UiO : University of Oslo

An introduction to *machine learning*

Pierre Lison,
Language Technology Group (LTG)
Department of Informatics

HiOA, October 3 2012

Outline

- Motivation
- Machine learning approaches
- My own research
- Conclusion

Motivation

- Certain tasks are extremely difficult to program by hand:

Spam filtering

Motivation

- Certain tasks are extremely difficult to program by hand:

Spam filtering

Face recognition

Motivation

- Certain tasks are extremely difficult to program by hand:

Spam filtering

Face recognition

Machine translation

Motivation

- Certain tasks are extremely difficult to program by hand:

«hi! how are you doing?»

Spam filtering

Face recognition

Machine translation

Speech recognition

Motivation

- Certain tasks are extremely difficult to program by hand:

Spam filtering

Face recognition

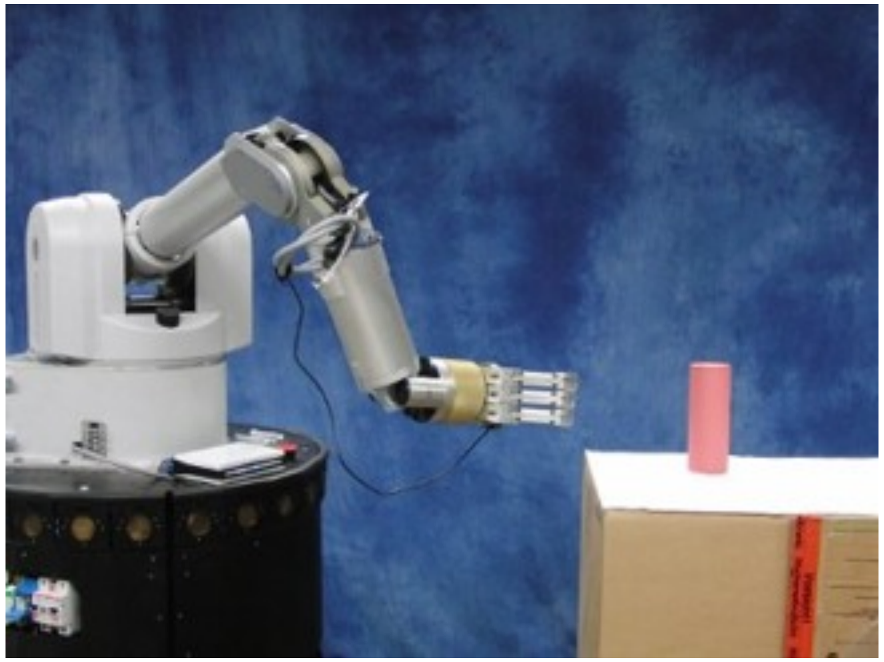
Machine translation

Speech recognition

Data mining

Motivation

- Certain tasks are extremely difficult to program by hand:



Spam filtering

Face recognition

Machine translation

Speech recognition

Data mining

Robot motion

Motivation

- General idea:
  - Collect **data** for our problem
  - Use this data to **learn** how to solve the task
- Key advantages:
  - Can robustly solve complex tasks
  - Reliance on *real-world data* instead of pure intuition
  - Can *adapt* to new situations (collect more data)

Generalities

- Virtually all learning problems can be formulated as (complex) mappings between inputs and outputs
  - We are trying to learn what is the best output **o** to produce for each possible input ***i***
  - Mathematically speaking, we search for a «good» function **F: I → O**, where **I** is the set of possible inputs, and **O** the set of possible outputs

Examples

	Input *i*	Output **o**
Spam filtering	An email	{spam, non-spam}
Face recognition	An image	Identified faces
Machine translation	A sentence in language A	A sentence in language B
Speech recognition	A speech signal	A (text) sentence
Data mining	A financial transaction	{fraud, non-fraud}
Robot motion	Sensory data	Motor control

Learning methods

- But how do we learn this mapping?
- The learning method depends on the kind of data that we have at our disposal
  - We can have examples of data where we have both the inputs and outputs: (***i***,***o***)
  - For some data, we only have the inputs ***i***
  - Sometimes we have no direct access to the «correct» output, but we can get some measure of the quality of an output ***o*** following input ***i***

Learning methods

- But how do we learn this mapping?
- The learning method depends on the kind of data that we have at our disposal
  - We can have examples of data where we have both the inputs and outputs: (***i***,***o***) → supervised learning
  - For some data, we only have the inputs ***i*** → unsupervised learning
  - Sometimes we have no direct access to the «correct» output, but we can get some measure of the quality of an output ***o*** following input ***i*** → reinforcement learning

Supervised learning

- In supervised learning, we have *training data* encoded as pairs (i,o), where the «correct» output is often manually annotated
  - E.g. spam filtering, machine translation, face recognition, etc.
- The function **F: I → O** is often dependent on a (sometimes large) set of parameters
- ... and the learning goal is to «adjust» these parameters in order to fit the data

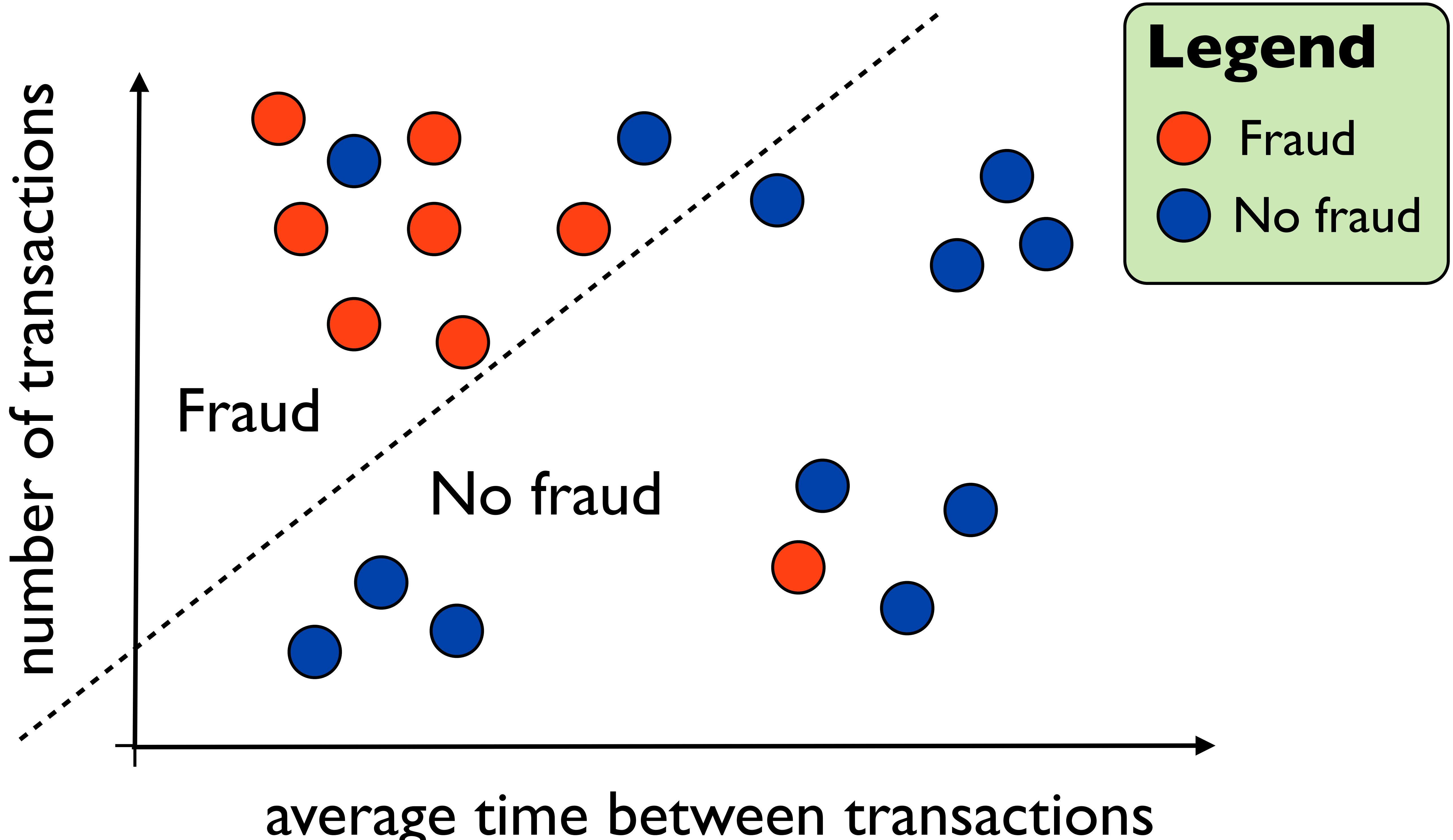
Supervised learning

- A spam filtering system might for instance have «weights» associated to each possible English word
  - The higher the weight, the more it contributes to the probability that the email is a spam
- The learning algorithm will then *adjust* these weights to fit the data

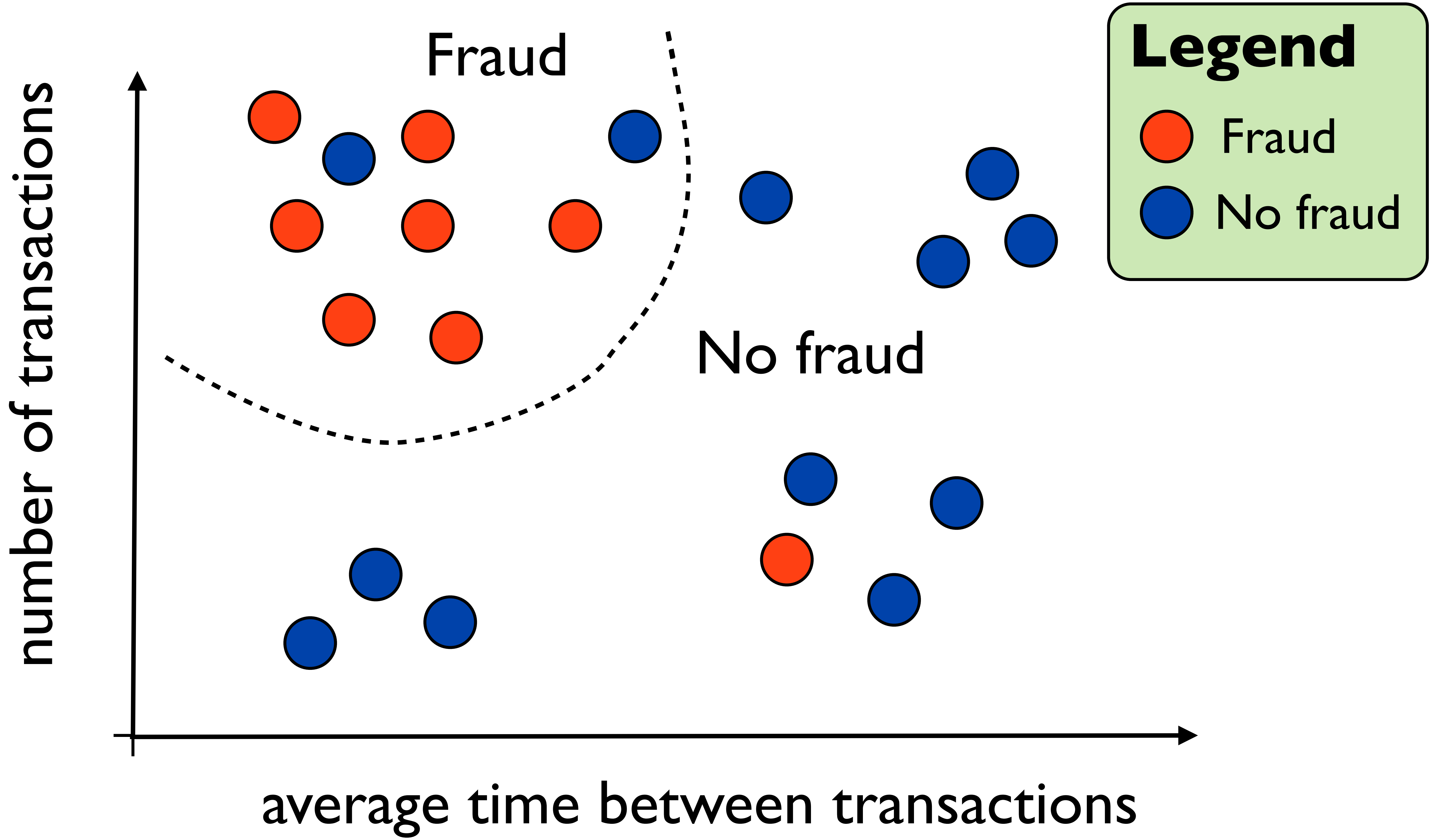
feature of the input, like presence/absence of a word

$$P(\text{email is spam}) \propto \sum_{w_i \in \text{weights}} w_i \boxed{f_i(\mathbf{i})}$$

Supervised learning



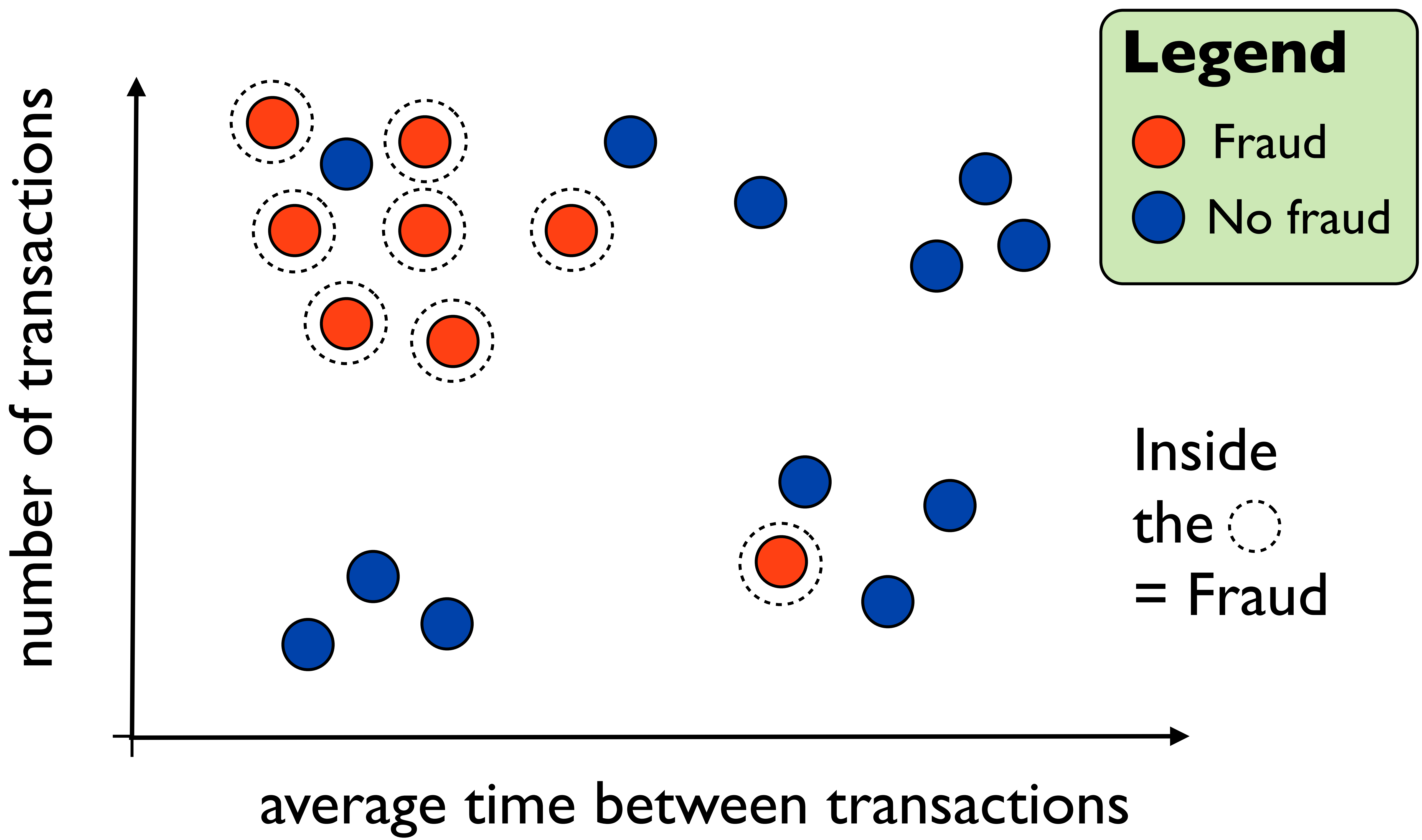
Supervised learning



Supervised learning

- The learning model might be represented in various ways
  - Types of features used to represent the input
  - Varying degrees of complexity
- Which model is the best?
  - First idea: take model with the best fit for the data
  - Problem: some models can be very good at the examples it has seen, but very bad with unseen data

Supervised learning



Generalisation

- A good learning model is a model that **generalises** well to new data
  - In other words, it is able to *abstract* over its experience to detect underlying patterns
  - The design and test of such models is a crucial part of machine learning
- Else, the model is said to be *overfitted*
  - In other words, it is very well «fitted» to the examples it has processed, but perform very poorly with unseen data

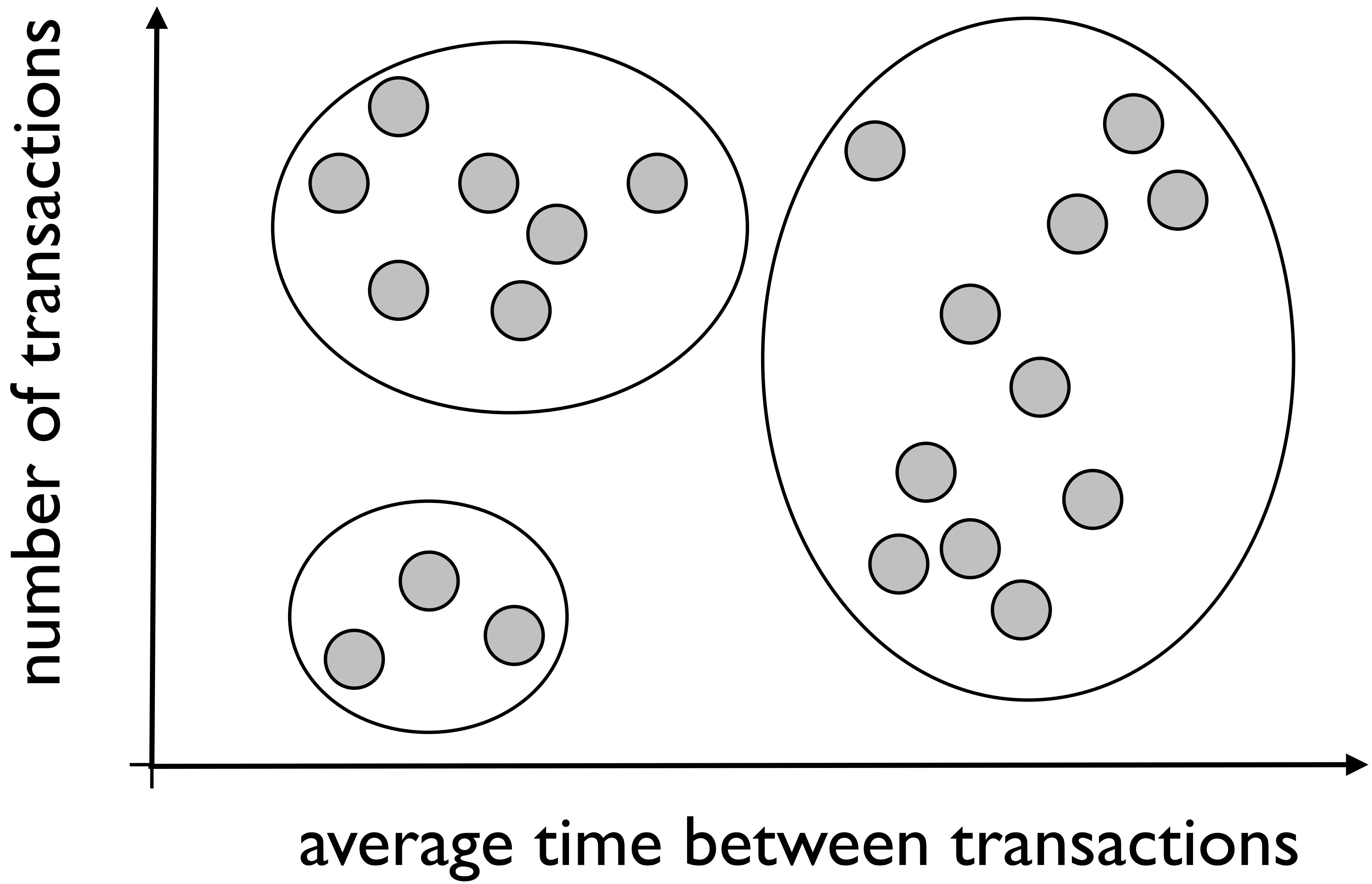
Prior knowledge

- Sometimes, we have some *prior knowledge* about how the model should be
  - Due to intuition, previous data sets, etc.
  - For instance, we can have some prior knowledge about the complexity of the model we should use
- We can integrate this prior knowledge to improve the system accuracy, often with a *Bayesian approach*

Unsupervised learning

- Sometimes, we don't have access to any output value ***o***, we simply have a collection of input examples ***i***
- In this case, what we try to do is to learn the *underlying patterns* of our data
  - is there any *correlations* between features?
  - can we *cluster* our data set in a few groups which behave similarly, and detect *outliers*?

Unsupervised learning



Reinforcement learning

- Finally, the last learning framework is *reinforcement learning*
- In this setting, we don't have direct access to «the» correct output ***o*** for an input ***i***
- But we can get a measure of «how good/ bad» an output is
  - Often called the *reward* (can be negative or positive)

Reinforcement learning

- Reinforcement learning:
  - An *agent* interact with its *environment*

Perception ***i*** → Action ***o*** → Reward ***r***

(state of the world)

repeat

Examples


Robot Motor Skill
Coordination with EM-based
Reinforcement Learning
Petar Kormushev, Sylvain Calinon,
and Darwin G. Caldwell
Italian Institute of Technology

Rewards

- The reward often encodes the *purpose* of the task
  - To learn how to flip pancakes, the reward could for instance be +3 if the pancake is flipped, -1 if the pancake stays in the pan, and -5 if it falls
- The goal of the agent is to learn the behaviour that maximises its *expected cumulative reward* over time

Rewards

- Expected cumulative reward

The agent must try to predict future inputs/rewards

The rewards accumulate over time

- How much worth is a reward expected at time (t+10) compared to one received right now?
  - We usually include a *discount factor* capturing this balance
  - Problem of *delayed gratification*

Learning

- The learning process in reinforcement learning is mostly similar to ones we already seen:
  - We have a model of the world/task, represented with e.g. features associated with parameters
  - Different types of models: some might try to capture all aspects of the environment, while others are purely reactive
- We then let the agent gather *its own experience* in the environment, receiving inputs and trying out actions
  - After each action, the system received a reward
  - The model parameters are then changed accordingly

Links with cognitive science?

- Obvious connections with cognitive and behavioural psychology
  - models often originally inspired by psychological theories
- Many issues in machine learning algorithms are also prevalent in human learning
  - Problem of generalisation / abstraction;
  - Prior knowledge in learning (Bayesian approach);
  - Delayed gratification;

Links with cognitive science?

- But there are important differences as well:
  - A good machine learning model is not necessarily a good model of human cognition (and vice versa)!
  - Role of embodiment, (social) situations, etc.
- But looking at similarities at a *functional* level can yield interesting insights

My own research

- I'm working on *spoken dialogue systems*
  - e.g. systems that can interact with humans using spoken dialogue
  - For instance: talking robots
- I'm using machine learning techniques to make the robot learn how to understand (and use) spoken language

My own research

Verbal interaction is complex (uncertainty, ambiguities, contextual factors, etc.)!

My own research

- I'm more specifically trying to teach the robot *what to say/do* in a given conversational situation
  - Using a mixture of supervised and reinforcement learning
  - Design good rewards for the interaction can be tricky
  - There's also a lot of prior domain knowledge to integrate

Conclusions

- I've described in this talk the major approaches to machine learning:
  - *Supervised learning*: learning from examples
  - *Unsupervised learning*: discovering underlying patterns
  - *Reinforcement learning*: learning a behaviour by interacting in an environment and receiving rewards
- Comparing these approaches to models of human learning can yield useful insights