Bayesian Classification, Inference and Learning in a Probabilistic Type Theory with Records

Staffan Larsson
Centre for Linguistic Theory and Studies in Probability (CLASP)
Dept. of Philosophy, Linguistics and Theory of Science
University of Gothenburg

Joint work with with Jean-Philippe Bernardy and Robin Cooper
(and at earlier stages, also Simon Dobnik and Shalom Lappin)

Outline

Outline

Introduction

- A probabilistic type theory was presented in Cooper *et al.* (2014) and Cooper *et al.* (2015), which extends Cooper's Type Theory with Records
  - (Cooper, 2012a),(Cooper and Ginzburg, 2015), (Cooper, in prep)).
- Non-probabilistic TTR (in common with other type theories) works with judgements of the form $a : T$ ("a is of type T") and assumes that such judgements are categorical.
- In probabilistic TTR (probTTR) we associate probabilities with judgements: $p(a : T)$ ("the probability that a is of type T").

Why probabilistic TTR?

A single framework for modelling

- the *gradience* of semantic judgements, allowing it to serve as the basis for an account of *vagueness* (Fernández and Larsson, 2014).
- semantic and factual *learning*, in a way that can be straightforwardly integrated into more general probabilistic explanations of learning.
- probabilistic *reasoning*, including logical and enthymematic inference (Breitholtz, 2020).
- *grounding language in perception* and the real world, by integrating low-level, sub-symbolic real-valued perceptual information and high-level symbolic information (see e.g. Larsson (2015) and Larsson (2020)).
- *interaction in dialogue*, including interactive learning and reasoning

Outline

TTR: A brief introduction I

- We will be formulating our account in a Type Theory with Records (TTR).
  - We can here only give a brief and partial introduction to TTR; see also Cooper (2005) and Cooper (2012b).
- $a : T$ is a judgment that a is of type T.
- A second kind of judgement (often written Ttrue in Martin-Löf type theory) is the judgement that there is something of type T (T is non-empty).

TTR: A brief introduction II

- Types may be either *basic* or *complex* (in the sense that they are structured objects which have types or other objects introduced in the theory as components).
- One *basic type* in TTR is *Ind*, the type of an individual
- Another basic type is *Real*, the type of real numbers.

TTR: A brief introduction III

- Among the complex types are *ptypes* which are constructed from a predicate and arguments of appropriate types as specified for the predicate.
- Examples are 'man(a)', 'see(a,b)' where $a, b : Ind$.
- The objects or *witnesses* of ptypes can be thought of as situations, states or events in the world which instantiate the type.
- Thus s : man(a) can be glossed as "s is a situation which shows (or proves) that a is a man".

Records and record types

- If
  - $a_1 : T_1$,
  - $a_2 : T_2(a_1)$,
  - $\ldots$,
  - $a_n : T_n(a_1, a_2, \ldots, a_{n-1})$,
  - where $T(a_1, \ldots, a_n)$ represents a type T which depends on the objects $a_1, \ldots, a_n$,
- ...the record to the left is of the record type to the right.

$$\begin{bmatrix} \ell_1 & = & a_1 \\ \ell_2 & = & a_2 \\ \ldots & & \\ \ell_n & = & a_n \\ \ldots & & \end{bmatrix} : \begin{bmatrix} \ell_1 & : & T_1 \\ \ell_2 & : & T_2(l_1) \\ \ldots & & \\ \ell_n & : & T_n(\ell_1, l_2, \ldots, l_{n-1}) \end{bmatrix}$$

- $\ell_1, \ldots \ell_n$ are *labels* which can be used elsewhere to refer to the values associated with them.

Records and record types

- Generic record and record type:

$$\begin{bmatrix} \ell_1 & = & a_1 \\ \ell_2 & = & a_2 \\ \ldots & & \\ \ell_n & = & a_n \\ \ldots & & \end{bmatrix} : \begin{bmatrix} \ell_1 & : & T_1 \\ \ell_2 & : & T_2(l_1) \\ \ldots & & \\ \ell_n & : & T_n(\ell_1, l_2, \ldots, l_{n-1}) \end{bmatrix}$$

- A sample record and record type:

$$\begin{bmatrix} \text{ref} & = & \text{obj}_{123} \\ \text{c}_{\text{man}} & = & \text{prf-man-obj}_{123} \\ \text{c}_{\text{run}} & = & \text{prf-run-obj}_{123} \end{bmatrix} : \begin{bmatrix} \text{ref} & : & \text{Ind} \\ \text{c}_{\text{man}} & : & \text{man(ref)} \\ \text{c}_{\text{run}} & : & \text{run(ref)} \end{bmatrix}$$

- We will introduce further details of TTR as we need them in subsequent sections.

Outline

Probabilistic TTR fundamentals I

- The core of ProbTTR is the notion of probabilistic judgement.
- There are two kinds of judgement corresponding to the two kinds of judgement in non-probabilistic TTR:
    1. A judgement that a situation, s, is of type, T, with some probability. $p(s : T)$ is the probability that s is a witness for T.
    2. a judgement that there is some witness of type T. $p(T)$ is the probability that there is some witness for T.
- This introduces a distinction that is not normally made explicit in the notation used in probability theory.

Probabilistic TTR fundamentals II

- It is useful to have type theoretic objects corresponding to judgements that a situation is of a type.
- Following terminology first introduced in Barwise (1989), we call these *Austinian propositions*.
- A *probabilistic Austinian proposition* is an object (a record) that corresponds to, or encodes, a probabilistic judgement.
- Probabilistic Austinian propositions are records of the type
$$\left[\begin{array}{lll} \text{sit} & : & \mathit{Sit} \\ \text{sit-type} & : & \mathit{Type} \\ \text{prob} & : & [0,1] \end{array}\right]$$
(where $[0, 1]$ represents the type of real numbers between 0 and 1).
- corresponding to / encoding the judgement p(φ.sit:φ.sit-type)= φ.prob

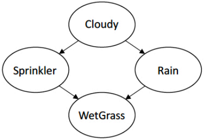
Outline

Bayesian inference I

- Bayesian Networks provide graphical models for probabilistic learning and inference (Pearl, 1990, Halpern, 2003).
- A Bayesian Network is a Directed Acyclic Graph (DAG).
- The nodes of the DAG are random variables
- Its directed edges express dependency relations among the variables.
- The graph describes a complete joint probability distribution (JPD) for its random variables.

Bayesian inference II

P(C=F)	P(C=T)
0.5	0.5



C	P(S=F)	P(S=T)
F	0.5	0.5
T	0.9	0.1

C	P(R=F)	P(R=T)
F	0.8	0.2
T	0.2	0.8

S	R	P(W=F)	P(W=T)
F	F	1.0	0.0
T	F	0.1	0.9
F	T	0.1	0.9
T	T	0.01	0.99

- Russell and Norvig (1995) give the example Bayesian Network above.
- The only directly observable evidence is whether it is cloudy or not, and the queried variable is whether the grass is wet or not.
- Whether it is raining and whether the sprinkler is on is not known, but both of these factors depend on whether it is cloudy, and both affect whether the grass is wet.

Bayesian inference III

- From this Bayesian Network we can compute the marginal probability of the grass being wet ($W = \mathrm{T}$):

$$p(W = \mathrm{T}) = \sum_{s,r,l \in \{T,F\}} p(W = \mathrm{T}, S = s, R = r, C = c)$$

- The Bayesian network allows us to simplify the computation of this JPD by encoding independence relations between variables, so that:

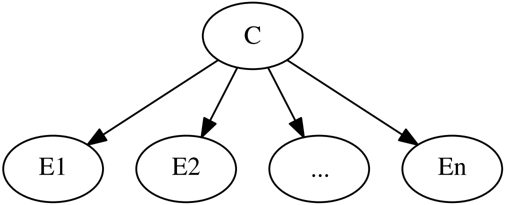
$$p(W, S, R, C) = p(W|S, R)p(S|C)p(R|C)p(C)$$

- and hence $p(W = \mathrm{T}) =$

$$\sum_{s,r,l \in \{T,F\}} p(W = \mathrm{T}|S = s, R = r)p(S = s|C = c)p(R = r|C = c)p(C = c)$$

Naive Bayes classifier I

- A standard Naive Bayes model is a Bayesian network with a single class variable C that influences a set of evidence variables $E_1, \ldots, E_n$ (the evidence), which do not depend on each other.



Naive Bayes classifier II

- A Naive Bayes classifier computes the marginal probability of a class, given the evidence:

$$p(c) = \sum_{e_1,\ldots,e_n} p(c \mid e_1, \ldots, e_n) p(e_1) \ldots p(e_n)$$

where c is the value of C, e_i is the value of E_i $(1 \leq i \leq n)$ and the conditional probability of the class given the evidence is estimated thus:

$$\hat{p}(c \mid e_1, \ldots, e_n) = \frac{p(c)p(e_1 \mid c) \ldots p(e_n \mid c)}{\sum_{C=c'} p(c')p(e_1 \mid c') \ldots p(e_n \mid c')}$$

Outline

Outline

Conditional probabilities in ProbTTR

- We use $p(T_1||T_2)$ to represent the estimated[1] conditional probability that any situation, s, is of type T_1 given that it is of type T_2.
- This contrasts with two other probability judgements in probTTR:
  - $p(s_1 : T_1|s_2 : T_2)$, the probability that a particular situation, s_1, is of type T_1 given that s_2 is of type T_2
  - $p(T_1|T_2)$, the probability that there is a situation of type T_1 given that there is a situation of type T_2.
- In addition there are "mixed" probabilities such as $p(T_1|s : T_2)$, the probability that there is a situation of type T_1 given that $s : T_2$.

[1] Estimating $p(T_1||T_2)$ is part of the learning theory.

Outline

Random variables in TTR I

- To do probabilistic inference in ProbTTR, we need a type theoretic counterpart of a random variable in probabilistic inference.
- Assume a single (discrete) random variable with a range of possible (mutually exclusive) values.
- We introduce a *variable type* $\mathbb{V}$ whose range is a set of *value types* $\mathfrak{R}(\mathbb{V}) = \{A_1, \ldots, A_n\}$ such that the following conditions hold.
  a. $A_j \sqsubseteq \mathbb{V}$ for $1 \leq j \leq n$
  b. $A_j \perp A_i$ for all i, j such that $1 \leq i \neq j \leq n$
  c. for any s, $p(s : \mathbb{V}) \in \{0, 1.0\}$ and $p(s : \mathbb{V}) = \sum_{A \in \mathfrak{R}(\mathbb{V})} p(s : A)$

Random variables in TTR II

a. $A_j \sqsubseteq \mathbb{V}$ for $1 \leq j \leq n$

- (a) says that all value types for a variable type $\mathbb{V}$ are subtypes of $\mathbb{V}$.
  - (A type T_1 is a subtype of type T_2, $T_1 \sqsubseteq T_2$, just in case $a : T_1$ implies $a : T_2$ no matter what we assign to the basic types.)
- A simple way of achieving this is to let $\mathbb{V} = A_i \vee \ldots \vee A_n$.
  - ($T_1 \vee T_2$ is the *join type* of T_1 and T_2. $a : T_1 \vee T_2$ just in case either $a : T_1$ or $a : T_2$).

Random variables in TTR III

b. $A_j \perp A_i$ for all i, j such that $1 \leq i \neq j \leq n$

c. for any s, $p(s : \mathbb{V}) \in \{0, 1.0\}$ and $p(s : \mathbb{V}) = \sum_{A \in \mathfrak{R}(\mathbb{V})} p(s : A)$

- (b) says that all value types for a given variable type V are mutually exclusive, i.e. there are no objects that are of two value types for V.
- (c) says that the probability of a situation s being of a variable type V is either 0 or 1.0.
  - If it is 0 (i.e., the variable has no value for the situation), then the probabilities that s is of each of the value types for $\mathbb{V}$ sum to 0;
  - otherwise these probabilities sum to 1.0.

Random variables in TTR IV

- (c) encodes a conceptual difference between the probability that something has a property (such as colour, $p(s{:}\mathrm{Colour})$), and the probability that it has a certain value of a variable (e.g. $p(s{:}\mathrm{Green})$).
- If the probability distribution over different values (colours) sums to 1.0, then the probability that the object in question has a colour is 1.0.
- The probability that an object has colour is either 0 or 1.0.
- We assume that certain ontological/conceptual type judgements of the form "physical objects have colour" are categorical (which in a probabilistic framework means they have probability 0 or 1.0).

Random variables in TTR V

- Sprinkler example:
- Four binary variable types *Grass*, *Sprinkler*, *Raining* and *Cloudy* with corresponding variable value types:
  $\mathfrak{R}(\mathit{Grass})=\{\mathit{GrassWet},\ \mathit{GrassDry}\}$
  $\mathfrak{R}(\mathit{Sprinkler})=\{\mathit{SprinklerOn},\ \mathit{SprinklerOff}\}$
  $\mathfrak{R}(\mathit{Raining})=\{\mathit{IsRaining},\ \mathit{IsNotRaining}\}$
  $\mathfrak{R}(\mathit{Cloudy})=\{\mathit{ItIsCloudy},\ \mathit{ItIsNotCloudy}\}$
- We assume $\mathit{Grass}=\mathit{GrassWet}\vee\mathit{GrassDry}$, and similarly for the other variable types.
- This ensures
  a. $\mathit{GrassWet}\sqsubseteq\mathit{Grass}$ etc.
  b. $\mathit{GrassWet}\perp\mathit{GrassDry}$ etc.
  c. $p(s : \mathit{Grass}) = p(s : \mathit{GrassWet}) + p(s : \mathit{GrassDry})$

Outline

A ProbTTR Naive Bayes classifier I

- Corresponding to the evidence, class variables, and their values, we associate with a ProbTTR Naive Bayes classifier κ
  a. a collection of m evidence variable types $\mathbb{E}^{\kappa}_1, \ldots, \mathbb{E}^{\kappa}_n$,
  b. associated sets of evidence value types $\mathfrak{R}(\mathbb{E}^{\kappa}_1), \ldots, \mathfrak{R}(\mathbb{E}^{\kappa}_n)$,
  c. a class variable type $\mathbb{C}^{\kappa}$, and
  d. an associated set of class value types $\mathfrak{R}(\mathbb{C}^{\kappa})$.

A ProbTTR Naive Bayes classifier II

- To classify a situation s using a classifier κ, the evidence is acquired by observing and classifying s with respect to the evidence types.
- This can be done through another layer of probabilistic classification based on yet another set of evidence types.
- Type judgements can also be obtained directly from probabilistic or non-probabilistic classification of low-level sensory readings supplied by observation.

A ProbTTR Naive Bayes classifier III

- We define a ProbTTR Bayes classifier κ as a function from a situation s (of the meet type of the evidence variable types $\mathbb{E}_1^\kappa, \ldots, \mathbb{E}_n^\kappa$) to a set of probabilistic Austinian propositions that define a probability distribution over the values of the class variable type $\mathbb{C}^\kappa$, given probability distributions over the values of each evidence variable type $\mathbb{E}_1^\kappa, \ldots, \mathbb{E}_n^\kappa$.

A ProbTTR Naive Bayes classifier IV

- Formally, a ProbTTR Naïve Bayes classifier is a function κ of the type

$$(\mathbb{E}_1^\kappa \wedge \ldots \wedge \mathbb{E}_n^\kappa \rightarrow \mathrm{Set}(\begin{bmatrix} \mathrm{sit} & : & \mathit{Sit} \\ \mathrm{sit\text{-}type} & : & \mathit{Type} \\ \mathrm{prob} & : & [0,1] \end{bmatrix})$$

such that if[2] $s : \mathbb{E}_1^\kappa \wedge \ldots \wedge \mathbb{E}_n^\kappa$, then

$$\kappa(s) = \{\begin{bmatrix} \mathrm{sit} & = & s \\ \mathrm{sit\text{-}type} & = & C \\ \mathrm{prob} & = & p^\kappa(s : C) \end{bmatrix} \mid C \in \mathfrak{R}(\mathbb{C}^\kappa)\}$$

where

$$p^\kappa(s : C) = \sum_{\substack{E_1 \in \mathfrak{R}(E_1^\kappa) \\ \ldots \\ E_n \in \mathfrak{R}(E_n^\kappa)}} p^\kappa(C || E_1 \wedge \ldots \wedge E_n) p(s : E_1) \ldots p(s : E_n)$$

A ProbTTR Naive Bayes classifier V

- ($T_1 \wedge T_2$ is the *meet type* of T_1 and T_2. $a : T_1 \wedge T_2$ just in case $a : T_1$ and $a : T_2$.)
- When using κ, we are interested in the marginal probability $p^{\kappa}(s : C)$ of the situation s being of a class value type C in light of the evidence concerning s.
- As in the case of standard Bayesian Networks, we obtain the marginal probabilities of a class value type C by summing over all combinations of evidence value types.
- The classifier gives a probability distribution over the class value types, encoded as a set of probabilistic Austinian propositions.

A ProbTTR Naive Bayes classifier VI

- As above, for the Naive Bayes classifier we estimate the conditional probability of the class given the evidence using the assumption that the evidence variable types are independent:

$\hat{p}^{\kappa}(C||E_1 \wedge \ldots \wedge E_n) =$

$$\frac{p(C)p(E_1||C)\ldots p(E_n||C)}{\sum_{C' \in \Re(\mathbb{C}^{\kappa})} p(C')p(E_1||C')\ldots p(E_n||C')}$$

[2] Recall that that $\mathbb{E}_1^{\kappa} \ldots \mathbb{E}_n^{\kappa}$ are variable types and that for any variable type V and situation s, $p(s : V) \in \{0, 1.0\}$. Therefore, any type judgement regarding a variable type, such as that involved in the classifier function, can be regarded as categorical.

Outline

Semantic Classification: Example I

- We will now illustrate classification in ProbTTR using a Naive Bayes classifier for fruits.
- We can imagine this classification taking place in the setting of an *Apple Recognition Game*.
- In this game a teacher shows a learning agent fruits (for simplicity, we assume there are only apples and pears in this instance of the game).
- The agent makes a guess, the teacher provides the correct answer, and the agent learns from these observations. (We first describe the classification step.)

Semantic Classification: Example II

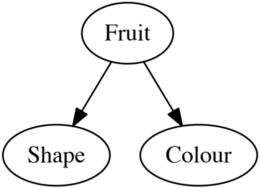
- We will use shorthand for the types corresponding to an object being an apple vs. a pear:
  - $Apple = \begin{bmatrix} \mathrm{x} & : & Ind \\ \mathrm{c_{apple}} & : & \mathrm{apple(x)} \end{bmatrix}$
  - $Pear = \begin{bmatrix} \mathrm{x} & : & Ind \\ \mathrm{c_{pear}} & : & \mathrm{pear(x)} \end{bmatrix}$

Semantic Classification: Example III

- Objects in the Apple Recognition Game have one of two shapes (a-shape or p-shape) and one of two colours (green or red).
  - $Ashape = \begin{bmatrix} \mathrm{x} & : & \mathit{Ind} \\ \mathrm{c} & : & \mathrm{ashape(x)} \end{bmatrix}$
  - $Pshape = \begin{bmatrix} \mathrm{x} & : & \mathit{Ind} \\ \mathrm{c} & : & \mathrm{pshape(x)} \end{bmatrix}$
  - $Green = \begin{bmatrix} \mathrm{x} & : & \mathit{Ind} \\ \mathrm{c} & : & \mathrm{green(x)} \end{bmatrix}$
  - $Red = \begin{bmatrix} \mathrm{x} & : & \mathit{Ind} \\ \mathrm{c} & : & \mathrm{red(x)} \end{bmatrix}$

Semantic Classification: Example IV

- The class variable type is *Fruit*, with value types $\mathfrak{R}(\mathit{Fruit}) = \{\mathit{Apple}, \mathit{Pear}\}$.
- The evidence variable types are
  - *Col*(our), with value types $\mathfrak{R}(\mathit{Col}) = \{\mathit{Green}, \mathit{Red}\}$
  - Shape, with value types $\mathfrak{R}(\mathit{Shape}) = \{\mathit{Ashape}, \mathit{Pshape}\}$.



Classification in the Apple game

- For a situation s the classifier $\text{FruitC}(s)$ returns a set of probabilistic Austinian propositions asserting that s instantiates a certain type of fruit.
- This set is a probability distribution over the variable types of *Fruit*.

$$\text{FruitC}(s) = \left\{ \begin{bmatrix} \text{sit} & = & s \\ \text{sit-type} & = & F \\ \text{prob} & = & p_{\mathcal{J}}^{\text{FruitC}}(s : F) \end{bmatrix} \mid F \in \mathcal{R}(\mathit{Fruit}) \right\}$$

- Probability of a fruit type judgement in the Apple Recognition Game:

$$p^{\text{FruitC}}(s : F) = \sum_{\substack{L \in \mathcal{R}(\text{Col}) \\ S \in \mathcal{R}(\text{Shape})}} p(F || L \wedge S) p(s : L) p(s : S)$$

Classification in the Apple game, cont'd

- To determine the probability that a situation is of the apple type, we sum over the various evidence type values for apple.
- $p^{\text{FruitC}}(s : Apple) =$

$$\sum_{\substack{L \in \mathcal{R}(\text{Col}) \\ S \in \mathcal{R}(\text{Shape})}} p(Apple || L \wedge S) p(s : L) p(s : S) =$$

$$p(Apple || Green \wedge Ashape) p(s : Green) p(s : Ashape) +$$
$$p(Apple || Green \wedge Pshape) p(s : Green) p(s : Pshape) +$$
$$p(Apple || Red \wedge Ashape) p(s : Red) p(s : Ashape) +$$
$$p(Apple || Red \wedge Pshape) p(s : Red) p(s : Pshape)$$

Conditional probabilities used by classifier

- Conditional probabilities for the fruit classifier are derived from previous judgements of the form $p(F||C \wedge S)$
- The example values in the matrix below illustrates a JPD for the apple classifier:

Apple/Pear	*Ashape*	*Pshape*
Green	0.93/0.07	0.63/0.37
Red	0.56/0.44	0.13/0.87

- For example, $p(Apple||Green \wedge Ashape) = 0.93$

Evidence used by the classifier

- The non-conditional probabilities are derived from the agents' take on the particular situation being classified; let's call it s_5.

	T=*Ashape*	T=*Pshape*	T=*Green*	T=*Red*
$\mathrm{p}(\mathrm{s}_5 : T)$	0.90	0.10	0.80	0.20

- We can think of these probabilities as resulting from probabilistic classification of real-valued visual input, where a classifier assigns to each image a probability that the image shows a situation of the respective type.

Classification in the Apple game, cont'd

With these numbers in place, we can compute the probability that the fruit being classified is an apple:

$$
\begin{aligned}
&p^{\mathsf{FruitC}}(\mathrm{s}_5\text{: } \mathit{Apple}) = \\
&\quad p(\mathit{Apple}||\mathit{Green} \wedge \mathit{Ashape})p(s : \mathit{Green})p(s : \mathit{Ashape}) + \\
&\quad p(\mathit{Apple}||\mathit{Green} \wedge \mathit{Pshape})p(s : \mathit{Green})p(s : \mathit{Pshape}) + \\
&\quad p(\mathit{Apple}||\mathit{Red} \wedge \mathit{Ashape})p(s : \mathit{Red})p(s : \mathit{Ashape}) + \\
&\quad p(\mathit{Apple}||\mathit{Red} \wedge \mathit{Pshape})p(s : \mathit{Red})p(s : \mathit{Pshape}) = \\
&\quad 0.93*0.80*0.90+ \\
&\quad 0.63*0.80*0.10+ \\
&\quad 0.56*0.20*0.90+ \\
&\quad 0.13*0.20*0.10= \\
&\quad 0.67+0.05+0.10+0.00= \\
&\quad 0.82
\end{aligned}
$$

Outline

Perceiving evidence I

- We might at this point ask, where do the non-conditional probabilities of the evidence variables concerning the situation s being classified come from?
- We suggest regarding these probabilities as resulting from probabilistic classification of real-valued (non-symbolic) visual input, where a classifier assigns to each image a probability that the image shows a situation of the respective type.
- Such a classifier can be implemented in a number of different ways, e.g. as a deep neural network, as long as it outputs a probability distribution.
- Larsson (2015) shows how perceptual classification can be modelled in TTR, and Larsson (2020) reformulates and extends this formalisation to probabilistic classification.

Perceiving evidence II

- Adapting the notion of a probabilistic TTR classifier to the current setting:
- a probabilistic perceptual (here, visual) classifier corresponding to an evidence value type $E_i (1 \leq i \leq n)$ provides a mapping
  - from perceptual input (of a type $\mathfrak{V}$, e.g. a digital image)
  - onto a probability distribution over evidence value types in $\mathfrak{R}(E_i^\kappa)$,
- encoded as a set of probabilistic Austinian propositions:

$$\pi_{E_i^\kappa} : Sit_{\mathfrak{V}} \rightarrow \{ \begin{bmatrix} \text{sit} & : & Sit_{\mathfrak{V}} \\ \text{sit-type} & : & RecType_R \\ \text{prob} & : & [0,1] \end{bmatrix} \mid R \in \mathfrak{R}(E_i^\kappa) \}$$

where $Sit_{\mathfrak{V}}$ is the type of situations where perception of some object (labelled x) yields visual information (labelled c) concerning x:

$$Sit_{\mathfrak{V}} = \begin{bmatrix} \text{x} & : & Ind \\ \text{c} & : & \mathfrak{V} \end{bmatrix}$$

Perceiving evidence III

- $RecType_R$ is the (singleton) type of record types that are identical to R, so that e.g. $T{:}RecType_{Green}$ iff $T{:}RecType$ and $T = Green$
- In the Apple game, an agent would be equipped with visual classifiers corresponding to *Shape* and *Col*, where e.g.

$$\pi_{Col} : \begin{bmatrix} \text{x} & : & Ind \\ \text{c} & : & \mathfrak{V} \end{bmatrix} \rightarrow$$
$$\left\{ \begin{bmatrix} \text{sit} & : & Sit_{\mathfrak{V}} \\ \text{sit-type} & : & RecType_{Green} \\ \text{prob} & : & [0,1] \end{bmatrix}, \begin{bmatrix} \text{sit} & : & Sit_{\mathfrak{V}} \\ \text{sit-type} & : & RecType_{Red} \\ \text{prob} & : & [0,1] \end{bmatrix} \right\}$$

Perceiving evidence IV

- If we e.g. assume $s_5 = \begin{bmatrix} \text{x} & = & \text{a}_{453} \\ \text{c} & = & \text{Img}_{9876} \end{bmatrix}$ where
  - a_{453}:*Ind*
  - $\text{Img}_{9876}:\mathfrak{V}$

  and we assume that

  $$\pi_{Col}(s_5) = \left\{ \begin{bmatrix} \text{sit} & = & s_5 \\ \text{sit-type} & = & \mathit{Green} \\ \text{prob} & = & 0.8 \end{bmatrix}, \begin{bmatrix} \text{sit} & = & s_5 \\ \text{sit-type} & = & \mathit{Red} \\ \text{prob} & = & 0.2 \end{bmatrix} \right\}$$

  then
  - $\text{p}(s_5:\mathit{Green})=0.8$
  - $\text{p}(s_5:\mathit{Red})=0.2$

  which, incidentally, are the probabilities also used above.
- This illustrates how ProbTTR allows combining probabilistic perceptual classification and probabilistic reasoning.

Outline

Bayesian networks in TTR I

- To extend the above to full Bayesian networks, we need to distinguish evidence variables from *unobserved variables*, and incorporate the latter into our classifier.
- A TTR Bayes net classifier is associated with
  - $\mathbb{E}^{\kappa}_1, \ldots, \mathbb{E}^{\kappa}_n$ is a collection of evidence variable types,
  - $\mathfrak{R}(\mathbb{E}^{\kappa}_1), \ldots, \mathfrak{R}(\mathbb{E}^{\kappa}_n)$ are sets of evidence value types,
  - $\mathbb{I}^{\kappa}_1, \ldots, \mathbb{I}^{\kappa}_m$ is a collection of unobserved variable types,
  - $\mathfrak{R}(\mathbb{I}^{\kappa}_1), \ldots, \mathfrak{R}(\mathbb{I}^{\kappa}_m)$ are sets of unobserved value types.

Bayesian networks in TTR II

- Given this, a TTR Bayes net classifier is a function κ of type

$$\mathbb{E}_1^\kappa \wedge \ldots \wedge \mathbb{E}_n^\kappa \rightarrow \mathrm{Set}(\left[\begin{array}{lcl} \mathrm{sit} & : & \mathit{Sit} \\ \mathrm{sit\text{-}type} & : & \mathit{Type} \\ \mathrm{prob} & : & [0{,}1] \end{array}\right])$$

such that if $s : \mathbb{E}_1^\kappa \wedge \ldots \wedge \mathbb{E}_n^\kappa$ and $1 \leq j \leq m$, then

$$\kappa(s) = \{\left[\begin{array}{lcl} \mathrm{sit} & = & s \\ \mathrm{sit\text{-}type} & = & I_j \\ \mathrm{prob} & = & p^\kappa(s : I_j) \end{array}\right] \mid I_j \in \mathfrak{R}(\mathbb{I}_j^\kappa)\}$$

where

$$p^\kappa(s : I_j) = \sum_{\substack{I_1 \in \mathfrak{R}(\mathbb{I}_1^\kappa) \\ \ldots \\ I_{j-1} \in \mathfrak{R}(\mathbb{I}_{j-1}^\kappa) \\ I_{j+1} \in \mathfrak{R}(\mathbb{I}_{j+1}^\kappa) \\ \ldots \\ I_m \in \mathfrak{R}(\mathbb{I}_m^\kappa) \\ E_1 \in \mathfrak{R}(\mathbb{E}_1^\kappa) \\ \ldots \\ E_n \in \mathfrak{R}(\mathbb{E}_n^\kappa)}} p(I_j || I_1 \wedge \ldots \wedge I_{j-1} \wedge I_{j+1} \wedge \ldots \wedge I_m \wedge E_1 \wedge \ldots \wedge E_n) p(s : E_1) \ldots p(s : E_n)$$

Bayesian networks in TTR III

- The dependencies encoded in a Bayes net will affect how the conditional probability

  $$p(C||I_1 \wedge \ldots I_{j-1} \wedge I_{j+1} \wedge I_m \wedge E_1 \wedge \ldots \wedge E_n)$$

  is computed.
- In the sprinkler example, we have three unobserved variable types *Grass*, *Sprinkler* and *Rain*, and one evidence variable type *Cloudy*.

Bayesian networks in TTR IV

- For $S \in \mathfrak{R}(\mathit{Sprinkler}), R \in \mathfrak{R}(\mathit{Rain}), L \in \mathfrak{R}(\mathit{Cloudy})$ and $G \in \mathfrak{R}(\mathit{Grass})$, the dependencies encoded in the Bayesian network above entail that $p(G||S \wedge R \wedge L) =$

$$p(G||S \wedge R)p(S||L)p(R||L)$$

and hence for $G \in \mathfrak{R}(\mathit{Grass})$,

$$p^{\kappa}(s : G) = \sum_{\substack{S \in \mathfrak{R}(\mathit{Sprinkler}) \\ R \in \mathfrak{R}(\mathit{Raining}) \\ L \in \mathfrak{R}(\mathit{Cloudy})}} p(G||S \wedge R)p(S||L)p(R||L)p(s : L)$$

Outline

Semantic learning I

- A central question is then how we get conditional probabilities of the from $p(C||E_1 \wedge \ldots \wedge E_n)$ (where $C \in \mathfrak{R}(\mathbb{C}), E_i \in \mathfrak{R}(\mathbb{E}_i), 1 \leq i \leq n$).
$\hat{p}^{\kappa}(C||E_1 \wedge \ldots \wedge E_n) =$

$$\frac{p(C)p(E_1||C)\ldots p(E_n||C)}{\sum_{C' \in \mathfrak{R}(\mathbb{C}^{\kappa})} p(C')p(E_1||C')\ldots p(E_n||C')}$$

- We need to estimate the conditional probabilities $p(E_i||C)$ and priors $p(C)$
- This is the role of the semantic learning component.
- There are several ways to approach this problem.

Outline

A frequentist approach to semantic learning I

- In standard probability theory, conditional probabilities can be estimated by counting previous instances of C and E_i:

$$p(E_i|C) = \frac{|E_i \& C|}{|C|}$$

- This relies on previous judgements being categorical rather than probabilistic.
- However, it appears reasonable to assume that agents sometimes make non-categorical judgements, assigning a probability other than 0 or 1 to a situation being of a certain type
- We want to explore the idea of using such non-categorical past judgements as a basis for future (probabilistic) judgements.

A frequentist approach to semantic learning II

- In Cooper *et al.* (2015), a solution with a frequentist flavour (but also with some differences to regular frequentist learning acccounts) is sketched, based on the idea that an agent makes judgements based on a finite string of probabilistic Austinian propositions, the *judgement history* $\mathfrak{J}$.
- When an agent A encounters a new situation s and wants to know if it is of type T or not, A uses probabilistic reasoning to determine $p(s : T)$ on the basis of A's previous judgements $\mathfrak{J}$.
- We expand on this sketch here.

A frequentist approach to semantic learning III

- So the history of judgements $\mathfrak{J}$ does not contain definite judgements, but rather probabilistic ones.
- How are these probabilities to be understood?
- We assume that each such probability corresponds to the (frequentist) *probability that a member of the linguistic community would judge s to be of type T*.
- Hence, each probabilistic judgement in the history can be considered to correspond to a large number N of independent categorical judgements.

A frequentist approach to semantic learning IV

- How do we motivate this?
- After all, language is categorical in nature at least insofar as a speaker makes or does not make an utterance U to describe some situation s, thus categorising s as (categorically) correctly described by U.
  - Put differently, it may be argued that one cannot make an utterance only to a certain degree, but that one either makes or does not make the utterance.
- Main answer: the categorical nature of language does not imply that agents cannot entertain non-categorical judgements, only that once they speak their judgements, the become categorical.

A frequentist approach to semantic learning V

- Complication 1: Natural languages have modifiers such as "possibly" or "sort of" indicating degrees of confidence in embedded propositions
  - But even utterances containing such modifiers are either made or not made (categorically).
  - The modifications only come into play when one wants to derive $s : P$ from s:possibly(P).
- Complication 2: hearers may assign probabilities to speakers having made an utterance U based on perceptual, semantic and pragmatic confidences. We ignore this here.

Computing conditional probabilities I

- When it comes to computing the probabilities needed for probabilistic classifiers, this means that $p(s : C)p(s : E_i)N$ of them are considered to be of type $C \wedge E_i$.
- On this basis, we can compute likelihoods and probabilities as ratio of the frequencies of occurrences, summed over all judgements in the history:

$$p(E_i||C) = \frac{\sum_{j\in\mathfrak{J}, j.\mathrm{sit}=s} p(s : C)p(s : E_i)N}{\sum_{j\in\mathfrak{J}, j.\mathrm{sit}=s} p(s : C)N}$$
$$= \frac{\sum_{j\in\mathfrak{J}, j.\mathrm{sit}=s} p(s : C)p(s : E_i)}{\sum_{j\in\mathfrak{J}, j.\mathrm{sit}=s} p(s : C)}$$

where $p(\varphi.\mathrm{sit}{:}\varphi.\mathrm{sit\text{-}type}) = \varphi.\mathrm{prob}$

Computing conditional probabilities II

- The above formula tells us that we can consider probabilities in the history of judgments as fractions of events
- This is justified by interpreting them as fractions of language-community speakers making the corresponding categorical judgement.
- In this sense, we are providing a frequentist interpretation of the conditional probability.

Computing conditional probabilities III

- For the purpouses of this paper, we will assume that the probabilities needed are indeed encoded directly in $\mathfrak{J}$, but of course in the general case this might not be the case.
- Cooper *et al.* (2015) explains how probabilities of complex types (such as meet types, join types, function types and record types) can be computed from simpler types.

Computing priors I

- In addition to conditional probabilities, we need the prior probabilies of the class value types $C \in \mathfrak{R}(\mathbb{C})$ and evidence value types $E_i \in \mathfrak{R}(\mathbb{E}_i)$.
- $\mathfrak{p}_{\mathfrak{J}}(T)$ represents the prior probability that an arbitrary situation is of type T given $\mathfrak{J}$.

$$\mathfrak{p}_{\mathfrak{J}}(T) = \frac{\sum_{j \in \mathfrak{J}_T} j.\text{prob}}{\mathrm{P}(\mathfrak{J})} \text{ if } \mathrm{P}(\mathfrak{J}) > 0, \text{ otherwise } 0$$

where $\mathfrak{J}_T$ is the set of all judgements concerning T:

$$\mathfrak{J}_T = \{j \mid j \in \mathfrak{J}, j.\text{sit-type} = T\}$$

and $\mathrm{P}(\mathfrak{J})$ is the cardinality of situations in $\mathfrak{J}$, i.e. the total number of situations in J^3:

$$\mathrm{P}(\mathfrak{J}) = |\{s | \exists j \in \mathfrak{J}, j.\text{sit} = s\}|$$

Computing priors II

- It is important to note that the prior probability for a value type A is not the same as the probility $p(A)$ that there is something of type A.
- To see this, imagine that $p(s_1 : A) = 0.8$ and $p(s_2 : A) = 0.2$, and that there are no judgements concerning other situations in J.
- In this case, $p(A)$, the probability that there is something of type A is 0.8.
- However, $\mathfrak{p}_{\mathfrak{J}}(A)$ is $(0.8 + 0.2)/2 = 0.5$.

Computing priors III

- We will use the priors (rather than probabilities of types) when estimating the JPD required by the classifier, so actually $\hat{p}^{\kappa}(C||E_1 \wedge \ldots \wedge E_n) =$

$$\frac{\mathfrak{p}_{\mathfrak{I}}(C)p(E_1||C)\ldots p(E_n||C)}{\sum_{C' \in \mathfrak{R}(\mathbb{C}^{\kappa})} \mathfrak{p}_{\mathfrak{I}}(C')p(E_1||C')\ldots p(E_n||C')}$$

[3] This replaces an earlier definition in Cooper *et al.* (2015).

Outline

Frequentist Semantic Learning: Example I

- Assume that $\mathfrak{J}$ is as follows, based on previous rounds of the game.

j.p	j.sit-type$\in \mathfrak{R}(Fruit)$		j.sit-type$\in \mathfrak{R}(Col)$		j.sit-type$\in \mathfrak{R}(Shape)$	
j.sit	*Apple*	*Pear*	*Green*	*Red*	*Ashape*	*Pshape*
s_1	1.0	0.0	0.9	0.1	0.7	0.3
s_2	0.5	0.5	0.7	0.3	0.6	0.4
s_3	0.9	0.1	1.0	0.0	1.0	0.0
s_4	0.1	0.9	0.0	1.0	0.0	1.0

- The recorded judgements concerning the types *Apple* and *Pear* are here assumed to be derived not only from the agent's own perception of the fruits in question...
- ...but also (and perhaps primarily) from a tutor's explicit judgements, possibly in combination with an estimation of the likelihood that the teacher is competent at judging apples and pears under whatever conditions (light etc.) held at the time of judgement.

Frequentist Semantic Learning: Example II

- In this particular example, all situations have been judged with respect to all variable value types, which means that $\mathfrak{J} = \mathfrak{J}_{Green} = \mathfrak{J}_{Red} = \mathfrak{J}_{Ashape} = \mathfrak{J}_{Pshape} = \mathfrak{J}_{Apple} = \mathfrak{J}_{Pear}$
- In our example, $p(F||L \wedge S)$ comes from previous experience as encoded in $\mathfrak{J}$:

$$p(F||L \wedge S) = \frac{\mathfrak{p}_{\mathfrak{J}}(F)p(L||F)p(S||F)}{\sum_{F' \in \mathfrak{R}(Fruit)} \mathfrak{p}_{\mathfrak{J}}(F')p(L||F')p(S||F')}$$

- To compute this we need the following for $F \in \{Apple, Pear\}$:
  - for all $L \in \{Green, Red\}$, $p(L||F)$
  - for all $S \in \{Ashape, Pshape\}$, $p(S||F)$
  - $\mathfrak{p}_{\mathfrak{J}}(F)$

Frequentist Semantic Learning: Example III

- We use

$$p(E_i||C) = \frac{\sum_{j \in \mathfrak{J}, j.\mathrm{sit}=s} p(s:C)p(s:E_i)}{\sum_{j \in \mathfrak{J}, j.\mathrm{sit}=s} p(s:C)}$$

so for example

$$p(Green||Apple) = \frac{\sum_{j \in \mathfrak{J}, j.\mathrm{sit}=s} p(s:Apple)p(s:Green)}{\sum_{j \in \mathfrak{J}, j.\mathrm{sit}=s} p(s:Apple)} =$$

$$\frac{0.9 * 1.0 + 0.7 * 0.5 + 1.0 * 0.9 + 0.0 * 0.1}{1.0 + 0.5 + 0.9 + 0.1} = \frac{2.15}{2.50} = 0.86$$

Frequentist Semantic Learning: Example IV

- We also use

$$\mathfrak{p}_{\mathfrak{J}}(T) = \frac{\sum_{j \in \mathfrak{J}_T} j.\mathrm{prob}}{\mathrm{P}(\mathfrak{J})} \text{ if } \mathrm{P}(\mathfrak{J}) > 0, \text{ otherwise } 0$$

so for example

$$\mathfrak{p}_{\mathfrak{J}}(Apple) = \frac{\sum_{j \in \mathfrak{J}, j.\mathrm{sit}=s} p(s : Apple)}{\mathrm{P}(\mathfrak{J})} =$$

$$\frac{1.0 + 0.5 + 0.9 + 0.1}{4} = \frac{2.50}{4} = 0.63$$

Outline

Semantic learning using a linear tranformation model

Working on it!

Outline

Future work I

- Finish learning using linear transformation model
- Relate probabilistic inference more explicitly to other types of inference in TTR (action rules, functions)
- Represent probabilistic dependencies inside ProbTTR (now, only represented in the classifier function)
- Investigate how learning of probabilistic dependencies (e.g. enthymemes) interacts with semantic learning

Future work II

- Investigate how learning of continuous and discrete random variables interacts
- Applying Bayesian inference and classification in ProbTTR to a variety of problems in natural language semantics, including vagueness (where some initial steps are taken in Fernández and Larsson (2014)), probabilistic reasoning in dialogue, and learning grounded meanings from interaction (along the lines of Larsson (2013)).
- Implement this integrated system in order to demonstrate its viability as a computational model of natural language learning, reasoning and interaction.

Barwise, Jon 1989.
The Situation in Logic.
CSLI Publications, Stanford.

Breitholtz, Ellen 2020.
Enthymemes and Topoi in Dialogue: The Use of Common Sense Reasoning in Conversation.
Brill, Leiden, The Netherlands.

Cooper, Robin and Ginzburg, Jonathan 2015.
Type theory with records for natural language semantics.
In Lappin, Shalom and Fox, Chris, editors 2015, *The Handbook of Contemporary Semantic Theory, Second Edition*. Wiley-Blackwell, Oxford and Malden.
375–407.

Cooper, Robin; Dobnik, Simon; Lappin, Shalom; and Larsson, Staffan 2014.
A probabilistic rich type theory for semantic interpretation.

In *Proceedings of the EACL 2014 Workshop on Type Theory and Natural Language Semantics (TTNLS)*. Gothenburg, Association of Computational Linguistics.
72–79.

Cooper, Robin; Dobnik, Simon; Lappin, Shalom; and Larsson, Staffan 2015.
Probabilistic type theory and natural language semantics.
Linguistic Issues in Language Technology 10 1–43.

Cooper, Robin 2005.
Records and record types in semantic theory.
Journal of Logic and Computation 15(2):99–112.

Cooper, Robin 2012a.
Type theory and semantics in flux.
In Kempson, Ruth; Asher, Nicholas; and Fernando, Tim, editors 2012a, *Handbook of the Philosophy of Science*, volume 14: Philosophy of Linguistics. Elsevier BV.
General editors: Dov M. Gabbay, Paul Thagard and John Woods.

Cooper, Robin 2012b.
Type theory and semantics in flux.
In Kempson, Ruth; Asher, Nicholas; and Fernando, Tim, editors 2012b, *Handbook of the Philosophy of Science*, volume 14: Philosophy of Linguistics. Elsevier BV.
General editors: Dov M. Gabbay, Paul Thagard and John Woods.

Cooper, Robin prep.
From perception to communication: An analysis of meaning and action using a theory of types with records (TTR).
Draft available from `https://sites.google.com/site/typetheorywithrecords/drafts`.

Fernández, Raquel and Larsson, Staffan 2014.
Vagueness and learning: A type-theoretic approach.
In *Proceedings of the 3rd Joint Conference on Lexical and Computational Semantics (∗SEM 2014)*.

Halpern, J. 2003.

Reasoning About Uncertainty.
MIT Press, Cambridge MA.

Larsson, Staffan 2013.
Formal semantics for perceptual classification.
Journal of Logic and Computation.

Larsson, Staffan 2015.
Formal semantics for perceptual classification.
Journal of Logic and Computation 25(2):335–369.
Published online 2013-12-18.

Larsson, Staffan 2020.
Discrete and probabilistic classifier-based semantics.
In *Proceedings of the Probability and Meaning Conference (PaM 2020)*, Gothenburg. Association for Computational Linguistics.
62–68.

Pearl, J. 1990.
Bayesian decision methods.

In Shafer, G. and Pearl, J., editors 1990, *Readings in Uncertain Reasoning*. Morgan Kaufmann.
345–352.

Russell, Stuart and Norvig, Peter 1995.
Artificial Intelligence: A Modern Approach.
Prentice Hall Series in Artificial Intelligence. Englewood Cliffs, New Jersey.