Supervised Learning cheatsheet

stanford.edu/~shervine/teaching/cs-229/cheatsheet-supervised-learning

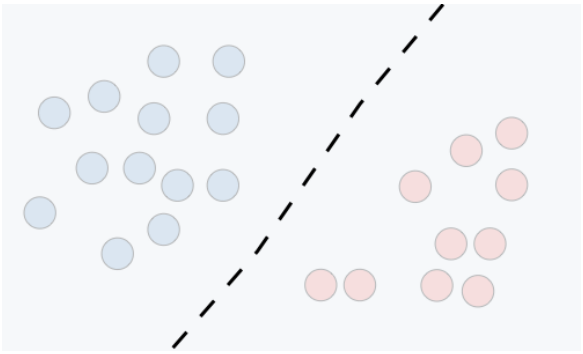
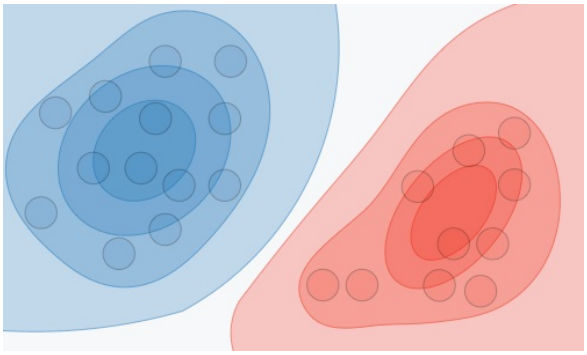
Introduction to Supervised Learning

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

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

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

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

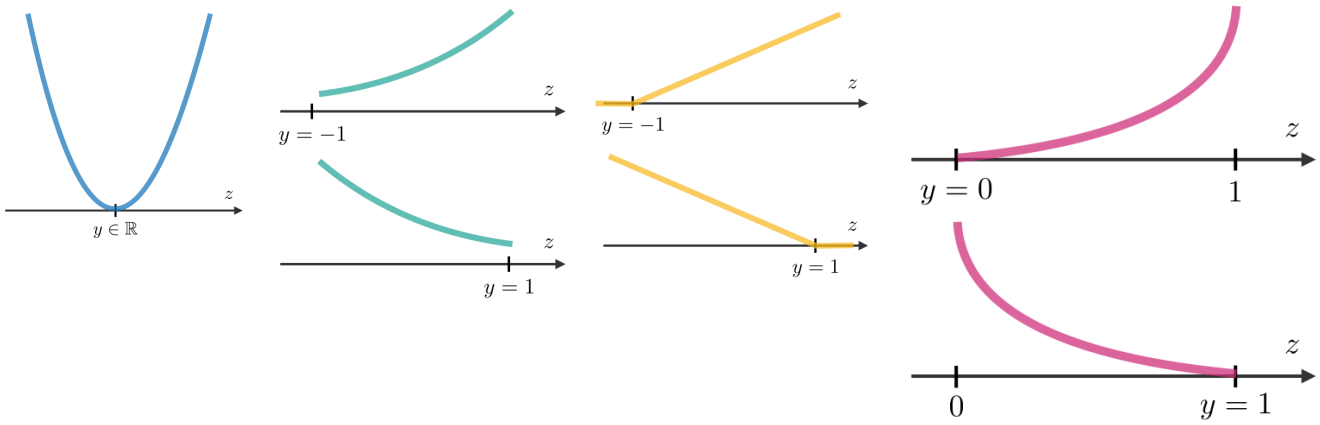
	Discriminative model	Generative model
Goal	Directly estimate P(y|x)P(y|x)	Estimate P(x|y)P(x|y) to then deduce P(y|x)P(y|x)
What's learned	Decision boundary	Probability distributions of the data
Illustration		
Examples	Regressions, SVMs	GDA, Naive Bayes

Notations and general concepts

Hypothesis — The hypothesis is noted hθhθ and is the model that we choose. For a given input data x(i)x(i) the model prediction output is hθ(x(i))hθ(x(i)).

Loss function — A loss function is a function L:(z,y)∈R×Y⟼L(z,y)∈RL:(z,y)∈R×Y⟼L(z,y)∈R that takes as inputs the predicted value zz corresponding to the real data value yy and outputs how different they are. The common loss functions are summed up in the table below:

Least squared error	Logistic loss	Hinge loss	Cross-entropy
12(y−z)212(y−z)2	log(1+exp(−yz))log(1+exp(−yz))	max(0,1−yz)max(0,1−yz)	−[ylog(z)+(1−y)log(1−z)]−[ylog(z)+(1−y)log(1−z)]



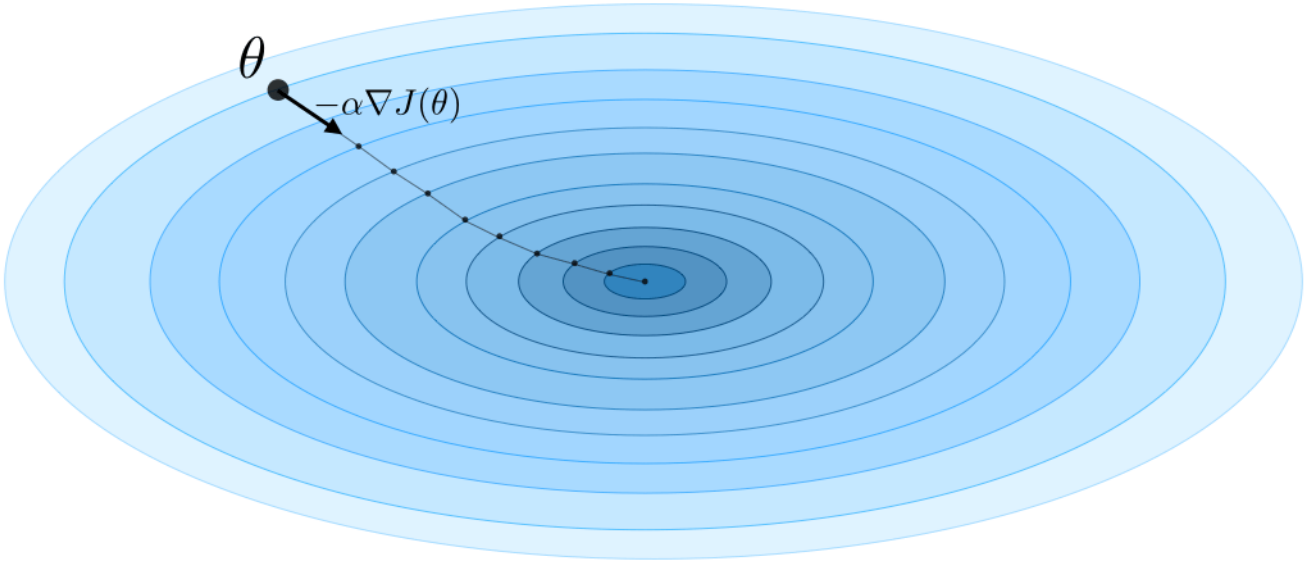
Linear regression | Logistic regression | SVM | Neural Network

Cost function — The cost function JJ is commonly used to assess the performance of a model, and is defined with the loss function LL as follows:

$$J(\theta)=m\sum i=1L(h\theta(x(i)),y(i))J(\theta)=\sum i=1mL(h\theta(x(i)),y(i))$$

Gradient descent — By noting $\alpha \in R\alpha \in R$ the learning rate, the update rule for gradient descent is expressed with the learning rate and the cost function JJ as follows:

$$\theta \longleftarrow \theta - \alpha \nabla J(\theta)\theta \longleftarrow \theta - \alpha \nabla J(\theta)$$



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 $L(\theta)L(\theta)$ given parameters $\theta\theta$ is used to find the optimal parameters $\theta\theta$ through maximizing the likelihood. In practice, we use the log-likelihood $\ell(\theta)=\log(L(\theta))\ell(\theta)=\log(L(\theta))$ which is easier to optimize. We have:

θopt=arg max θL(θ)θopt=arg max θL(θ)

Newton's algorithm — The Newton's algorithm is a numerical method that finds θθ such that ℓ′(θ)=0ℓ′(θ)=0. Its update rule is as follows:

θ←θ−ℓ′(θ)ℓ′′(θ)θ←θ−ℓ′(θ)ℓ″(θ)

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

θ←θ−(∇2θℓ(θ))−1∇θℓ(θ)θ←θ−(∇θ2ℓ(θ))−1∇θℓ(θ)

Linear models

Linear regression

We assume here that y|x;θ∼N(μ,σ2)y|x;θ∼N(μ,σ2)

Normal equations — By noting XX the design matrix, the value of θθ that minimizes the cost function is a closed-form solution such that:

θ=(XTX)−1XTyθ=(XTX)−1XTy

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

∀j,θj←θj+αm∑i=1[y(i)−hθ(x(i))]x(i)j∀j,θj←θj+α∑i=1m[y(i)−hθ(x(i))]xj(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)(x)w(i)(x), which is defined with parameter τ∈Rτ∈R as:

w(i)(x)=exp(−(x(i)−x)22τ2)w(i)(x)=exp(−(x(i)−x)22τ2)

Classification and logistic regression

Sigmoid function — The sigmoid function gg, also known as the logistic function, is defined as follows:

∀z∈R,g(z)=11+e−z∈]0,1[∀z∈R,g(z)=11+e−z∈]0,1[

Logistic regression — We assume here that y|x;θ∼Bernoulli(ϕ)y|x;θ∼Bernoulli(ϕ). We have the following form:

ϕ=p(y=1|x;θ)=11+exp(−θTx)=g(θTx)ϕ=p(y=1|x;θ)=11+exp(−θTx)=g(θTx)

Remark: there is no closed form solution for the case of logistic regressions.

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 θK=0θK=0, which makes the Bernoulli parameter ϕiϕi of each class ii equal to:

ϕi=exp(θTix)K∑j=1exp(θTjx)ϕi=exp(θiTx)∑j=1Kexp(θjTx)

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, ηη, a sufficient statistic T(y)T(y) and a log-partition function a(η)a(η) as follows:

p(y;η)=b(y)exp(ηT(y)−a(η))p(y;η)=b(y)exp(ηT(y)−a(η))

Remark: we will often have T(y)=yT(y)=y. Also, exp(−a(η))exp(−a(η)) can be seen as a normalization parameter that will make sure that the probabilities sum to one.

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

Distribution	ηη	T(y)T(y)	a(η)a(η)	b(y)b(y)
Bernoulli	log(ϕ1−ϕ)log(ϕ1−ϕ)	yy	log(1+exp(η))log(1+exp(η))	11
Gaussian	μμ	yy	η22η22	1√2πexp(−y22)12πexp(−y22)
Poisson	log(λ)log(λ)	yy	eηeη	1y!1y!
Geometric	log(1−ϕ)log(1−ϕ)	yy	log(eη1−eη)log(eη1−eη)	11

Assumptions of GLMs — Generalized Linear Models (GLM) aim at predicting a random variable yy as a function of x∈Rn+1x∈Rn+1 and rely on the following 3 assumptions:

(1)y|x;θ∼ExpFamily(η)(1)y|x;θ∼ExpFamily(η)

(2)hθ(x)=E[y|x;θ](2)hθ(x)=E[y|x;θ]

(3)η=θTx(3)η=θTx

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

Support Vector Machines

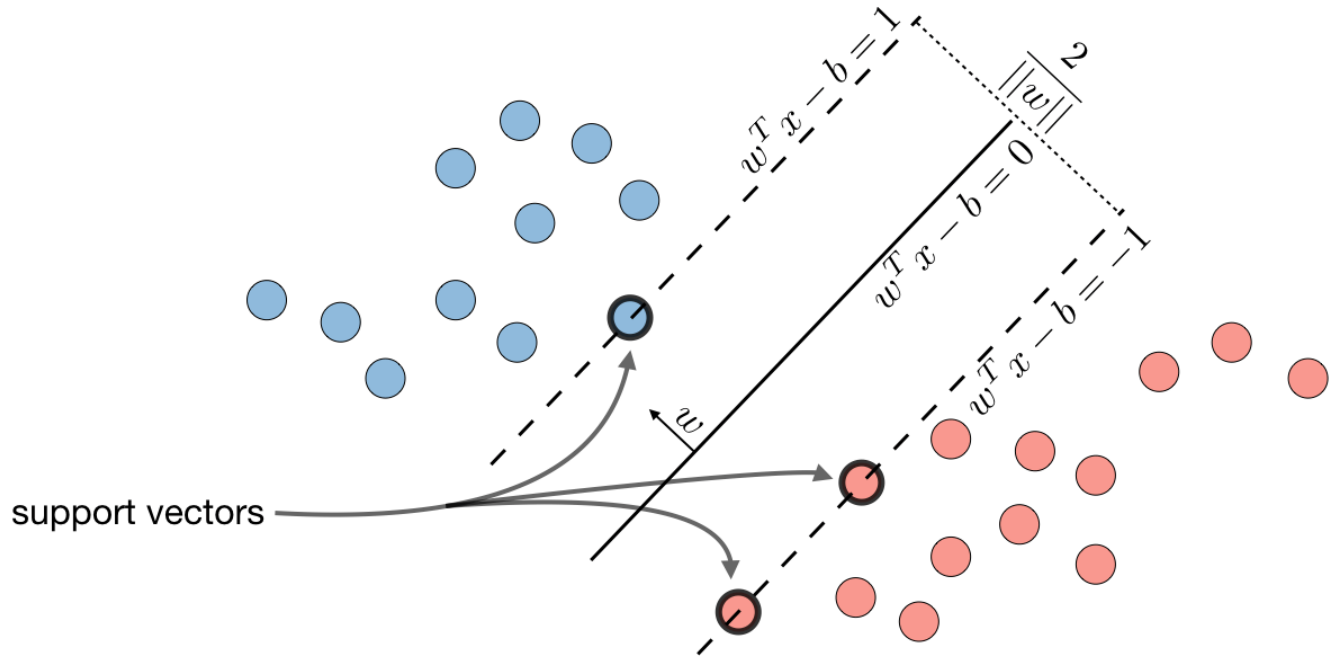
The goal of support vector machines is to find the line that maximizes the minimum distance to the line.

Optimal margin classifier — The optimal margin classifier hh is such that:

h(x)=sign(wTx−b)h(x)=sign(wTx−b)

where (w,b)∈Rn×R(w,b)∈Rn×R is the solution of the following optimization problem:

min12||w||2such that y(i)(wTx(i)−b)⩾1min12||w||2such that y(i)(wTx(i)−b)⩾1



Remark: the line is defined as wTx−b=0wTx−b=0.

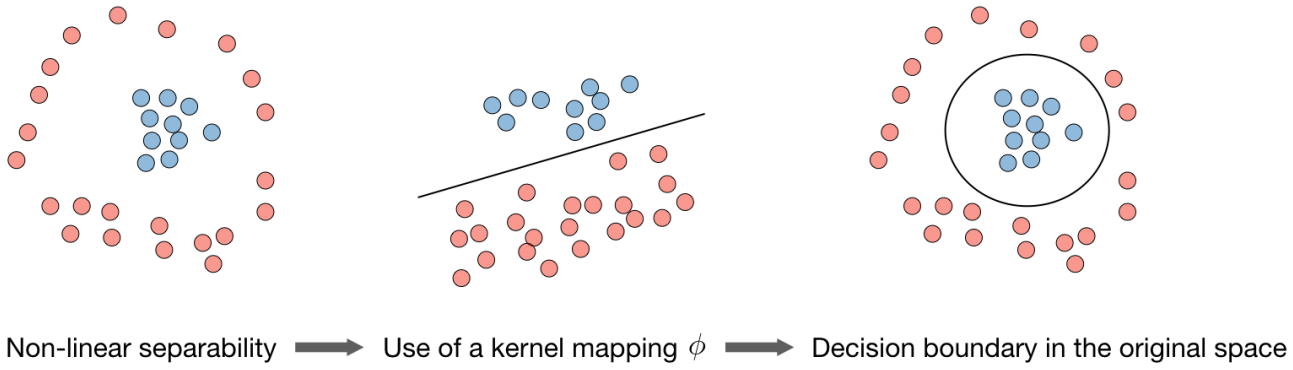
Hinge loss — The hinge loss is used in the setting of SVMs and is defined as follows:

L(z,y)=[1−yz]+=max(0,1−yz)L(z,y)=[1−yz]+=max(0,1−yz)

Kernel — Given a feature mapping $\phi\phi$, we define the kernel KK to be defined as:

$K(x,z)=\phi(x)T\phi(z)K(x,z)=\phi(x)T\phi(z)$

In practice, the kernel KK defined by $K(x,z)=\exp(-||x-z||22\sigma2)K(x,z)=\exp(-||x-z||22\sigma2)$ is called the Gaussian kernel and is commonly used.



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 $\phi\phi$, which is often very complicated. Instead, only the values $K(x,z)K(x,z)$ are needed.

Lagrangian — We define the Lagrangian $L(w,b)L(w,b)$ as follows:

$L(w,b)=f(w)+l\sum i=1\beta ihi(w)L(w,b)=f(w)+\sum i=1l\beta ihi(w)$

Remark: the coefficients $\beta i\beta i$ are called the Lagrange multipliers.

Generative Learning

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

Gaussian Discriminant Analysis

Setting — The Gaussian Discriminant Analysis assumes that yy and $x|y=0x|y=0$ and $x|y=1x|y=1$ are such that:

$(1)y\sim Bernoulli(\phi)(1)y\sim Bernoulli(\phi)$

$(2)x|y=0\sim N(\mu 0,\Sigma)(2)x|y=0\sim N(\mu 0,\Sigma)$

$(3)x|y=1\sim N(\mu 1,\Sigma)(3)x|y=1\sim N(\mu 1,\Sigma)$

Estimation — The following table sums up the estimates that we find when maximizing the likelihood:

$\hat{}\phi\phi\hat{}$	$\hat{}\mu j(j=0,1)\mu j\hat{}(j=0,1)$	$\hat{}\Sigma\Sigma\hat{}$
$1mm\sum i=11\{y(i)=1\}1m\sum i=1m1\{y(i)=1\}$	$\sum mi=11\{y(i)=j\}x(i)\sum mi=11\{y(i)=j\}\sum i=1m1\{y(i)=j\}x(i)\sum i=1m1\{y(i)=j\}$	$1mm\sum i=1(x(i)-\mu y(i))(x(i)-\mu y(i))T1m\sum i=1m(x($

Naive Bayes

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

$P(x|y)=P(x1,x2,...|y)=P(x1|y)P(x2|y)...=n\prod i=1P(xi|y)P(x|y)=P(x1,x2,...|y)=P(x1|y)P(x2|y)...=\prod i=1nP(xi|y)$

Solutions — Maximizing the log-likelihood gives the following solutions, with $k\in\{0,1\},l\in[[1,L]]k\in\{0,1\},l\in[[1,L]]$

P(y=k)=1m×#{j|y(j)=k} and P(xi=l|y=k)=#{j|y(j)=k and x(j)i=l}#{j|y(j)=k}P(y=k)=1m×#{j|y(j)=k} and P(xi=l|y=k)=#{j|y(j)=k and xi(j)=l}#{j|y(j)=k}

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.

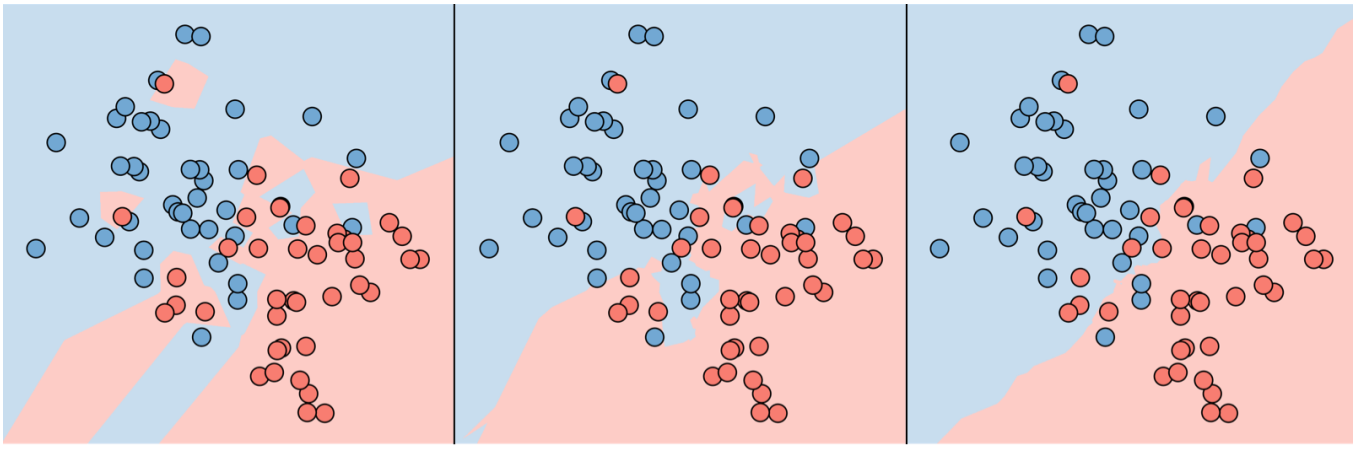
Boosting — The idea of boosting methods is to combine several weak learners to form a stronger one. The main ones are summed up in the table below:

Adaptive boosting	Gradient boosting
• Known as Adaboost • High weights are put on errors to improve at the next boosting step	• Weak learners trained on remaining errors

Other non-parametric approaches

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

Remark: The higher the parameter kk, the higher the bias, and the lower the parameter kk, the higher the variance.

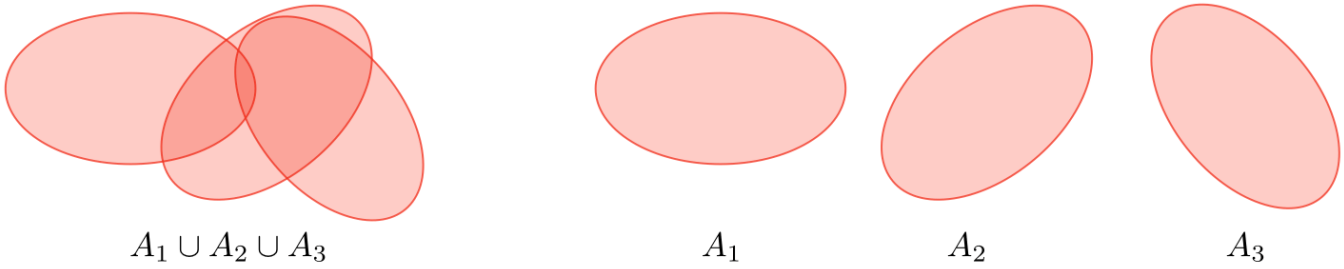


$k = 1$ $k = 3$ $k = 11$

Learning Theory

Union bound — Let A1,...,AkA1,...,Ak be kk events. We have:

P(A1∪...∪Ak)⩽P(A1)+...+P(Ak)P(A1∪...∪Ak)⩽P(A1)+...+P(Ak)



Hoeffding inequality — Let Z1,..,ZmZ1,..,Zm be mm iid variables drawn from a Bernoulli distribution of parameter ϕϕ. Let ^ϕϕ^ be their sample mean and γ>0γ>0 fixed. We have:

P(|ϕ−^ϕ|>γ)⩽2exp(−2γ2m)P(|ϕ−ϕ^|>γ)⩽2exp(−2γ2m)

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

Training error — For a given classifier hh, we define the training error ^ϵ(h)ϵ^(h), also known as the empirical risk or empirical error, to be as follows:

^ϵ(h)=1mm∑i=11{h(x(i))≠y(i)}ϵ^(h)=1m∑i=1m1{h(x(i))≠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),...,x(d)}S={x(1),...,x(d)}, and a set of classifiers HH, we say that HH shatters SS if for any set of labels {y(1),...,y(d)}{y(1),...,y(d)}, we have:

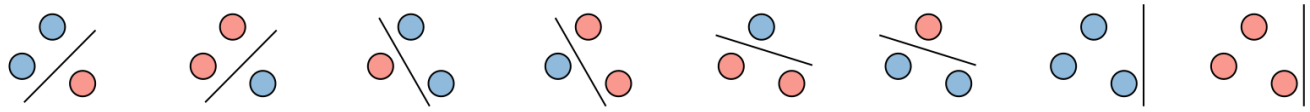
∃h∈H,∀i∈[[1,d]],h(x(i))=y(i)∃h∈H,∀i∈[[1,d]],h(x(i))=y(i)

Upper bound theorem — Let HH be a finite hypothesis class such that |H|=k|H|=k and let δδ and the sample size mm be fixed. Then, with probability of at least 1−δ1−δ, we have:

ϵ(^h)⩽(minh∈Hϵ(h))+2√12mlog(2kδ)ϵ(h^)⩽(minh∈Hϵ(h))+212mlog(2kδ)

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

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



Theorem (Vapnik) — Let HH be given, with VC(H)=dVC(H)=d and mm the number of training examples. With probability at least 1−δ1−δ, we have:

ϵ(^h)⩽(minh∈Hϵ(h))+O(√dmlog(md)+1mlog(1δ))ϵ(h^)⩽(minh∈Hϵ(h))+O(dmlog(md)+1mlog(1δ))