta}\right) P\left(D \mid \boldsymbol{\theta}\right)}{P(D)}$$



In MLE, we pick the most plausible hypothesis by maximising the likelihood, *with uninformative prior*.

In our case, the log-likelihood reads

$$L(\boldsymbol{\theta}) = -\frac{1}{2\sigma^2}\sum_{i=1}^{m}(y_i - \boldsymbol{w}^T\boldsymbol{x}^{(i)})^2 - \frac{m}{2}\log(2\pi\sigma^2)$$

$$= -\frac{1}{2\sigma^2}\|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|_2^2 - \frac{m}{2}\log(2\pi\sigma^2)$$

Maximising the log-likelihood, with respect to $\boldsymbol{w}$, is identical to solving the least square problem!
In addition, the variance σ^2 of the noise can be learned here!

Recalling

$$P\left(\boldsymbol{\theta} \mid D\right) = \frac{P\left(\boldsymbol{\theta}\right) P\left(D \mid \boldsymbol{\theta}\right)}{P(D)}$$

In MLE, we pick the most plausible hypothesis by maximising the likelihood, *independent of prior*.

Third (alternative): However, if we have prior belief that not all hypothesis are weighted equally, then the prior will affect the model selection. By accounting for the prior, and picking the hypothesis that maximises the posterior is called the maximum-a-posteriori (MAP) estimate.

$$\boxed{\hat{\boldsymbol{\theta}}_{MAP} \equiv \arg\max_{\boldsymbol{\theta}} \left[\log P(D \mid \boldsymbol{\theta}) + \log P(\boldsymbol{\theta})\right]}$$

In homework 2, you'll show that *MAP estimate* with *the priors*

$$P(\boldsymbol{\theta} \,|\, \lambda) = \Pi_j \left[\sqrt{\frac{\lambda}{2\pi}} e^{-\lambda \theta_j^2} \right] \qquad \text{(\textit{Gaussian prior})}$$

$$P(\boldsymbol{\theta} \,|\, \lambda) = \Pi_j \left[\frac{\lambda}{2} e^{-\lambda |\theta_j|} \right] \qquad \text{(\textit{Laplace prior})}$$

is similar to solving the Ridge regression, and LASSO, respectively.

In homework 2, you'll show that *MAP estimate* with *the priors*

$$P(\boldsymbol{\theta} \mid \lambda) = \Pi_j \left[\sqrt{\frac{\lambda}{2\pi}} e^{-\lambda \theta_j^2} \right] \qquad \text{(}\textit{Gaussian prior}\text{)}$$

$$P(\boldsymbol{\theta} \mid \lambda) = \Pi_j \left[\frac{\lambda}{2} e^{-\lambda |\theta_j|} \right] \qquad \text{(}\textit{Laplace prior}\text{)}$$

is similar to solving the Ridge regression, and LASSO, respectively.

Note that, *informative priors reduce variance*, *while increasing bias of a generative model*.

In homework 2, you'll show that *MAP estimate* with *the priors*

$$P(\boldsymbol{\theta} \mid \lambda) = \Pi_j \left[\sqrt{\frac{\lambda}{2\pi}} e^{-\lambda \theta_j^2} \right] \qquad \textit{(Gaussian prior)}$$

$$P(\boldsymbol{\theta} \mid \lambda) = \Pi_j \left[\frac{\lambda}{2} e^{-\lambda |\theta_j|} \right] \qquad \textit{(Laplace prior)}$$

is similar to solving the Ridge regression, and LASSO, respectively.

Note that, *informative priors reduce variance*, *while increasing bias of a generative model*.

There's much more to be discussed regarding the choice of priors! Tools in information theory will be useful for understanding the role of priors. Flexibility in priors also takes a huge hit from critiques of Bayesian statistics, as different conclusions can be reached if one is not careful.

But, in defense of subjective priors in Bayesian methods, E.T. Jaynes[1] once stated

> "The only thing objectivity requires of a scientific approach is that experimenters with the same state of knowledge reach the same conclusion."

1. Edwin Jaynes is a legendary physicist who married frequentist statistics, Bayesian statistics, and statistical mechanics view of the universe together. His masterful thoughts on probabilities are written down in his masterpiece's [Probability Theory: The Logic of Science]

In homework 2, you'll show that *MAP estimate* with *the priors*

$$P(\boldsymbol{\theta} \mid \lambda) = \Pi_j \left[\sqrt{\frac{\lambda}{2\pi}} e^{-\lambda \theta_j^2} \right] \qquad \textit{(Gaussian prior)}$$

$$P(\boldsymbol{\theta} \mid \lambda) = \Pi_j \left[\frac{\lambda}{2} e^{-\lambda |\theta_j|} \right] \qquad \textit{(Laplace prior)}$$

is similar to solving the Ridge regression, and LASSO, respectively.

Note that, *informative priors reduce variance*, *while increasing bias of a generative model*.

There's much more to be discussed regarding the choice of priors! Tools in information theory will be useful for understanding the role of priors. Flexibility in priors also takes a huge hit from critiques of Bayesian statistics, as different conclusions can be reached if one is not careful.

But, in defense of subjective priors in Bayesian methods, E.T. Jaynes[1] once stated

> "The only thing objectivity requires of a scientific approach is that experimenters with the same state of knowledge reach the same conclusion."

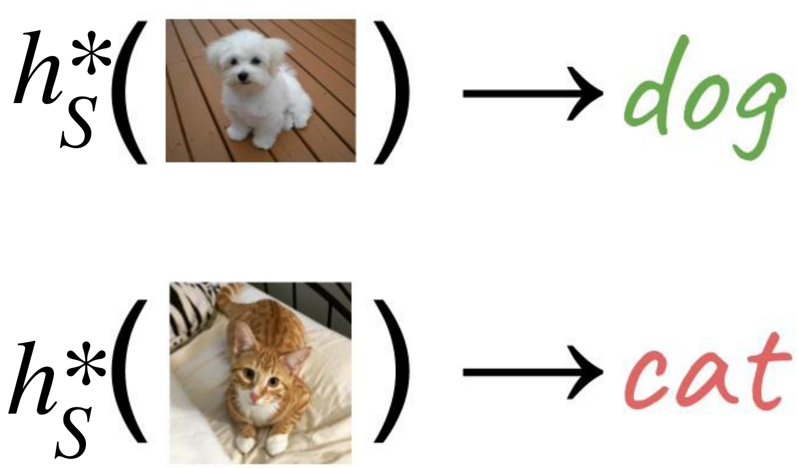
We'll move away from Bayesian inference for now! Hopefully we'll revisit this rich topic again soon!

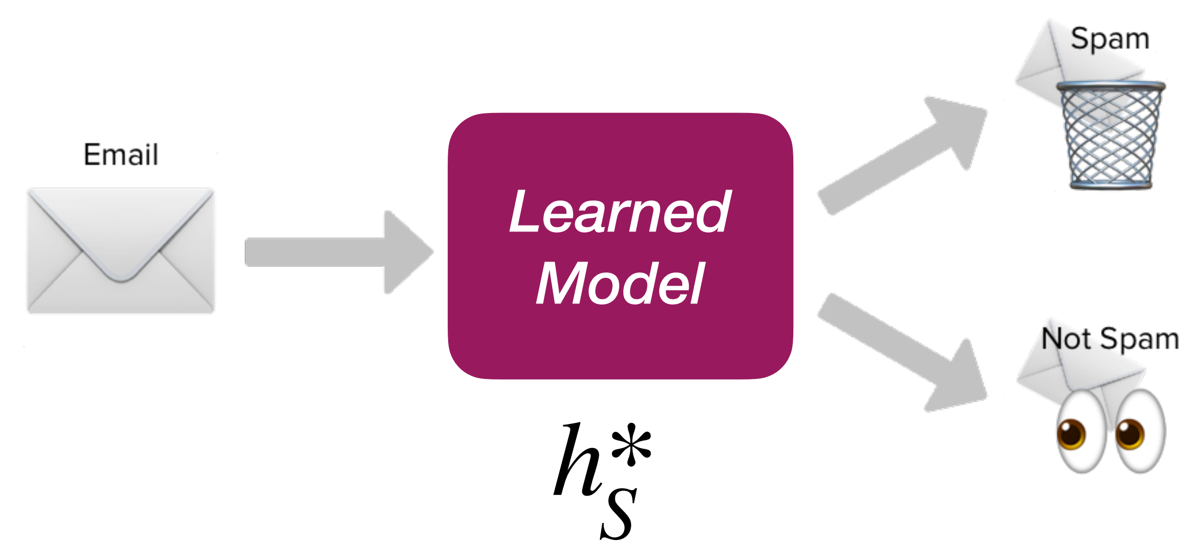
1. Edwin Jaynes is a legendary physicist who married frequentist statistics, Bayesian statistics, and statistical mechanics view of the universe together. His masterful thoughts on probabilities are written down in his masterpiece's [Probability Theory: The Logic of Science]

Back to (Supervised) Statistical Learning Theory

So far, we've seen examples on continuous label (regression) supervised learning.

What about discrete label (classification)?





Framework of Statistical Learning Theory

(supervised learning)

X: Instance Space (e.g. $\mathbb{R}^{16\times 16}$ for 16x16 greyscale images)

Y: Label Space (e.g. $\mathbb{R}$ for regression or $\{1,\ldots,k\}$ for multi-class classification)

$\mathscr{D}$: Probability Distribution over $X \times Y$ (unknown, *but can sample from*)

$\ell : Y \times Y \to \mathbb{R}_{\geq 0}$ Loss or Cost Function (e.g. $\ell(y,\hat{y}) = (y-\hat{y})^2$ for $Y = \mathbb{R}$)

Objective

Given a training set $S = \left\{ \left(x_i, y_i\right) \right\}_{i=1}^{m}$ drawn i.i.d. from $\mathscr{D}$, return hypothesis (predictor)

$h : X \to Y$ that minimizes the population loss or expected risk:

$$\boxed{L_{\mathscr{D}}(h) := \mathbb{E}_{(x,y)\sim\mathscr{D}}[\ell(y, h(x))]}$$

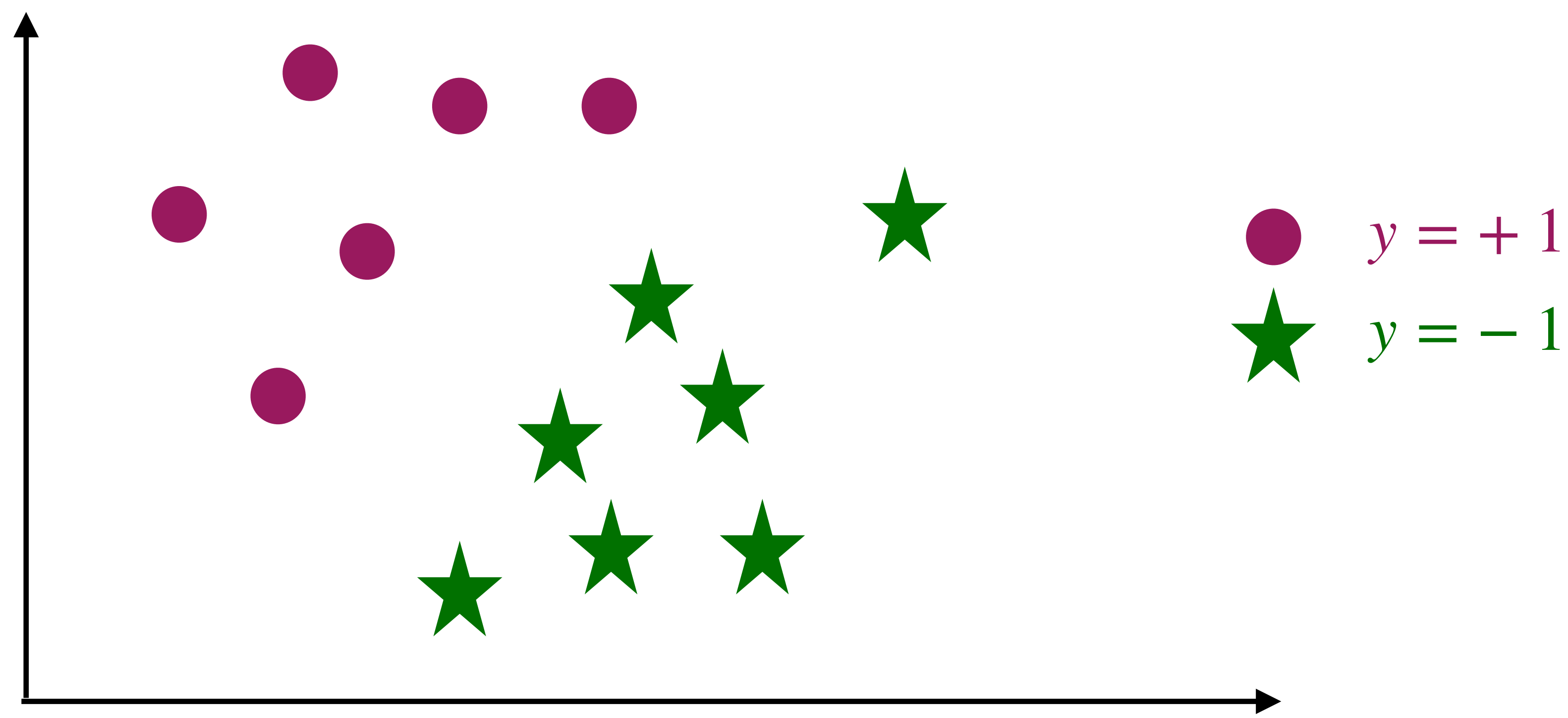
Approximate Approach

Predetermine or assume a hypotheses space $\mathscr{H} \subset Y^X$, and return hypothesis $h \in \mathscr{H}$ that minimizes sample loss or empirical loss or empirical risk:

$$\boxed{L_S(h) := \frac{1}{m}\sum_{i=1}^{m} \ell\left(y_i, h\left(x_i\right)\right)}$$

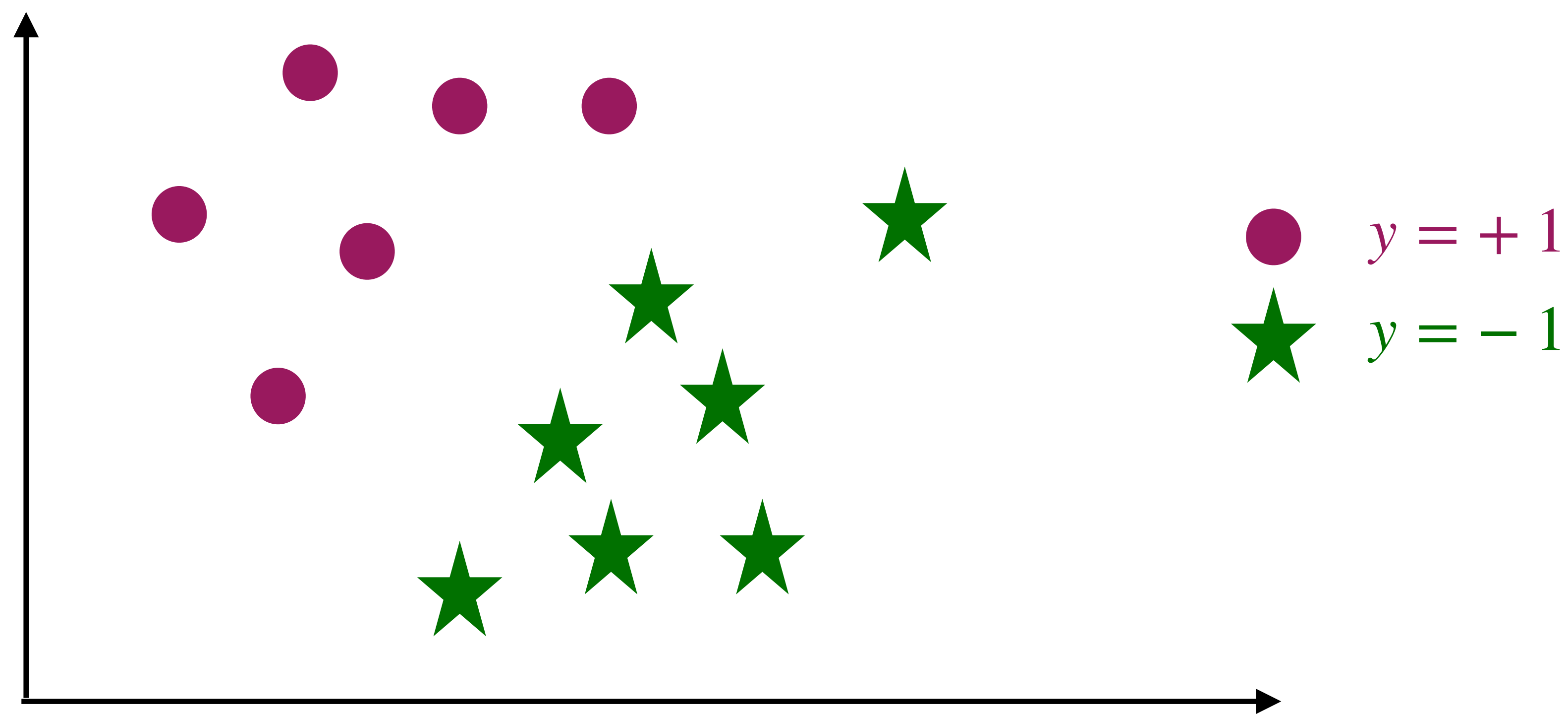
Empirical Risk Minimization

Simplest Binary Classification: the Perceptron Algorithm



Simplest Binary Classification: the Perceptron Algorithm

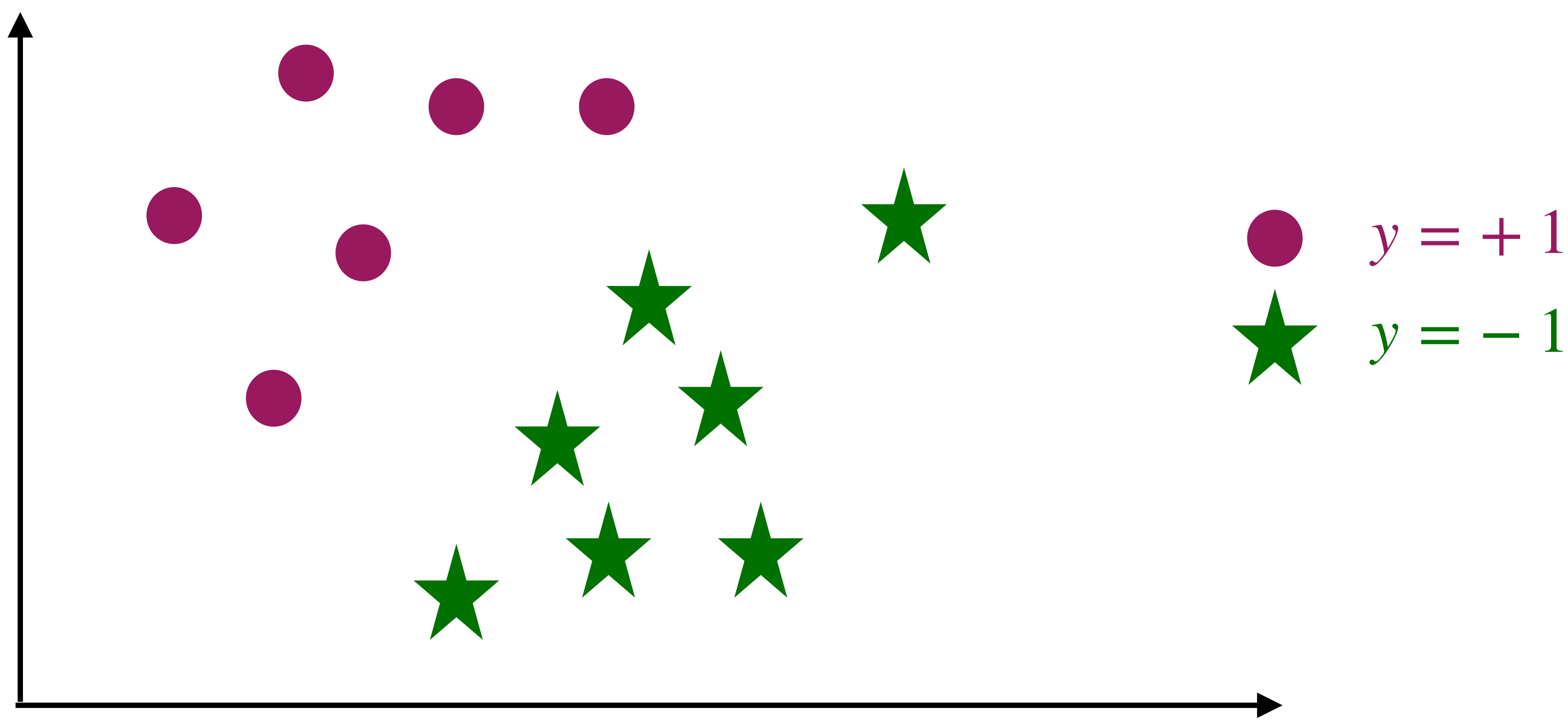
The first model of “neural computation” (Rosenblatt 1957)

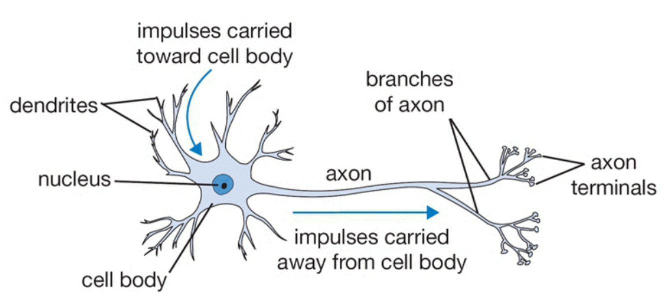


$$h_\theta(\boldsymbol{x}) = \text{sign}\left(\boldsymbol{w}^T \boldsymbol{x} + b\right)$$

Simplest Binary Classification: the Perceptron Algorithm

The first model of "neural computation" (Rosenblatt 1957)

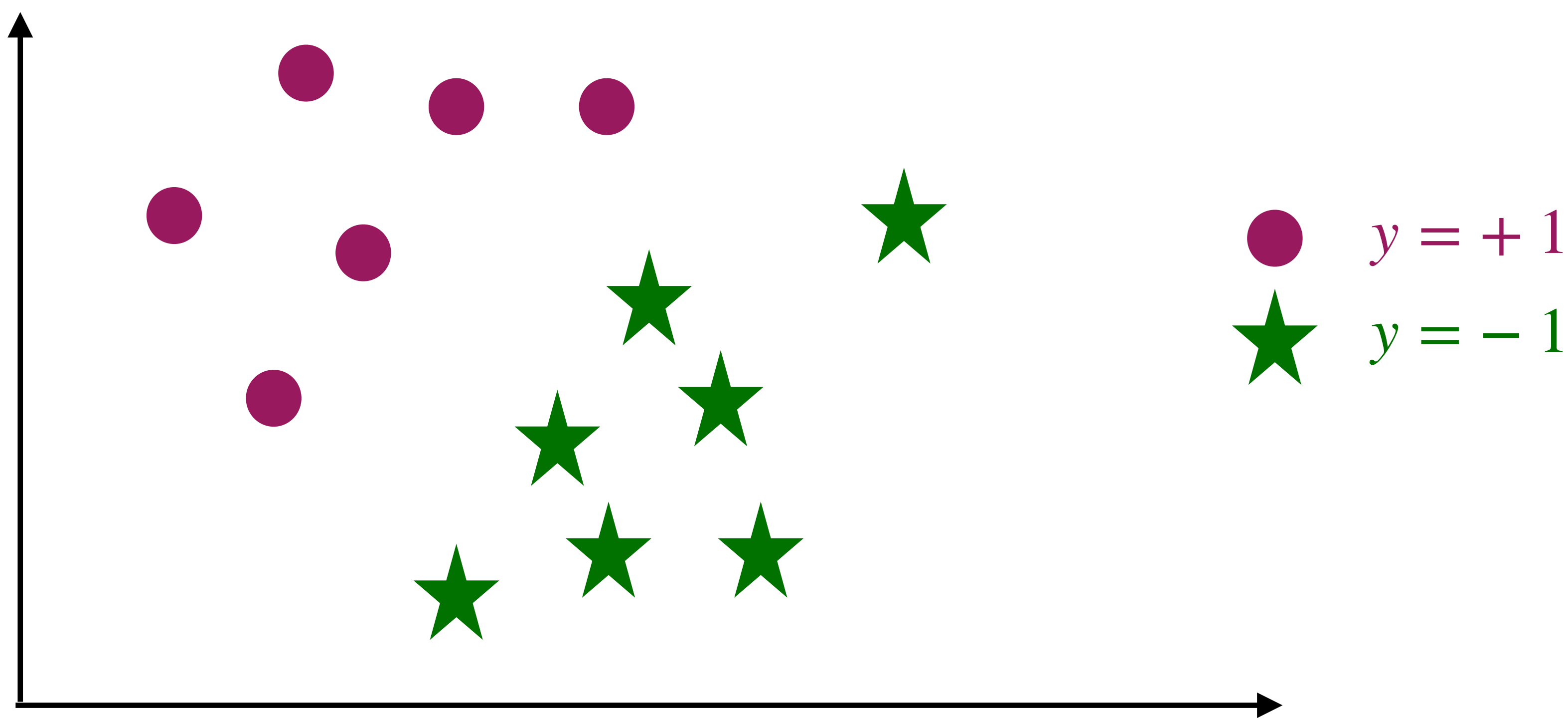




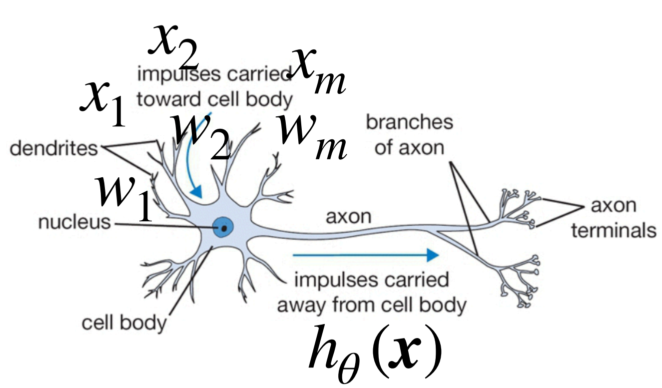
$$h_\theta(\boldsymbol{x}) = \text{sign}\left(\boldsymbol{w}^T\boldsymbol{x} + b\right)$$

Simplest Binary Classification: the Perceptron Algorithm

The first model of “neural computation” (Rosenblatt 1957)

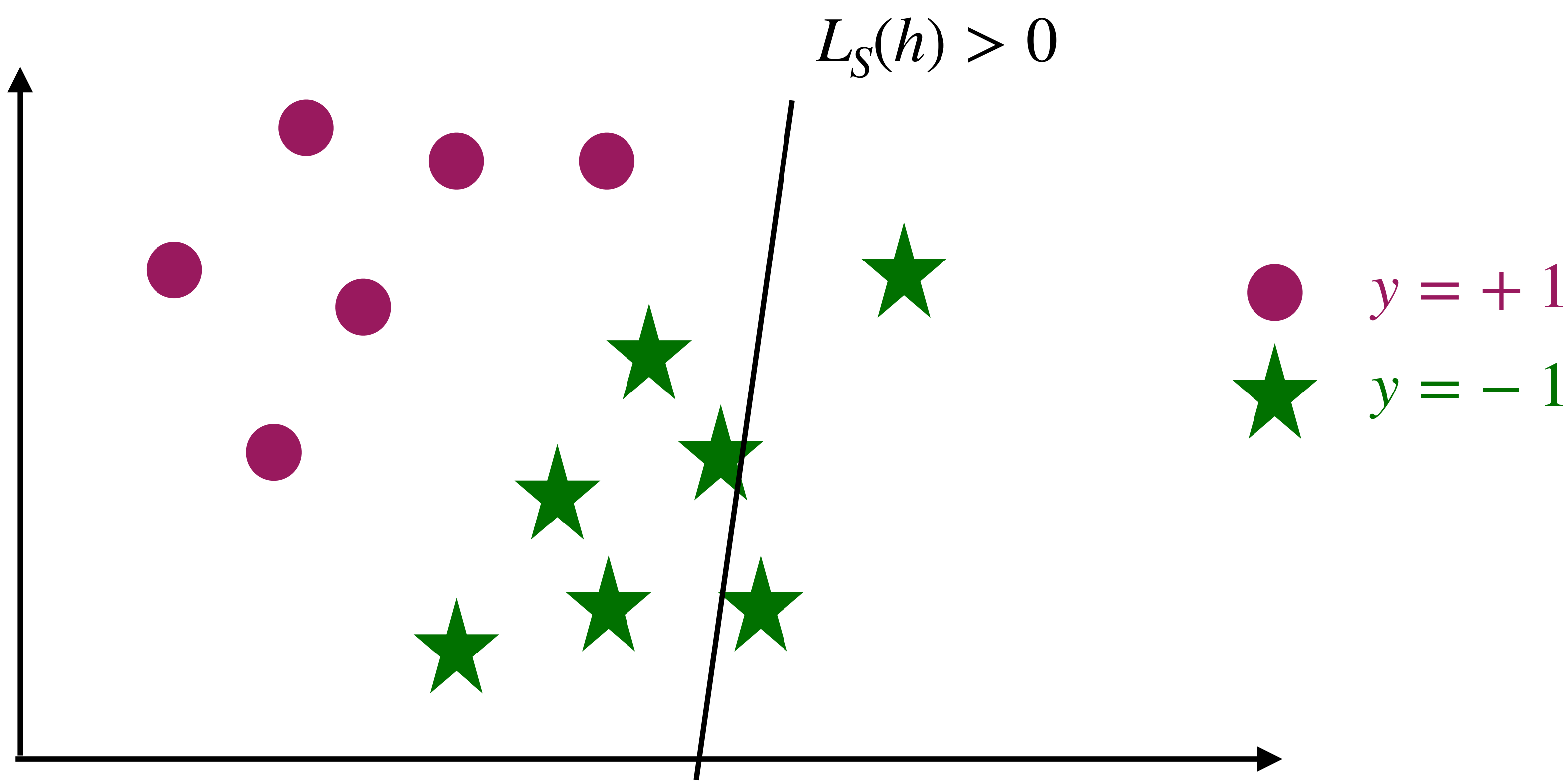


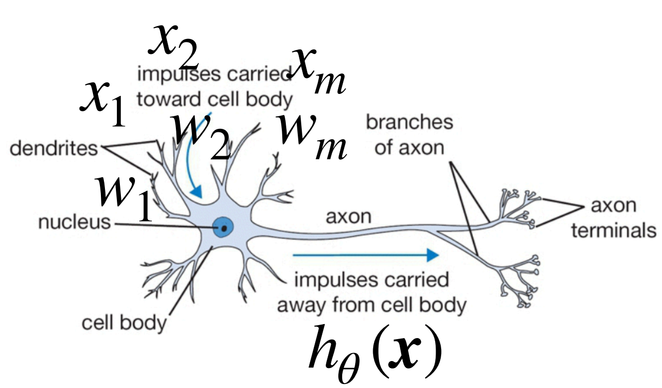
$$h_\theta(\boldsymbol{x}) = \mathrm{sign}\left(\boldsymbol{w}^T\boldsymbol{x} + b\right)$$



Simplest Binary Classification: the Perceptron Algorithm

The first model of “neural computation” (Rosenblatt 1957)





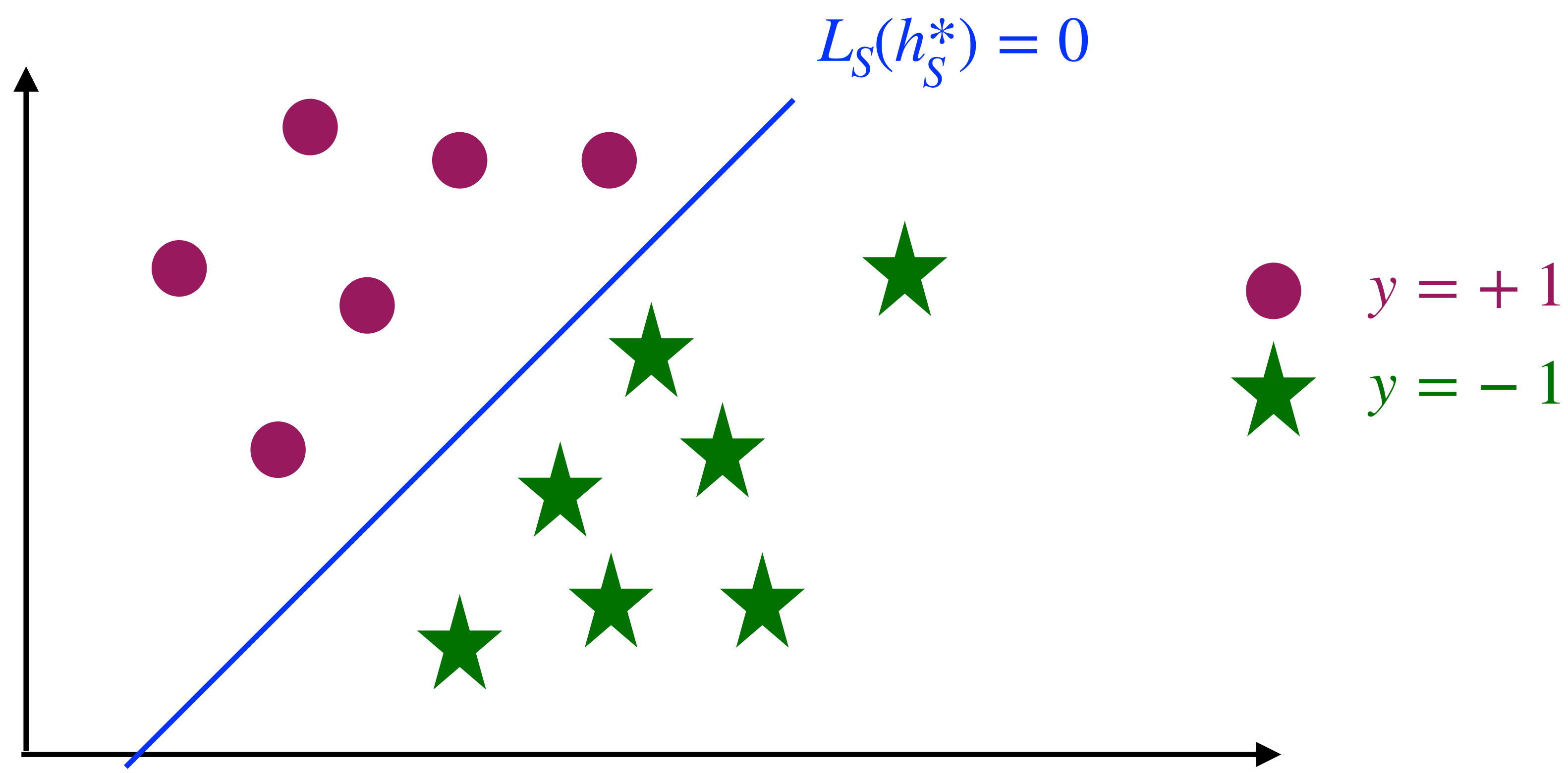
$$h_\theta(\boldsymbol{x}) = \text{sign}\left(\boldsymbol{w}^T\boldsymbol{x} + b\right)$$

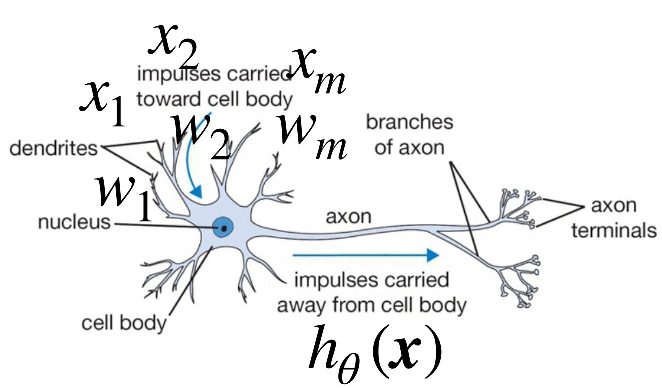
$$\ell(y_i, h_\theta(\boldsymbol{x}_i)) = \mathbf{1}_{h_\theta(\boldsymbol{x}_i) \neq y_i}$$ (misclassification costs 1 point deduction)

$$L_S(h) = \frac{1}{m}\sum_{i=1}^{m} \ell(y_i, h_\theta(\boldsymbol{x}_i))$$

Simplest Binary Classification: the Perceptron Algorithm

The first model of “neural computation” (Rosenblatt 1957)





$$h_\theta(\boldsymbol{x}) = \operatorname{sign}\left(\boldsymbol{w}^T\boldsymbol{x} + b\right)$$

$$\ell(y_i, h_\theta(\boldsymbol{x}_i)) = \mathbf{1}_{h_\theta(\boldsymbol{x}_i) \neq y_i}$$ (misclassification costs 1 point deduction)

$$L_S(h) = \frac{1}{m}\sum_{i=1}^{m} \ell(y_i, h_\theta(\boldsymbol{x}_i))$$

A perceptron learning algorithm guarantees convergence to an empirically optimal hypothesis, *if the training set is linearly separable.*