# Learning concepts

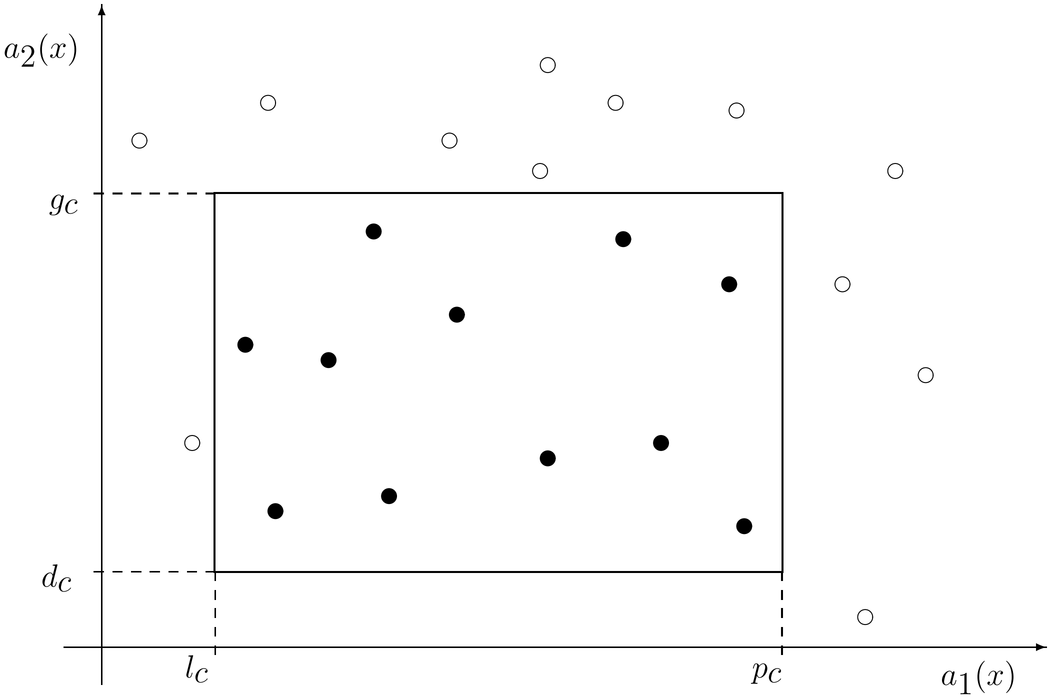
We will consider the domain set $X$, attributes $a \in \mathcal{A}$ as arbitrary functions defined on this domain $a : X \mapsto A$, and a class of concepts $\mathcal{C}$. The concepts $c \in \mathcal{C}$ are functions $c : X \mapsto C$ where $C$ is a set of concept categories of class $\mathcal{C}$.

The **single** concept $c$ has the set of categories $C = \{0, 1\}$. Each single concept determines a subset of a domain including positive samples of this concept $X^c = \{x \in X \mid c(x) = 1\}$.

A **multiple** concept has a set of categories of cardinality $|C| > 2$. In the sample set $P \subseteq X$ of this concept we may distinguish the set of samples with category $d \in C$ denoted by $P^{cd}$ or shorter $P^d$. So $P^d = \{x \in P \mid c(x) = d\}$, for some concept $c$. Therefore, if $c$ is a single concept, then the set of positive samples of the concept $X^c = X^1$, and the set $X^0 = X \setminus X^1$ is the set of all its negative samples.

# Example: axis-aligned rectangles[1]

A single concept $c$ will be designated by a pair of points $(l_c, d_c)$ and $(p_c, g_c)$, while the set of samples of concept $c$ we have as:
$X^c = \{x \in \mathcal{R}^2 \mid l_c \leqslant a_1(x) \leqslant p_c \wedge d_c \leqslant a_2(x) \leqslant g_c\}$



[1] The axis-aligned rectangles example originally from [Kearns, Vazirani, 1994].

# Example: boolean functions

The domain will be a set of $n$-element binary chains, for some $n \geqslant 1$, so $X = \{0, 1\}^n$. The samples of some concept will be $n$-element strings of zeros and ones. They can be described by $n$ attribute functions $a_1, ..., a_n$ representing subsequent bits, where $a_i : X \mapsto \{0, 1\}$.

The concepts can also be represented by logical formulas. For example, for $n = 3$ a concept $c$ could be represented by the formula $\neg a_1(x) \wedge (a_2(x) \vee a_3(x))$, which means that the positive samples of this concept will be all strings with a zero in the first bit, and a one in either the second or the third bit.

The set of all possible different single concepts is equivalent to the set of all the different boolean functions for $n$-element binary strings, and has the cardinality $2^{2^n}$.

# Hypotheses

Hypotheses will represent the results of learning. The set of all hypotheses will be denoted by $\mathcal{H}$. Each hypothesis $h \in \mathcal{H}$ is a function $h : X \mapsto C$, just like the concepts. Exact learning of a concept $c$ is possible only if $\mathcal{C} \subseteq \mathcal{H}$, since only then we can be sure that there is a hypothesis in the hypothesis set, that is identical to the concept we are learning. And the other way around, exact learning of a concept may not be possible if $\mathcal{H} \subset \mathcal{C}, \mathcal{H} \neq \mathcal{C}$.

Since the training samples are described by attributes, then the hypotheses must have the form of functions $h : A_1 \times A_2 ... \times A_n \mapsto C$ where $A_1, A_2, ..., A_n$ are codomains of the attributes defined on $X$. If all the attributes were equal for two different samples $x_1, x_2$, then for any hypothesis $h$ we would have $h(x_1) = h(x_2)$.

# Training information

For the set of samples $P \subseteq X$ we denote:

$P^h$ set of samples from $P$ **covered** by the hypothesis $h$, they are such samples for which $h(x) = 1$ (for single concepts)

$P^{hd}$ set of samples from $P$, for which $h(x) = d, d \in C$ (for any concepts)

A **labeled sample** of a concept $c$ defined for the domain $X$ we denote by $\langle x, c(x) \rangle$, where $x \in X$

A **training series** is a series of labeled samples, where the object $x$ is represented by a vector of attribute values, and the label is its assigned category.

Practically, a training series is a series of attribute vectors with labels.

# Errors in learning concepts

Generally: an error in learning represents the measure of agreement in classifying samples by the goal concept and hypothesis.

The **sample error** $e_P^c(h)$ for the sample set $P$

$$e_P^c(h) = \frac{|\{x \in P | h(x) \neq c(x)\}|}{|P|}$$

where

$$r_P^c(h) = |\{x \in P | h(x) \neq c(x)\}|$$

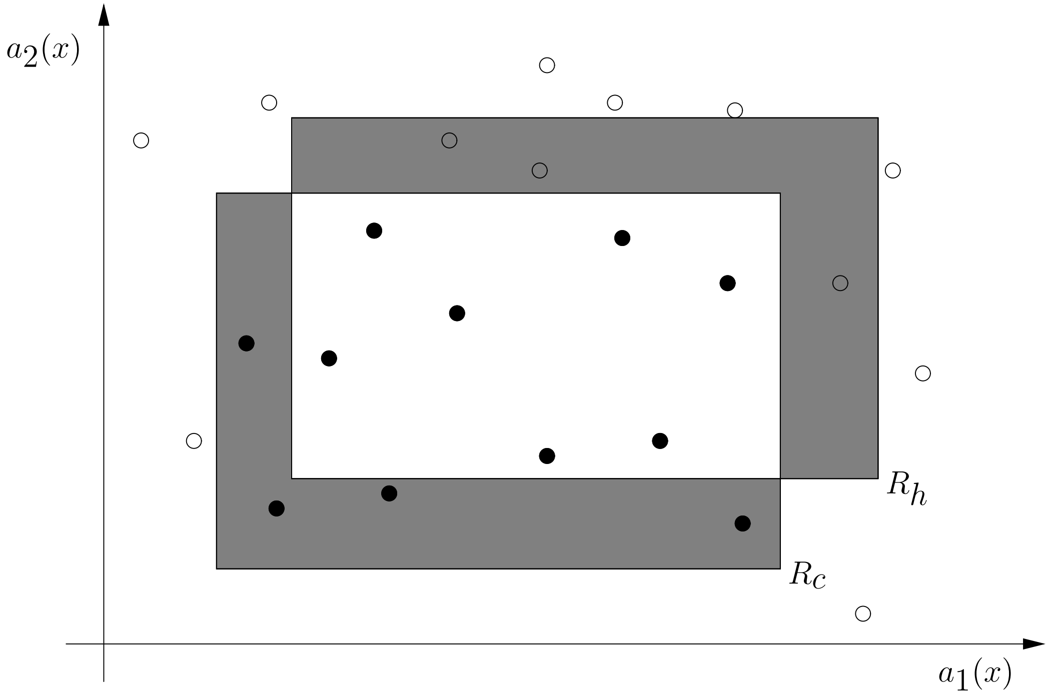
is the number of incorrectly classified samples from set $P$.

The **real error** of a hypothesis is the expected value of the sample error for a randomly chosen set of samples. If the samples are chosen from the domain according to some probability distribution $\Omega$ then the real error shall be:

$$e_\Omega^c(h) = \mathsf{Pr}_{x \in \Omega}(h(x) \neq c(x))$$

where $\mathsf{Pr}_{x \in \Omega}$ denotes the probability computed with the selection of $x$ from $X$ according to the distribution $\Omega$.

# Errors in learning concepts — example



For the concept, the hypothesis, and the 23 samples depicted in the above figure the sample error $e^c_P(h) = \frac{7}{23}$, since 7 samples have been incorrectly classified by the hypothesis (4 positive, and 3 negative).

The real error $e^c_\Omega(h)$ is the probability that a randomly selected sample according to the distribution $\Omega$ will fall into the shaded area, or $\Pr_\Omega(R_c \div R_h)$, where $R_c$ and $R_h$ denote the rectangles representing the concept and the hypothesis, and $R_c \div R_h$ is the symmetric difference of sets.

# The task of inductive concept learning

Given a domain $X$, class of concepts $\mathcal{C}$, and the space of hypotheses $\mathcal{H}$, as well as some fixed but unknown goal concept $c \in \mathcal{C}$, and the training set $T \subseteq X$ and a training set of labeled samples from this set, one has to find a hypothesis $h \in \mathcal{H}$ best approximating the concept $c$ according to some criteria.

The criterion, in the simplest case, could be to minimize the sample error for the training set $e_T^c(h)$.

In reality the goal of learning is to achieve the best approximation of the goal concept in general. If the samples are selected from the domain with a distribution $\Omega$, and the training set was also constructed according to this distribution, then the criterion to select the hypothesis could be to minimize the real error $e_\Omega^c(h)$.

# Other types of inductive learning

When the categories of the goal concept samples are not known, it may be necessary to create them. The task of learning the **concept creation** is two-staged: first stage is to group the samples of a training series into subgroups corresponding to selected categories, and only then it is possible to learn so defined concept.

In learning concepts the set of values is typically a small set of categories. Sometimes it may also be something like the set of real numbers, and then the task of learning a concept becomes learning the **approximation of an unknown function** from a series of examples.

# Learning modes

**Batch mode** — the whole training series is applied at once, with no interactions with the teacher or the environment. This is the simplest learning mode, all algorithms can be used in this mode.

**Incremental mode** — training samples are supplied one by one and the learner should each time adjust her current hypothesis. This mode is useful when there is no definite training set and learning proceeds by making actual observations.

**Epoch mode** — this is similar to the incremental mode, but samples are supplied in batches. The same algorithms are applicable.

**Corrective mode** — the training samples are supplied one by one, but with no labels, and the learner must first calculate her hypothesis for a given sample, and the receives a correction from the teacher.

# Estimating hypotheses errors

**Interval estimation** — estimating some unknown parameter of the population from an **estimator**, which is a random variable with values determined from a random trial of the elements of the population.

Interval estimation works by determining the **confidence intervals** for the estimated parameter based on the estimator. This is done for a specific **confidence level**, which is a probability of the real value of the parameter falling in that interval.

The confidence interval for a parameter $p$ at the confidence level $1-\delta$ for $0 \leqslant \delta < 1$ is each interval to which the value of $p$ belongs with probability $1-\delta$.

The confidence intervals for the real error of a hypothesis can be calculated using the Bernoulli distribution ($k$ successes in $n$ trials). Assuming that a "success" will be an error in classification for the next sample selected from the domain with the distribution $\Omega$, the probability of such success is the real error.

# The PAC model

The PAC (Probably Approximately Correct) model makes the same assumption as the inductive learning task. Additionally, it is assumed the the subsequent samples of the training set $T$ are generated by an **oracle**, which is defined as a random variable EX$(c, \Omega)$ returning labeled samples $\langle x, c(x) \rangle$ where $x \in X$ is a sample drawn according to the distribution $\Omega$.

The task of a learner is to generate a hypothesis $h \in \mathcal{H}$ minimizing the real error $e^c_\Omega(h)$.

The PAC model aims to determine the conditions under which it is possible to determine a hypothesis $h$ with a bounded real error and with a definite probability. These conditions are called the conditions of PAC-learnability.

# PAC-learnability

A class of concepts $\mathcal{C}$ is **PAC-learnable** with $\mathcal{H}$, if there exists an algorithm, which for any $0 < \epsilon < 1$ and $0 < \delta < 1$ as a result of working with an oracle EX producing training information, with probability $(1 - \delta)$ will find a hypothesis $h$, for which $e^c_\Omega(h) \leqslant \epsilon$.

Note, that the hypothesis must be found for any given parameters $\epsilon$ and $\delta$, and for any concept $c$ and distribution $\Omega$.

One might think that these conditions are hard to meet and that PAC-learnability will be rare. But we may view this from a different perspective. If we consider a hypothesis $h$ which often gives significant error $(> \epsilon)$, then after testing it on a sufficient number of samples, its fault will be discovered and it could be rejected. On the contrary, if we can come up with a hypothesis consistent with arbitrarily large number of training examples (ie. with the hypothesis error $\leqslant \epsilon$), then probably such hypothesis is correct, and this probability will be larger with a larger number of samples.

So PAC-learnability could be achieved with a competent hypothesis-generation algorithm with a large number of training examples.

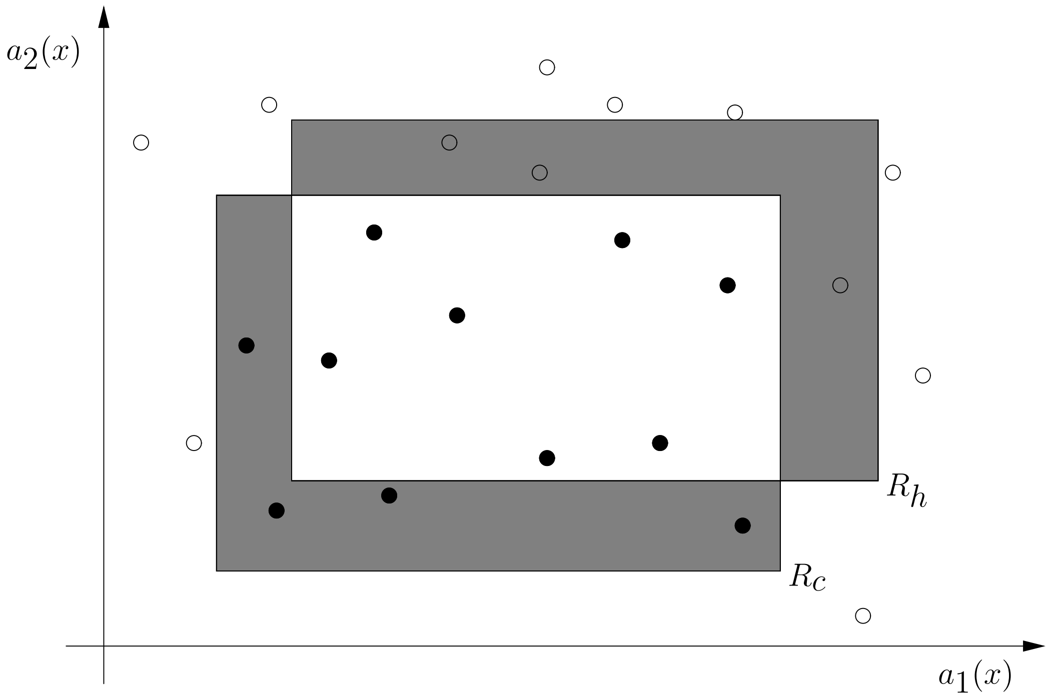
# Effective PAC-learnability

A class of concepts $\mathcal{C}$ is **effectively PAC-learnable** if it is PAC-learnable, and there exists an algorithm of PAC-learning, which works in polynomial time with respect to $1/\delta$, $1/\epsilon$, size of sample from $X$ and size of concept from $\mathcal{C}$.

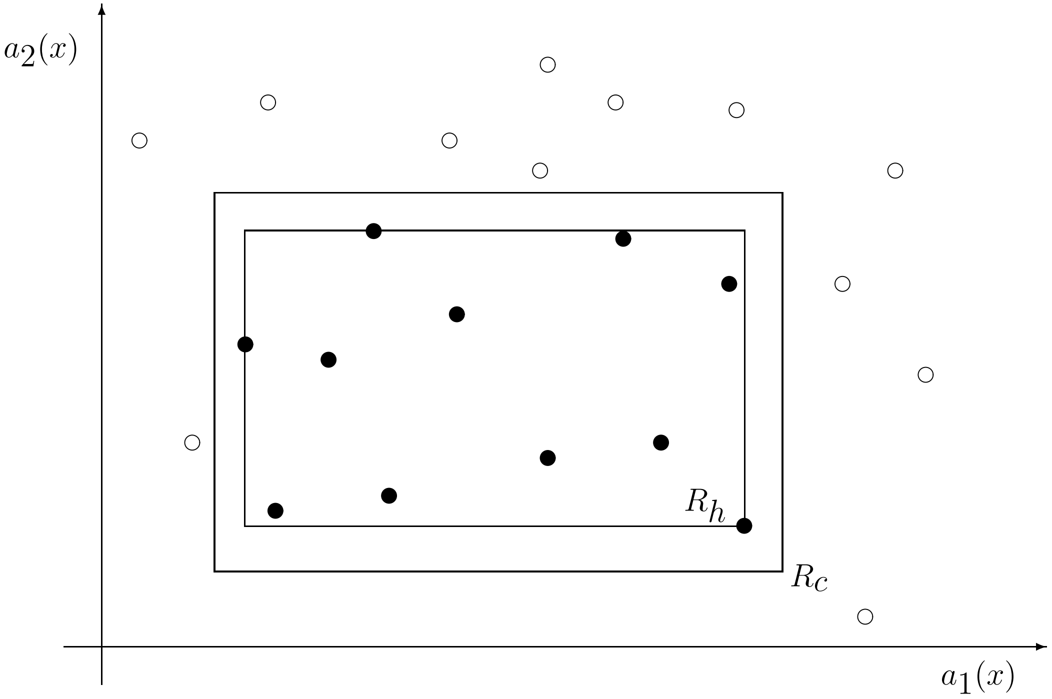
The sizes of samples and concepts are included here as a measure of their complexity which should be related to the computation necessary to process them. The size of the samples could be the number of their attributes. Determining the size of a concept might be more tricky. For example, for the rectangle domain the size could be fixed at 4, since each concept could be represented with four numbers. For the boolean functions example, the concept size could be the number of literals necessary to to write the concept function. Here we will have concepts of larger and smaller sizes.

# PAC-learnability for rectangles

We will now apply the concept of PAC-learnability to the example domain of rectangles. The real error of any given hypothesis $h$ with respect to a given concept $c$ and the probability distribution $\Omega$ is: $e^c_\Omega(h) = \mathsf{Pr}_\Omega(R_c \div R_h)$, where $R_c$ and $R_h$ denote the rectangles corresponding to the concept and the hypothesis, respectively, and $R_c \div R_h = (R_c \setminus R_h) \cup (R_h \setminus R_c)$ denotes the set symmetric difference.



Let us consider the closest fit algorithm, which generates for a given training set $T$ the smallest rectangle enclosing all the positive samples. Then we have $R_h \subseteq R_c$, and so $R_c \div R_h = R_c \setminus R_h$.



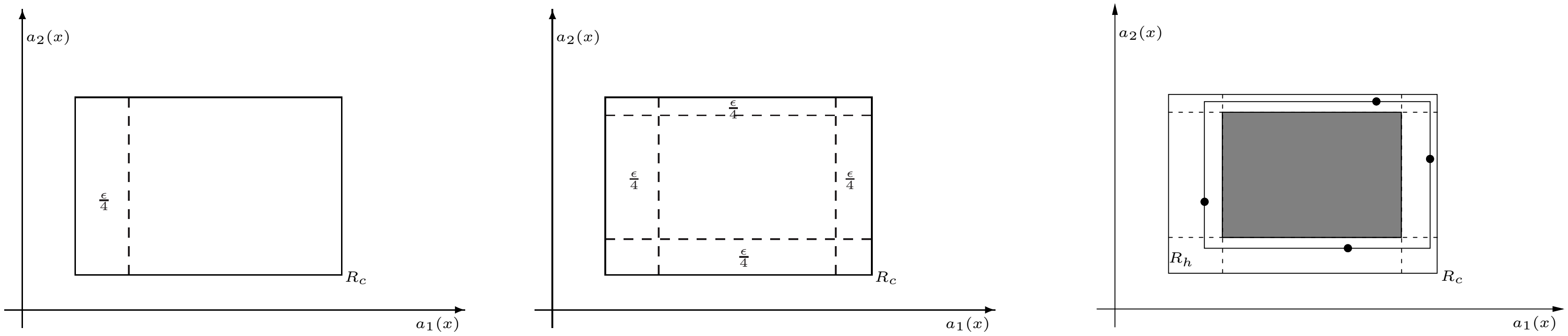
We'd like to relate the value of the real error of the hypothesis with the probability of obtaining this error. What is the probability that a newly generated point within $R_c$ will not be in $R_h$?

First, we can safely assume:

$$\text{Pr}_\Omega(R_c) \geqslant \epsilon$$

(If, on the contrary, $\text{Pr}_\Omega(R_c) < \epsilon$ then it would also be true that $e^c_\Omega(h) = \text{Pr}_\Omega(R_c \setminus R_h) < \epsilon$ so the error would always be below the permitted bound.)

Consider sweeping the left side of the concept rectangle $R_c$ inside so long, until the probability of drawing a point within the rectangle decreases by $\frac{\epsilon}{4}$. It certainly is possible (see the left figure).



Now let us repeat the operation for the remaining three sides of the rectangle $R_c$ (middle figure). For the hypothesis error of $h$ not to exceed $\epsilon$ it is sufficient to ensure, that in all the side bands of the rectangle $R_c$ there will be at least one positive training sample (right figure). Then, with the closest fit algorithm we will have:

$$\text{Pr}_\Omega(R_c \setminus R_h) \leqslant \epsilon$$

How can we ensure, that there will be at least one positive example in each side band of the concept rectangle?

This would not happen, if at least one of the side bands was missed by all of the samples of the training set. The probability of missing one specific side band by the whole training series $T$ is at most: $(1-\frac{\epsilon}{4})^{|T|}$, and the probability of missing any of them is four times higher: $4(1-\frac{\epsilon}{4})^{|T|}$.

We want this probability not to exceed $\delta$:

$$4(1-\frac{\epsilon}{4})^{|T|} \leqslant \delta$$

which by using the inequality $1+\alpha \leqslant e^{\alpha}$ with $\alpha = -\frac{\epsilon}{4}$ can be converted to:

$$|T| \geqslant \frac{4}{\epsilon}(\ln 4 + \ln \frac{1}{\delta})$$

The reasoning presented here proves PAC-learnability of the rectangles domain (using the above tightest fit algorithm), and also provides the minimal number of samples guaranteeing PAC-learnability.

# A general condition for PAC-learnability

It can be proved, that in a general case, if there exists an algorithm generating hypotheses consistent with a sufficiently long training series, then the class of concepts is PAC-learnable.

Specifically, for the number of samples, called the **sample complexity**, given by:

$$N \geqslant \frac{1}{\epsilon}(\ln \frac{1}{\delta} + \ln |\mathcal{H}|)$$

if there exists and algorithm generating hypotheses consistent with this number of samples, for any $\epsilon$ and $\delta$, then with probability at least $1 - \delta$ these hypotheses give error not exceeding $\epsilon$.

Note the above statement says that any hypothesis consistent with a given count of samples satisfies the PAC-learnability condition. Note also, that being consistent with the specified number of samples is a sufficient condition for PAC-learnability.

# The hypothesis space size

The value $|\mathcal{H}|$ in the formula for sample complexity is the size of the hypothesis space, which eg. for the set of boolean function is $2^{2^n}$. So the sample complexity of this space grows as $2^n$. Since the number of the positive samples in this domain is also $2^n$, so achieving PAC-learnability for the class of boolean functions may require examining all, or almost all, examples.

The reason for this is obvious — the hypothesis space contains the hypotheses (boolean functions) which classify arbitrary samples in an arbitrary way. For any number $N$ of samples, the set of hypotheses consistent with them contains equal numbers of such that classify sample $x_{n+1}$ as positive and negative. If we indeed allow any arbitrary boolean functions, then without examining all, or almost all, samples, learning such a function might not be possible.

Efficient learning, which amounts to generalization, will only be possible when the class of boolean functions considered (the hypotheses set $\mathcal{H}$) is significantly restricted. However, restricting this class, with simultaneously satisfying the general condition $\mathcal{C} \subseteq \mathcal{H}$, is only possible using some additional information about the class $\mathcal{C}$.

We could outline several general schemes of using additional information about the class $\mathcal{C}$ to facilitate efficient learning.

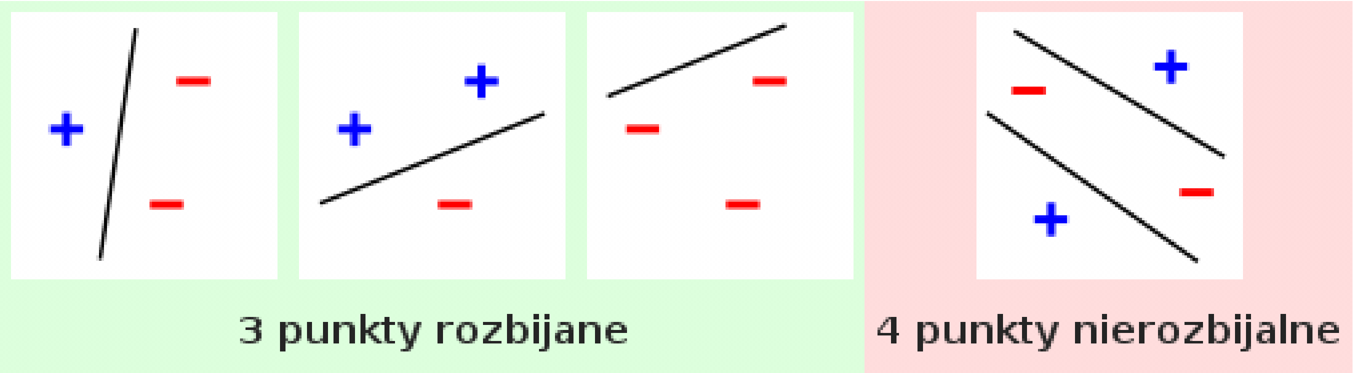
One such scheme is the condition for using only simple hypotheses. For example, in decision trees learning, the algorithm generates the hypothesis which minimizes the entropy.

Another method is to use a restricted hypothesis space which is more tractable, assuming that a hypothesis found this way will be close enough to the true class function.

Still another approach is to use prior knowledge about the problem.

# Shattering the set of samples

If the functions from the hypothesis set $\mathcal{H}$ can divide the set $P$ containing $m$ samples into all $2^m$ ways, independently of the labeling of the samples, then we say that $\mathcal{H}$ **shatters** the set $P$.



As we can see in the above pictures, each set of three points in a plane can be shattered by a straight line, but not for any four points treated as labeled samples there exists a straight line which properly separates them. We can say that the hypotheses set $\mathcal{H}$ consisting of the straight lines in a plane shatters any set of three (non-colinear) points, but does not shatter each set of four points (because there exist such labelings, which cannot be separated).

# The Vapnik-Chervonenkis dimension

We want to estimate the ability of some classifier to learn the classification of various concepts. Some machine learning algorithms are capable of better learning some concepts but unable to learn others.

A class of classifiers $f$ shatters the set of points $(x_1, x_2, ..., x_n)$ if for any labeling of these points there exists a classifier from $f$ which properly classifies the points.

The **Vapnik-Chervonenkis dimension** (VCdim) of a class $f$ is the maximum number of points that $f$ shatters.

For example, the class of classifiers implemented by straight lines in $R^2$ has the VC dimension equal to 3.

The Vapnik-Chervonenkis dimension is a useful measure of the expressiveness of a set of hypotheses. In order for a hypothesis consistent with the training set to have a chance for a good generalization, the cardinality of the training set must significantly exceed the VCdim for a given hypothesis space and a specific problem.

# The relationship of the Vapnik-Chervonenkis dimension to PAC-learnability

Theorem: The hypothesis space $\mathcal{H}$ is PAC-learnable iff it has a finite VCdim dimension.

Theorem: The set of hypotheses $\mathcal{H}$ is properly PAC-learnable if:

1. $m \geq (\frac{1}{\epsilon}) \max[4 \lg(\frac{2}{\delta}), 8 \text{ VCdim } \lg(\frac{13}{\epsilon})]$,

2. there exists an algorithm that generates a hypothesis $h \in \mathcal{H}$ consistent with the training set in polynomial time (relative to $m$ and $n$).