

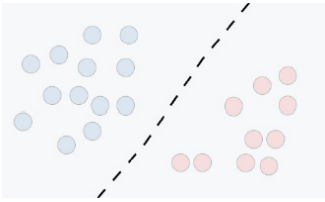
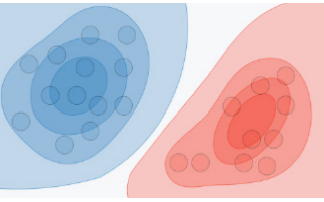
# INTRODUCTION TO SUPERVISED LEARNING

Given a set of data points  $\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$  associated to a set of outcomes  $\{\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(m)}\}$ , we want to build a classifier that learns how to predict  $\mathbf{y}$  from  $\mathbf{x}$ .

- **Type of prediction** - The different types of predictive models are summed up in the table below:

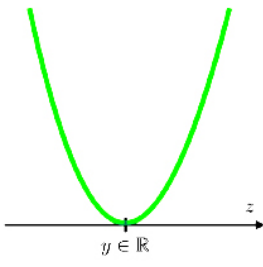
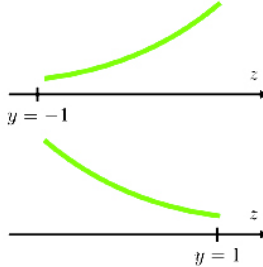
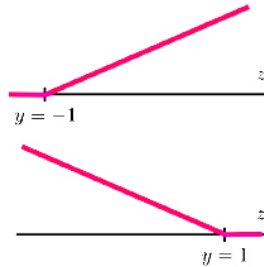
	Regression	Classification
Outcome	Continuous	Class
Examples	Linear regression	Logistic regression, SVM, Naive Bayes

- **Type of model** - The different models are summed up in the table below:

	Discriminative model	Generative model
Goal	Directly estimate $\mathbf{P}(\mathbf{y} \mathbf{x})$	Estimate $\mathbf{P}(\mathbf{y} \mathbf{x})$ to then deduce $\mathbf{P}(\mathbf{x} \mathbf{y})$
What's learned	Decision boundary	Probability distributions of the data
Illustration		
Examples	Regressions, SVMs	GDA, Naive Bayes

## NOTATIONS AND GENERAL CONCEPTS

- **Hypothesis** - The hypothesis is noted  $\mathbf{h}\boldsymbol{\theta}$  and is the model that we choose. For a given input data  $\mathbf{x}^{(i)}$  the model prediction output is  $\mathbf{h}\boldsymbol{\theta}(\mathbf{x}^{(i)})$ .
- **Loss function** - A loss function is a function  $\mathbf{L} : (\mathbf{z}, \mathbf{y}) \in \mathbf{R} \times \mathbf{Y} \rightarrow \mathbf{L}(\mathbf{z}, \mathbf{y}) \in \mathbf{R}$  that takes as inputs the predicted value  $\mathbf{z}$  corresponding to the real data value  $\mathbf{y}$  and outputs how different they are. The common loss functions are summed up in the table below:

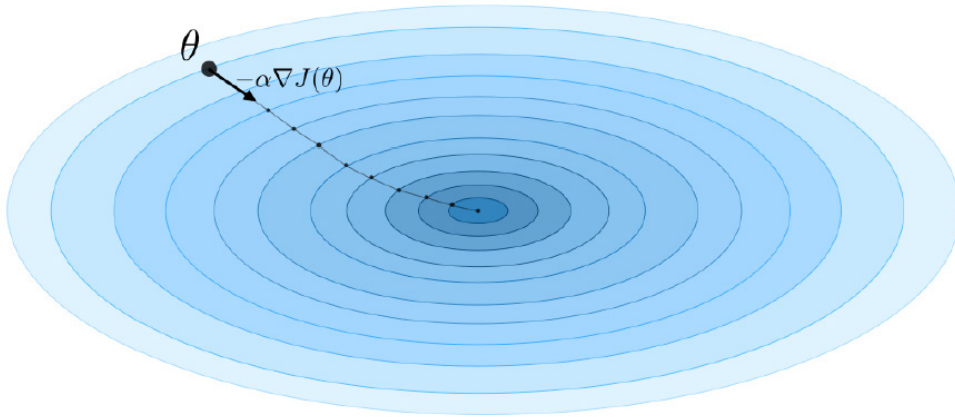
Least squared error	Logistic loss	Hinge loss
$\frac{1}{2}(y - z)^2$	$\log(1 + \exp(-yz))$	$\max(0, 1 - yz)$
		
Linear regression	Logistic regression	SVM

- **Cost function** - The cost function  $\mathbf{J}$  is commonly used to assess the performance of a model, and is defined with the loss function  $\mathbf{L}$  as follows:

$$J(\boldsymbol{\theta}) = \sum_{i=1}^m L(h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}), \mathbf{y}^{(i)})$$

- **Gradient descent** - By noting  $\boldsymbol{\alpha} \in \mathbf{R}$  the learning rate, the update rule for gradient descent is expressed with the learning rate and the cost function  $\mathbf{J}$  as follows:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \boldsymbol{\alpha} \nabla J(\boldsymbol{\theta})$$



Remark: Stochastic gradient descent (SGD) is updating the parameter based on each training example, and batch gradient descent is on a batch of training examples.

- **Likelihood** - The likelihood of a model  $L(\theta)$  given parameters  $\theta$  is used to find the optimal parameters  $\theta$  through likelihood maximization. We have:

$$\theta^{\text{opt}} = \arg \max_{\theta} L(\theta)$$

Remark: in practice, we use the log-likelihood  $\ell(\theta) = \log(L(\theta))$  which is easier to optimize.

- **Newton's algorithm** - Newton's algorithm is a numerical method that finds  $\theta$  such that  $\ell'(\theta) = 0$ . Its update rule is as follows:

$$\theta \leftarrow \theta - \frac{\ell'(\theta)}{\ell''(\theta)}$$

Remark: the multidimensional generalization, also known as the Newton-Raphson method, has the following update rule:

$$\theta \leftarrow \theta - (\nabla_{\theta}^2 \ell(\theta))^{-1} \nabla_{\theta} \ell(\theta)$$

# LINEAR REGRESSION

We assume here that  $y|x; \theta \sim N(\mu, \sigma^2)$

- **Normal equations** - By noting  $\mathbf{x}$  the design matrix, the value of  $\theta$  that minimizes the cost function is a closed-form solution such that:

$$\theta = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

- **LMS algorithm** - By noting  $\alpha$  the learning rate, the update rule of the Least Mean Squares (LMS) algorithm for a training set of  $m$  data points, which is also known as the Widrow-Hoff learning rule, is as follows:

$$\forall j, \quad \theta_j \leftarrow \theta_j + \alpha \sum_{i=1}^m [y^{(i)} - h_{\theta}(x^{(i)})] x_j^{(i)}$$

Remark: the update rule is a particular case of the gradient ascent.

- **LWR** - Locally Weighted Regression, also known as LWR, is a variant of linear regression that weights each training example in its cost function by  $w^{(i)}(\mathbf{x})$ , which is defined with parameter  $\tau \in \mathbb{R}$  as:

$$w^{(i)}(x) = \exp\left(-\frac{(x^{(i)} - x)^2}{2\tau^2}\right)$$

## Classification and logistic regression

- **Sigmoid function** - The sigmoid function  $g$ , also known as the logistic function, is defined as follows:

$$\forall z \in \mathbb{R}, \quad g(z) = \frac{1}{1 + e^{-z}} \in ]0, 1[$$

- **Logistic regression** - We assume here that  $\mathbf{y} | \mathbf{x}; \boldsymbol{\theta} \sim \text{Bernoulli}(\phi)$ . We have the following form:

$$\phi = p(y = 1 | \mathbf{x}; \boldsymbol{\theta}) = \frac{1}{1 + \exp(-\boldsymbol{\theta}^T \mathbf{x})} = g(\boldsymbol{\theta}^T \mathbf{x})$$

Remark: logistic regressions do not have closed form solutions.

- **Softmax regression** - A softmax regression, also called a multiclass logistic regression, is used to generalize logistic regression when there are more than 2 outcome classes. By convention, we set  $\boldsymbol{\theta} \mathbf{K} = \mathbf{0}$ , which makes the Bernoulli parameter of each class be such that:

$$\phi_i = \frac{\exp(\boldsymbol{\theta}_i^T \mathbf{x})}{\sum_{j=1}^K \exp(\boldsymbol{\theta}_j^T \mathbf{x})}$$

## Generalized Linear Models

- **Exponential family** - A class of distributions is said to be in the exponential family if it can be written in terms of a natural parameter, also called the canonical parameter or link function,  $\boldsymbol{\eta}$ , a sufficient statistic  $\mathbf{T}(\mathbf{y})$  and a log-partition function  $a(\boldsymbol{\eta})$  as follows:

$$p(\mathbf{y}; \boldsymbol{\eta}) = b(\mathbf{y}) \exp(\boldsymbol{\eta}^T \mathbf{T}(\mathbf{y}) - a(\boldsymbol{\eta}))$$

Remark: we will often have  $\mathbf{T}(\mathbf{y}) = \mathbf{y}$ . Also,  $(-a(\boldsymbol{\eta}))$  can be seen as a normalization parameter that will make sure that the probabilities sum to one.

The most common exponential distributions are summed up in the following table:

Distribution	$\boldsymbol{\eta}$	$T(\mathbf{y})$	$a(\boldsymbol{\eta})$	$b(\mathbf{y})$
<b>Bernoulli</b>	$\log\left(\frac{\phi}{1-\phi}\right)$	$y$	$\log(1 + \exp(\boldsymbol{\eta}))$	1
<b>Gaussian</b>	$\boldsymbol{\mu}$	$y$	$\frac{\boldsymbol{\eta}^2}{2}$	$\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{y^2}{2}\right)$
<b>Poisson</b>	$\log(\lambda)$	$y$	$e^{\boldsymbol{\eta}}$	$\frac{1}{y!}$
<b>Geometric</b>	$\log(1 - \phi)$	$y$	$\log\left(\frac{e^{\boldsymbol{\eta}}}{1 - e^{\boldsymbol{\eta}}}\right)$	1

- **Assumptions of GLMs** - Generalized Linear Models (GLM) aim at predicting a random variable as  $\mathbf{y}$  a function of  $\mathbf{x} \in \mathbb{R}^{n+1}$  and rely on the following 3 assumptions:

1.  $\mathbf{y} | \mathbf{x}; \boldsymbol{\theta} \sim \text{ExpFamily}(\boldsymbol{\eta})$
2.  $\mathbf{h}_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{E}[\mathbf{y} | \mathbf{x}; \boldsymbol{\theta}]$
3.  $\boldsymbol{\eta} = \boldsymbol{\theta}^T \mathbf{x}$

Remark: ordinary least squares and logistic regression are special cases of generalized linear models.

## SUPPORT VECTOR MACHINES

The goal of support vector machines is to find the line that maximizes the minimum distance to the line.

- **Optimal margin classifier** - The optimal margin classifier  $\mathbf{h}$  is such that:

$$\mathbf{h}(\mathbf{x}) = \text{sign}(\mathbf{w}^T \mathbf{x} - b)$$

where  $(\mathbf{w}, b) \in \mathbb{R}^n \times \mathbb{R}$  is the solution of the following optimization problem:

$$\min \frac{1}{2} \|\mathbf{w}\|^2 \quad \text{such that}$$

$$\mathbf{y}^{(i)} (\mathbf{w}^T \mathbf{x}^{(i)} - b) \geq 1$$

Remark: we say that we use the “kernel trick” to compute the cost function using the kernel because we actually don’t need to know the explicit mapping  $\phi$ , which is often very complicated. Instead, only the values  $\mathbf{K}(\mathbf{x}, \mathbf{z})$  are needed.

■ **Lagrangian** - We define the Lagrangian  $\mathcal{L}(\mathbf{w}, \mathbf{b})$  as follows:

$$\mathcal{L}(\mathbf{w}, \mathbf{b}) = f(\mathbf{w}) + \sum_{i=1}^l \beta_i h_i(\mathbf{w})$$

Remark: the coefficients  $\beta_i$  are called the Lagrange multipliers.

## GENERATIVE LEARNING

A generative model first tries to learn how the data is generated by estimating  $\mathbf{P}(\mathbf{x}|\mathbf{y})$ , which we can then use to estimate  $\mathbf{P}(\mathbf{y}|\mathbf{x})$  by using Bayes’ rule.

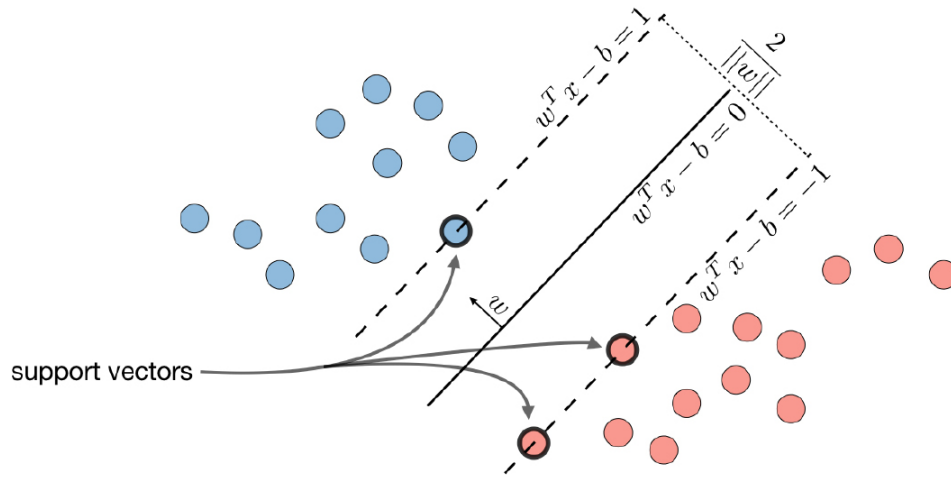
## Gaussian Discriminant Analysis

■ **Setting** - The Gaussian Discriminant Analysis assumes that  $\mathbf{y}$  and  $\mathbf{x}|\mathbf{y} = \mathbf{0}$  and  $\mathbf{x}|\mathbf{y} = \mathbf{1}$  are such that:

1.  $\mathbf{y} \mid \text{Bernoulli}(\phi)$
2.  $\mathbf{x}|\mathbf{y} = \mathbf{0} \sim \mathcal{N}(\mu_0, \Sigma)$
3.  $\mathbf{x}|\mathbf{y} = \mathbf{1} \sim \mathcal{N}(\mu_1, \Sigma)$

■ **Estimation** - The following table sums up the estimates that we find when maximizing the likelihood:

$\hat{\phi}$	$\hat{\mu}_j \quad (j = 0, 1)$	$\hat{\Sigma}$
$\frac{1}{m} \sum_{i=1}^m 1_{\{y^{(i)}=1\}}$	$\frac{\sum_{i=1}^m 1_{\{y^{(i)}=j\}} x^{(i)}}{\sum_{i=1}^m 1_{\{y^{(i)}=j\}}}$	$\frac{1}{m} \sum_{i=1}^m (x^{(i)} - \mu_{y^{(i)}})(x^{(i)} - \mu_{y^{(i)}})^T$



Remark: the decision boundary is defined as  $\mathbf{w}^T \mathbf{x} - \mathbf{b} = 0$

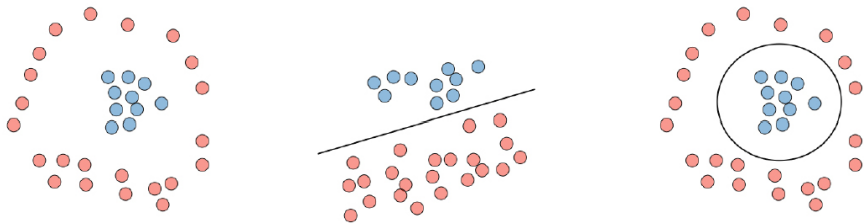
■ **Hinge loss** - The hinge loss is used in the setting of SVMs and is defined as follows:

$$\mathcal{L}(\mathbf{z}, \mathbf{y}) = [1 - \mathbf{y}\mathbf{z}]_+ = \max(0, 1 - \mathbf{y}\mathbf{z})$$

■ **Kernel** - Given a feature mapping  $\phi$ , we define the kernel  $\mathbf{K}$  as follows:

$$\mathbf{K}(\mathbf{x}, \mathbf{z}) = \phi(\mathbf{x})^T \phi(\mathbf{z})$$

In practice, the kernel  $\mathbf{K}$  defined by  $\mathbf{K}(\mathbf{x}, \mathbf{z}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{z}\|^2}{2\sigma^2}\right)$  is called the Gaussian kernel and is commonly used.



Non-linear separability  $\longrightarrow$  Use of a kernel mapping  $\phi$   $\longrightarrow$  Decision boundary in the original space

# Naive Bayes

- **Assumption** - The Naive Bayes model supposes that the features of each data point are all independent:

$$P(x|y) = P(x_1, x_2, \dots | y) = P(x_1|y)P(x_2|y)\dots = \prod_{i=1}^n P(x_i|y)$$

- **Solutions** - Maximizing the log-likelihood gives the following solutions:

$$P(y = k) = \frac{1}{m} \times \#\{j|y^{(j)} = k\}$$

and

$$P(x_i = l|y = k) = \frac{\#\{j|y^{(j)} = k \text{ and } x_i^{(j)} = l\}}{\#\{j|y^{(j)} = k\}}$$

with

$$k \in \{0, 1\} \text{ and } l \in [1, L]$$

Remark: Naive Bayes is widely used for text classification and spam detection.

# TREE-BASED AND ENSEMBLE METHODS

These methods can be used for both regression and classification problems.

- **CART** - Classification and Regression Trees (CART), commonly known as decision trees, can be represented as binary trees. They have the advantage to be very interpretable.
- **Random forest** - It is a tree-based technique that uses a high number of decision trees built out of randomly selected sets of features. Contrary to the simple decision tree, it is highly uninterpretable but its generally good performance makes it a popular algorithm.

Remark: random forests are a type of ensemble methods.

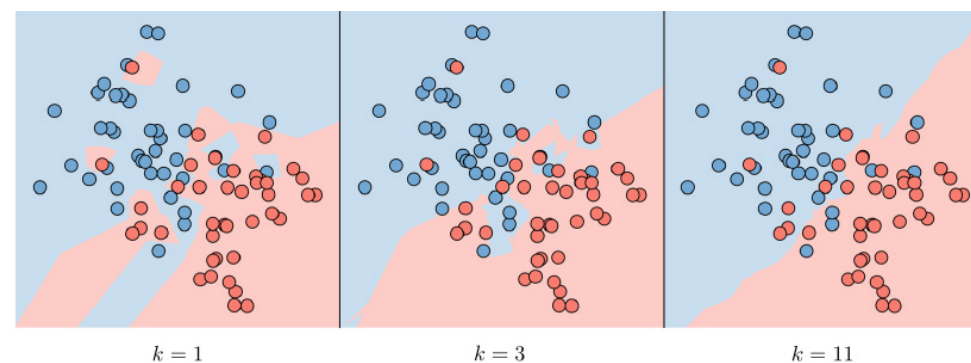
- **Boosting** - The idea of boosting methods is to combine several weak learners to form a stronger one. The main ones are summed up in the table below:

Adaptive boosting	Gradient boosting
<ul style="list-style-type: none"><li>• High weights are put on errors to improve at the next boosting step</li><li>• Known as Adaboost</li></ul>	<ul style="list-style-type: none"><li>• Weak learners are trained on residuals</li><li>• Examples include XGBoost</li></ul>

# OTHER NON-PARAMETRIC APPROACHES

- **k - nearest neighbors** - The **k**-nearest neighbors algorithm, commonly known as **k**-NN, is a non-parametric approach where the response of a data point is determined by the nature of its **k** neighbors from the training set. It can be used in both classification and regression settings.

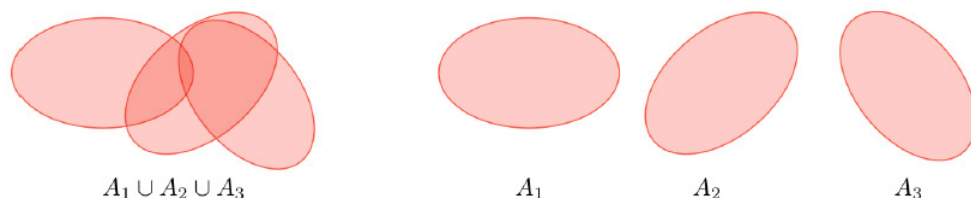
Remark: the higher the parameter **k**, the higher the bias, and the lower the parameter **k**, the higher the variance.



# LEARNING THEORY

■ **Union bound** - Let  $A_1, \dots, A_k$  be events. We have:

$$P(A_1 \cup \dots \cup A_k) \leq P(A_1) + \dots + P(A_k)$$



■ **Hoeffding inequality** - Let  $Z_1, \dots, Z_m$  be  $m$  iid variables drawn from a Bernoulli distribution of parameter  $\phi$ . Let  $\hat{\phi}$  be their sample mean and  $y > 0$  fixed. We have:

$$P(|\phi - \hat{\phi}| > y) \leq 2 \exp(-2y^2 m)$$

Remark: this inequality is also known as the Chernoff bound.

■ **Training error** - For a given classifier  $h$ , we define the training error  $\hat{\epsilon}(h)$ , also known as the empirical risk or empirical error, to be as follows:

$$\hat{\epsilon}(h) = \frac{1}{m} \sum_{i=1}^m 1_{\{h(x^{(i)}) \neq y^{(i)}\}}$$

■ **Probably Approximately Correct (PAC)** - PAC is a framework under which numerous results on learning theory were proved, and has the following set of assumptions:

- the training and testing sets follow the same distribution
- the training examples are drawn independently

■ **Shattering** - Given a set  $S = \{x(1), \dots, x(d)\}$ , and a set of classifiers  $H$ , we say that  $H$  shatters  $S$  if for any set of labels  $\{y(1), \dots, y(d)\}$ , we have:

$$\exists h \in H, \quad \forall i \in [1, d], \quad h(x^{(i)}) = y^{(i)}$$

■ **Upper bound theorem** - Let  $H$  be a finite hypothesis class such that  $|H| = k$  and let  $\delta$  and the sample size  $m$  be fixed. Then, with probability of at least  $1 - \delta$ , we have:

$$\epsilon(\hat{h}) \leq \left( \min_{h \in H} \epsilon(h) \right) + 2 \sqrt{\frac{1}{2m} \log \left( \frac{2k}{\delta} \right)}$$

■ **VC dimension** - The Vapnik-Chervonenkis (VC) dimension of a given infinite hypothesis class  $H$ , noted (VC)  $H$  is the size of the largest set that is shattered by  $H$ .

Remark: the VC dimension of  $H = \{\text{set of linear classifiers in 2 dimensions}\}$  is 3.



■ **Theorem (Vapnik)** - Let  $H$  be given, with (VC)  $H = d$  and  $m$  the number of training examples. With probability at least  $1 - \delta$ , we have:

$$\epsilon(\hat{h}) \leq \left( \min_{h \in H} \epsilon(h) \right) + O \left( \sqrt{\frac{d}{m} \log \left( \frac{m}{d} \right)} + \frac{1}{m} \log \left( \frac{1}{\delta} \right) \right)$$