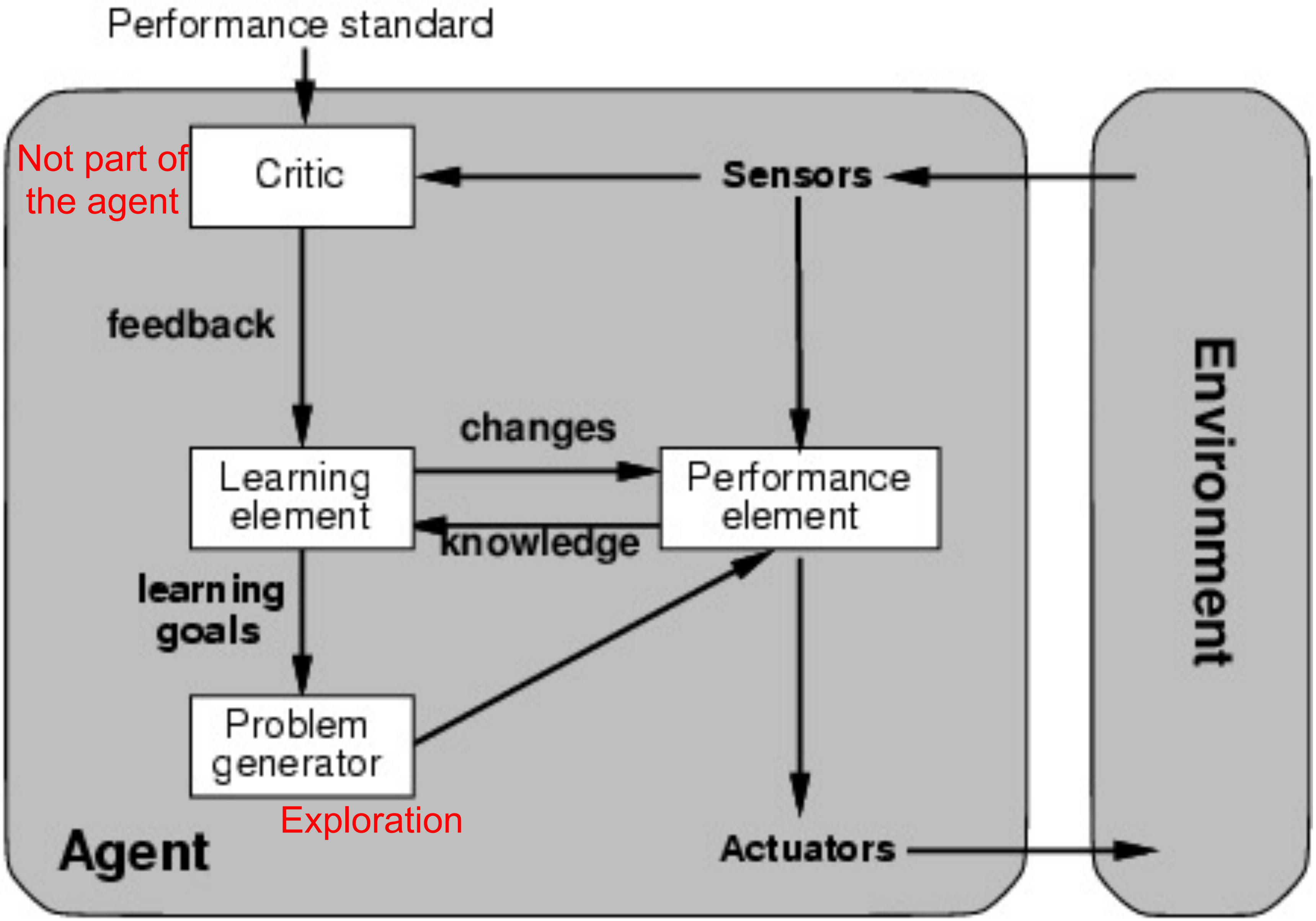
Learning

Learning Agent



Not part of the agent

Exploration

Designing a Learning Agent

- Which components are to be affected by learning?
- What is already known (prior knowledge)?
- How are the components and the data represented:
  - Factored representation: A vector of attributes (numerical or categorical, continuous or discrete)
- What kind of feedback is available, if any?

Types of Learning

- **Unsupervised learning**: The agent needs to organize the object or data with no outside help and no feedback.
  - Clustering: To group objects into meaningful or useful "clusters".
- **Reinforcement leaning**: The available feedback is the outcome (reward or penalty). This can happen long after the decisions or actions that lead to the outcome.
  - Goal of learning: To minimize penalty and maximize reward.
- **Supervised learning**: The correct answers are given by an "teacher" during the learning process.
  - Goal of learning: To generate "correct" answers given the inputs.

Learning from Examples

- A form of supervised learning
- A set of example cases (training samples) and their corresponding correct/desired outputs are given.
- The goal is to form a mechanism that, when given an example case, will produce the correct output.

Example of Learning from Examples

Example problem: Learning to decide whether to wait for a table when arriving at a restaurant:

Should I keep waiting, or should I go to 7-eleven instead?

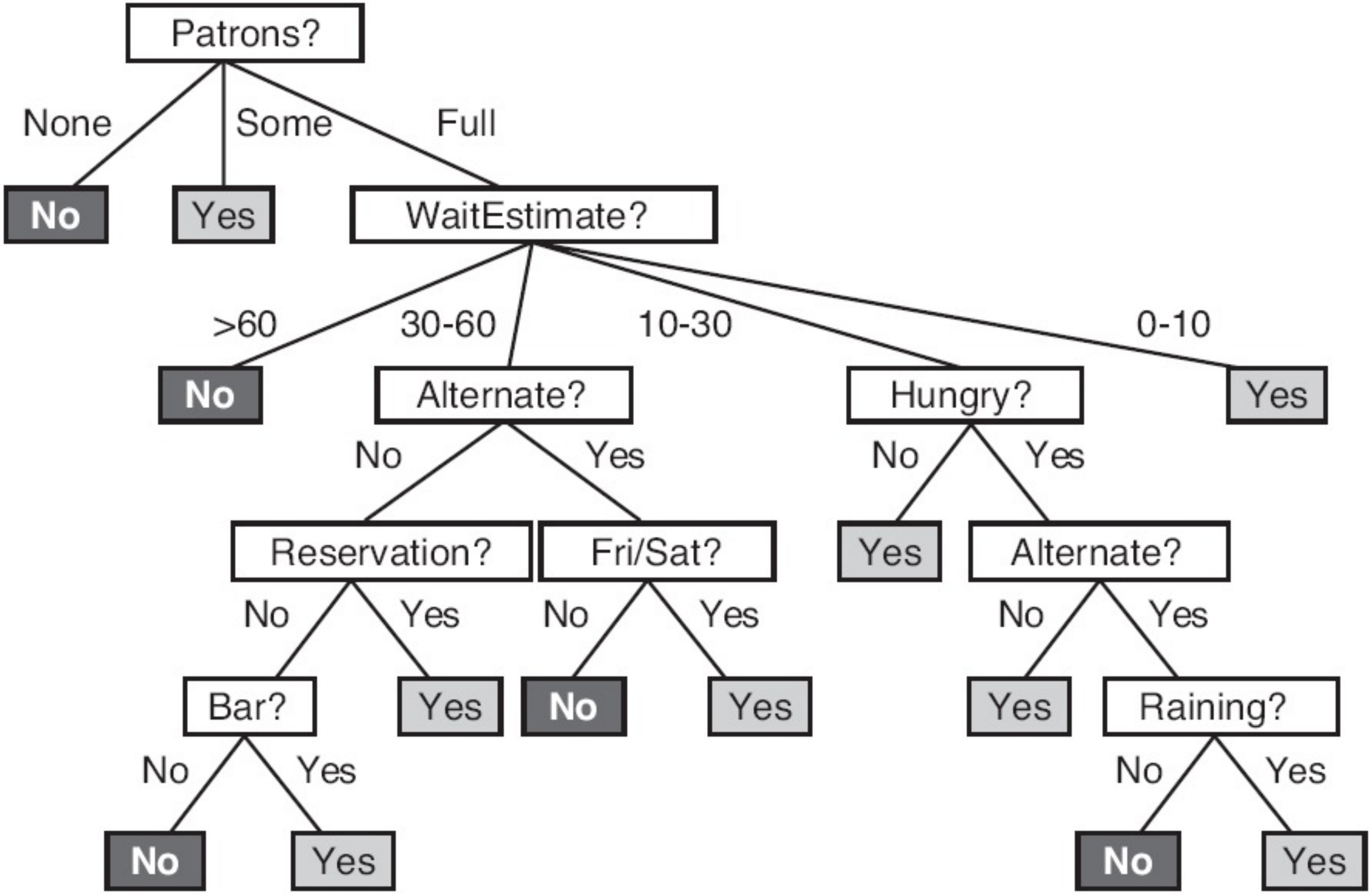
Example of Learning from Examples

The example cases:

Examples	Input Attributes										Output
	Alt.	Bar	Fri/Sat	Hun	Pat.	Price	Rain	Res.	Type	Time	Wait?
x_1	Y	N	N	Y	Some	$$$	N	Y	French	0-10	Y
x_2	Y	N	N	Y	Full	$	N	N	Thai	30-60	N
x_3	N	Y	N	N	Some	$	N	N	Burger	0-10	Y
x_4	Y	N	Y	Y	Full	$	Y	N	Thai	10-30	Y
x_5	Y	N	Y	N	Full	$$$	N	Y	French	>60	N
x_6	N	Y	N	Y	Some	$$	Y	Y	Italian	0-10	Y
x_7	N	Y	N	N	None	$	Y	N	Burger	0-10	N
x_8	N	N	N	Y	Some	$$	Y	Y	Thai	0-10	Y
x_9	N	Y	Y	N	Full	$	Y	N	Burger	>60	N
x_{10}	Y	Y	Y	Y	Full	$$$	N	Y	Italian	10-30	N
x_{11}	N	N	N	N	None	$	N	N	Thai	0-10	N
x_{12}	Y	Y	Y	Y	Full	$	N	N	Burger	30-60	Y

Decision Tree

The input-output relations are organized into a tree. Each node divides its set of samples into subsets according to a particular input attribute.



Supervised Learning Overview

- Main usages of supervised learning:
  - **Classification**: The expected output is one of several categories. Such an agent / program is called a **classifier**.
  - **Regression**: The expected output is a numerical value. Example: temperature in weather forecast.
- Both involve some input/output mapping: $f : \boldsymbol{x} \rightarrow \boldsymbol{y}$

Supervised Learning Overview

Training and testing:

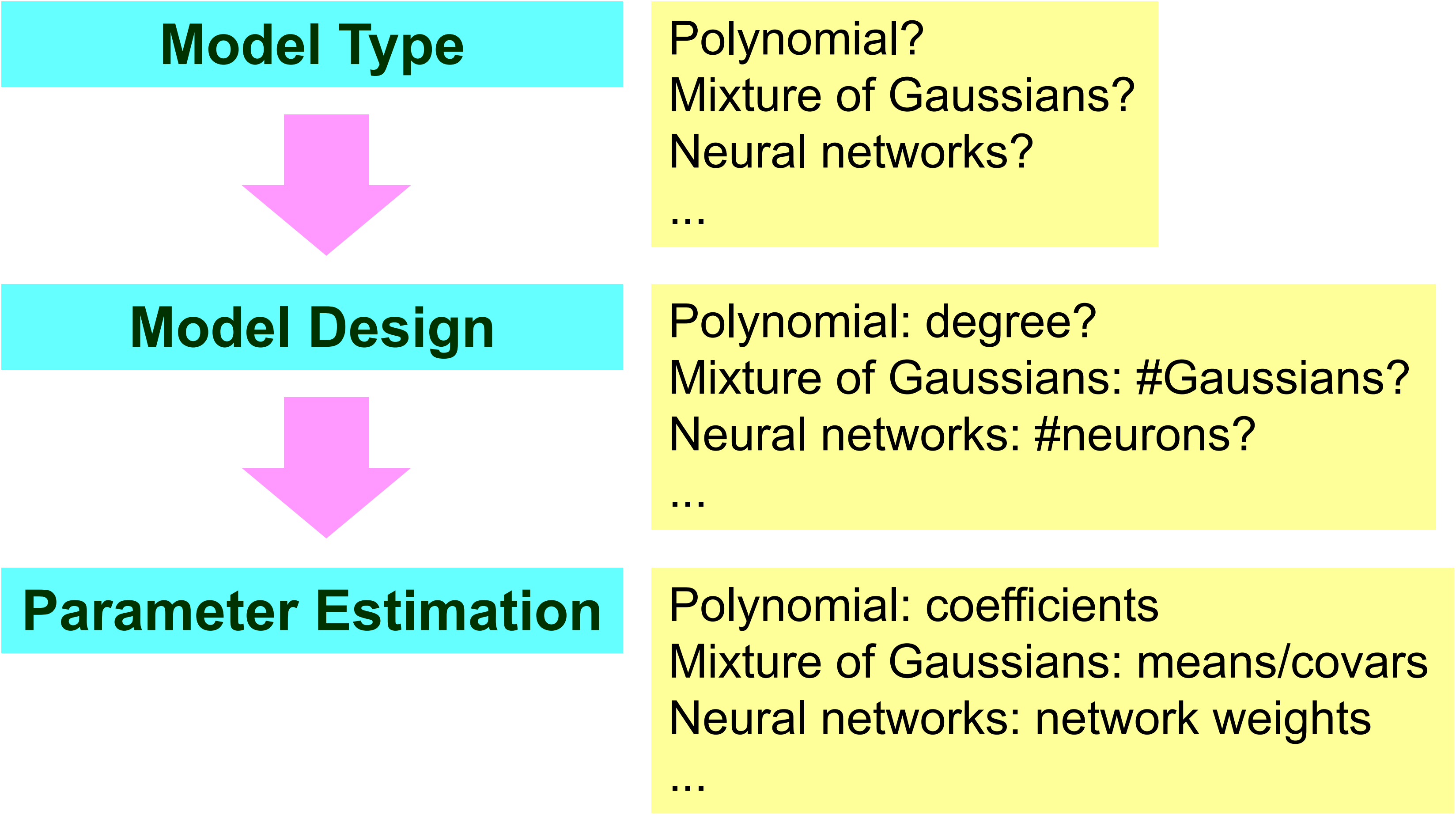
- Training data: Collect samples in the form of $(\boldsymbol{x} \rightarrow \boldsymbol{y})$.
  - For such a pair of $(\boldsymbol{x} \rightarrow \boldsymbol{y})$, $\boldsymbol{y}$ is often called the ground-truth or target output of $\boldsymbol{x}$.
  - If $\boldsymbol{y}$ represents a category (class), then it is also called the class label (or simply the label) of $\boldsymbol{x}$.
- Training: Derive an estimated model of the mapping f using the training data.
- Testing: Using the derive model, determine $\boldsymbol{y}$ for some given $\boldsymbol{x}$ (normally not in the training data).

Supervised Learning Overview

- There are numerous possible forms (model types) to represent the mapping. Examples:
  - Rule Sets
  - Polynomials
  - Gaussian Mixtures
  - k Nearest Neighbors
  - Decision Trees
  - Neural Networks
  - …
- There are many possible forms for representing the mapping of a given set of training data. This means that the actual form used is a design choice.
- For each given form of representation, there are parameters to be determined according to the training data.

Supervised Learning Overview

- Stages of learning a model for $f : x \rightarrow y$:



Model Type

Polynomial?
Mixture of Gaussians?
Neural networks?
...

Model Design

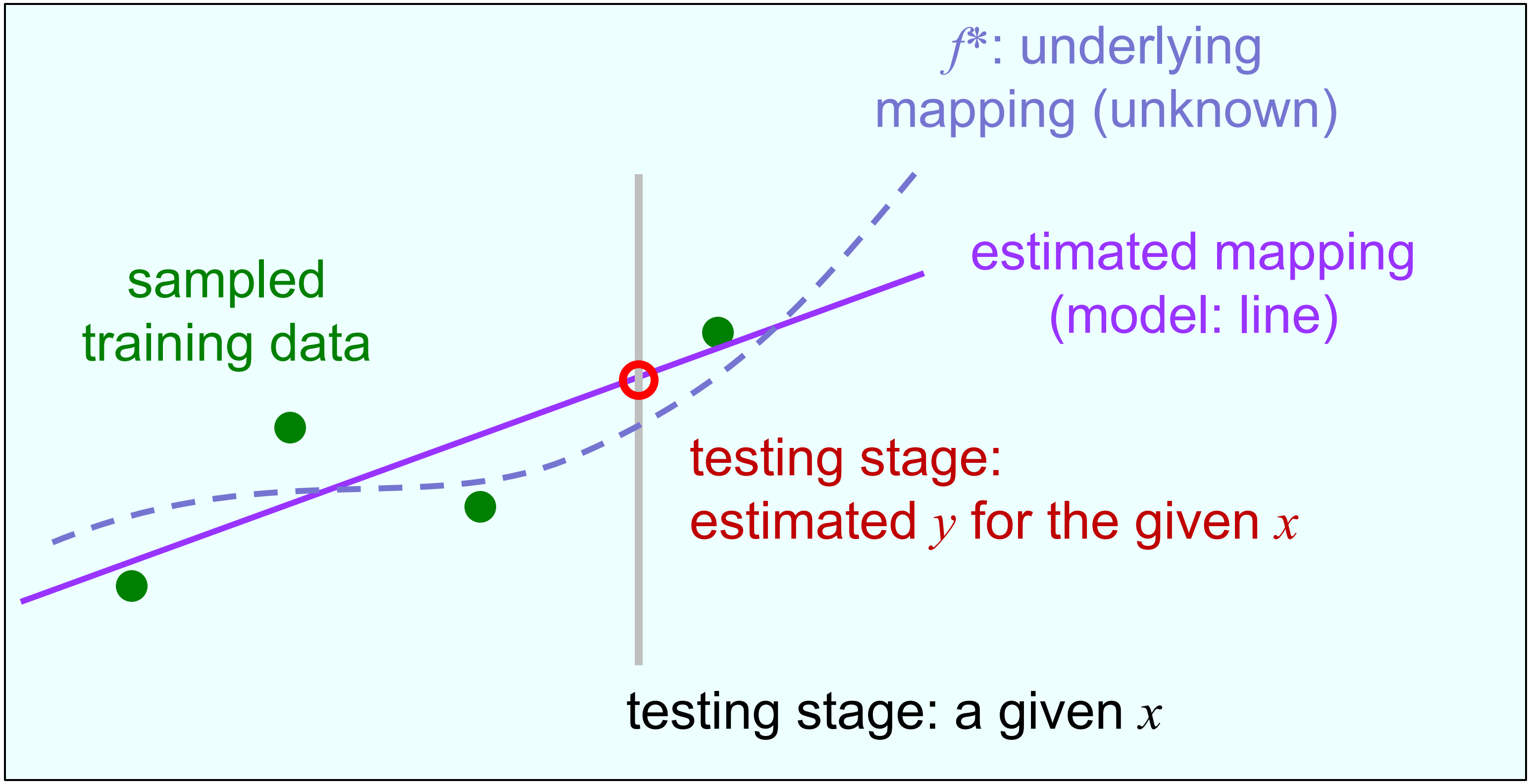
Polynomial: degree?
Mixture of Gaussians: #Gaussians?
Neural networks: #neurons?
...

Parameter Estimation

Polynomial: coefficients
Mixture of Gaussians: means/covars
Neural networks: network weights
...

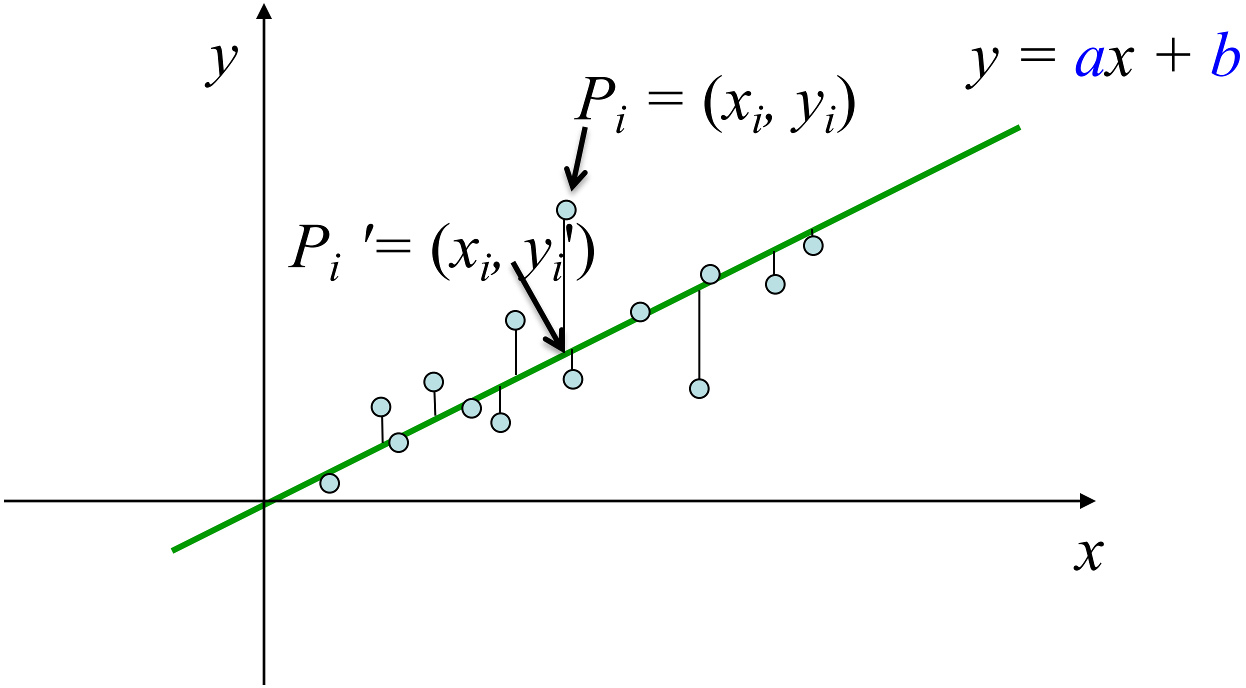
Training and Testing

Here we will use regression as an example.


f*: underlying mapping (unknown)
sampled training data
estimated mapping (model: line)
testing stage: estimated y for the given x
testing stage: a given x

Line fitting

- y-offsets minimization



ML Materials@CS 221 Stanford

- https://stanford-cs221.github.io/autumn2020/modules/