# Building classifiers that cope with small training sets


Erik Quaeghebeur, Gert de Cooman & Dirk Aeyels

UNIVERSITEIT GENT

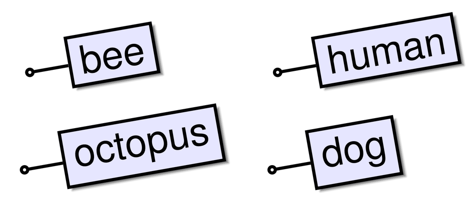
SYSTeMS Research Group
Department of Electrical Energy, Systems & Automation, Ghent University
Technologiepark-Zwijnaarde 914, 9052 Zwijnaarde, Belgium
Erik.Quaeghebeur@UGent.be

FACULTY OF ENGINEERING


## Classification

### Classifying is the act of attaching a class to an object described by some attributes.

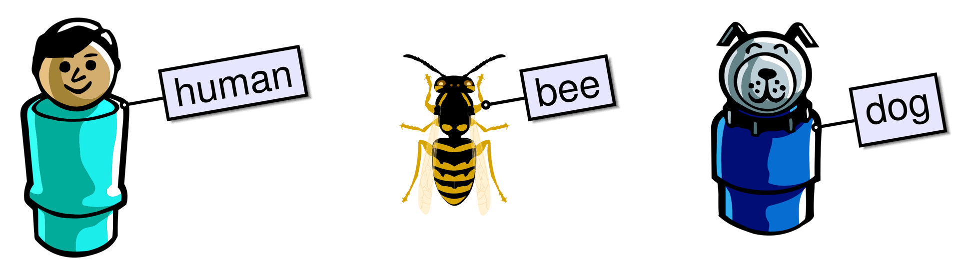
**Classes**




**Attributes**

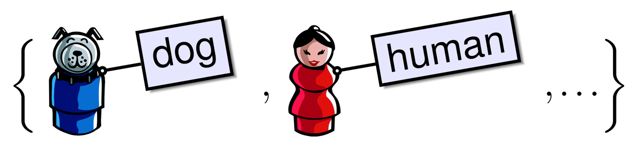
#legs $\in \{0, 4, 6, \geq 8\}$,
hairiness $\in$ {none, light, heavy, complete},
weight $\in \mathbb{R}^+$.

**Classified objects**



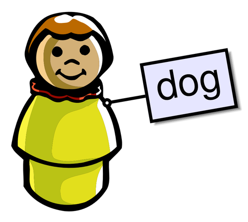

### A classifier automates classification.

To make one, we need a training set of (assumed) correctly pre-classified objects.

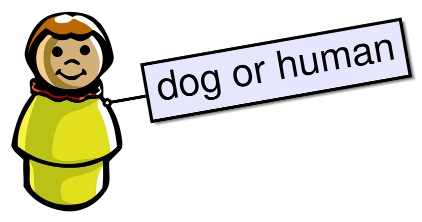



The design will be influenced by prior information about, for example, the distribution of the attribute values (e.g., weight being distributed normally).

The quality of the output of a classifier will depend, among other things, on the size of the training set.




Classical classifiers don't cope well with small training sets (⟵ left), credal classifiers cope by allowing sets of classes to be attached (right ⟶).

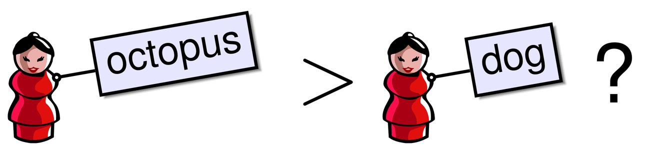



## Classifying using probabilistic models

### Pairwise comparisons can be used to build a classifier.

**Idea**: Compare how good one class fits an object relative to how well another class fits the same object.

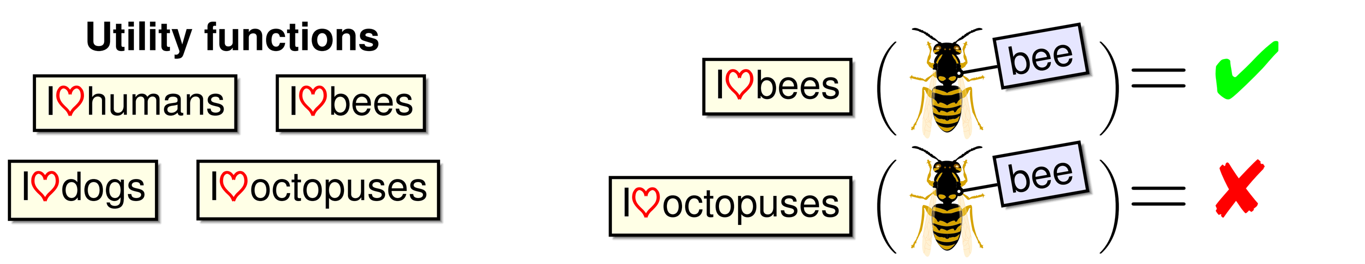
When this is done for all pairs of classes, the best fitting class(es) is (are) attached.



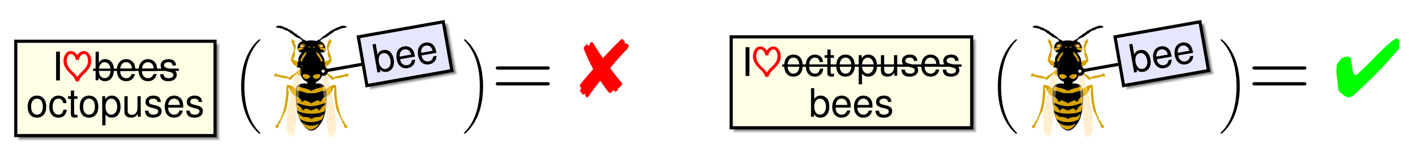

A comparison criterion based on prior information and the training set is needed. For this, we need two building blocks: first utility functions and then conditional lower expectations.

### Utility functions encode the usefulness of attaching a class to an object with a certain (possibly other) class.

**Utility functions**




Prior information determines the choice of utility function. By taking their difference, two such utility functions can be combined into a function that encodes the usefulness of switching from one class to another.



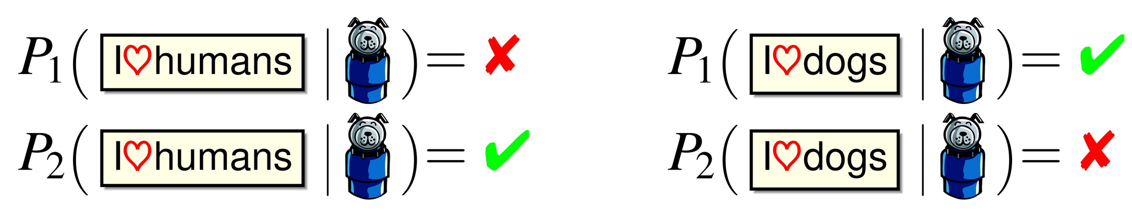

These functions can be used to compare the usefulness of switching from one class to another for an object with a certain class. However, when classifying, we are actually uncertain of its class, so we will work with our expectation of its class.

### For a given object, a conditional lower expectation encodes the uncertainty about its class.

We will give a step-by-step explanation of what a conditional lower expectation is.

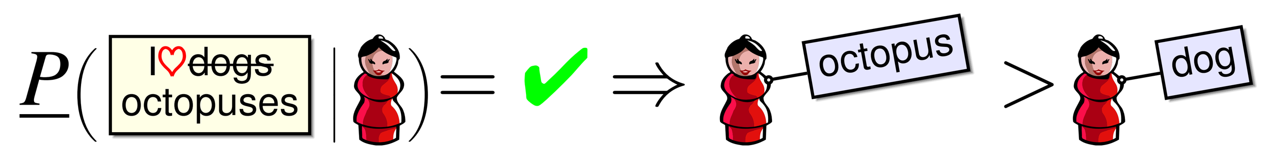
- A (classical) expectation $P$ is an operator that returns the expected value of any function on the set of possible classes. A utility function is an example of such a function. It is based on prior information and the learning set, but not on the object's attributes.


- A conditional expectation returns more detailed information, it does take the object's attributes into account. As an example, two conditional expectations:

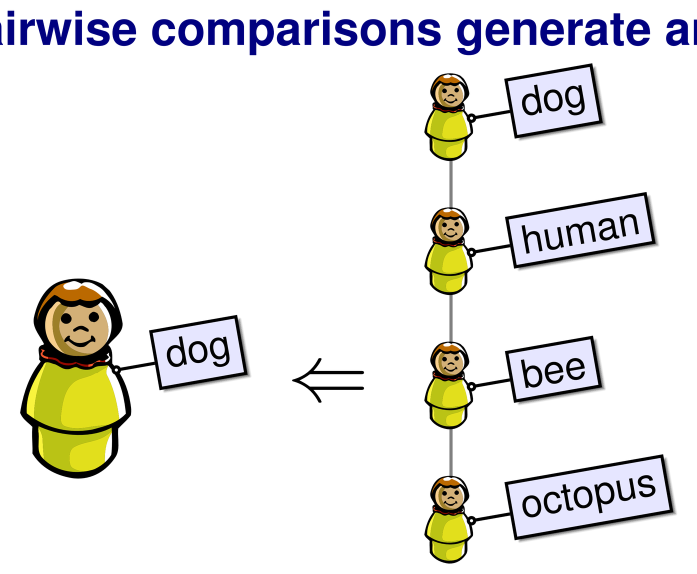



- Often, there is no unique conditional expectation compatible with the prior information and the learning set: a set of compatible conditional expectations is obtained. This set can be written as a conditional lower expectation. It takes the minimum value of the compatible conditional expectations for any function. If the two conditional expectations in the example above are both compatible, we get


### Utility functions and conditional lower expectations can be combined into a criterion for pairwise comparison.

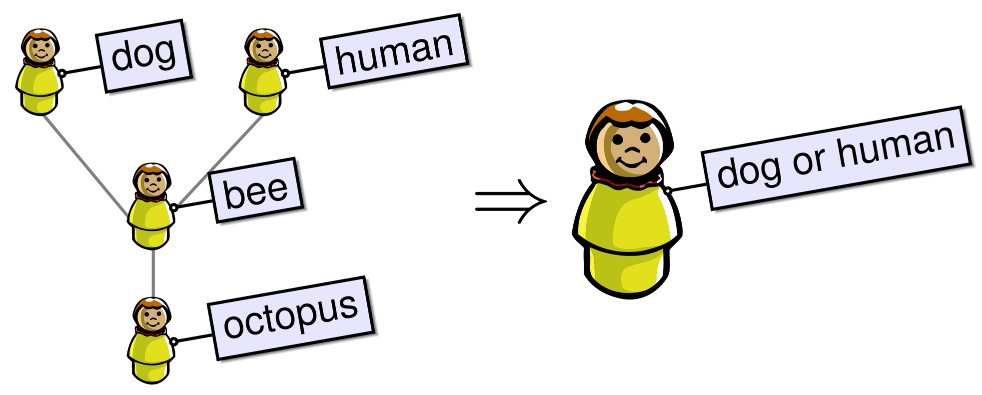



### Pairwise comparisons generate an ordering of the classes; the classifier returns the maximal element(s).




In the criterion, a conditional lower expectation is used instead of a conditional expectation. Otherwise, with multiple compatible conditional expectations, choosing one would lead to a classification that is too precise.

- When using a conditional prevision in the comparison criterion, the resulting ordering would be complete (⟵ left).
- When using a conditional lower prevision, the ordering is partial, which means some classes are incomparable (right ⟶).




## Probabilistic models: a quick glance at the details

### A conditional lower expectation is built using prior information and the learning set

The learning set is seen as a random sample of the set of all objects to classify. It contains information on how likely it is to encounter:

- a certain class (allows a class model to be built),
- a certain set of attributes for a given class (allows an attribute model to be built).

Using imprecise probability theory, the class and attribute models can be combined into the conditional lower expectation used for classifying.

### Our contribution: we designed attribute models that are usable for a wide class of attribute types

The distribution of the attributes (usually part of the prior information) determines, in a large part, the design of the attribute model.

Previously, only models for discrete attributes (e.g., #legs) existed. We designed models for attributes that are distributed according to an exponential family, such as the normal distribution (for, e.g., weight).

These (still) mostly theoretical results are also applicable in other domains.