Semantic classification and learning using a linear tranformation model in a Probabilistic Type Theory with Records

Staffan Larsson **Jean-Philippe Bernardy**
Centre for Linguistic Theory and Studies in Probability (CLASP)
Dept. of Philosophy, Linguistics and Theory of Science
University of Gothenburg, Box 200, SE 40530 Sweden
sl@ling.gu.se jean-philippe.bernardy@gu.se

Abstract

Starting from an existing account of semantic classification and learning from interaction formulated in a Probabilistic Type Theory with Records, encompassing Bayesian inference and learning with a frequentist flavour, we observe some problems with this account and provide an alternative account of classification learning that addresses the observed problems. The proposed account is also broadly Bayesian in nature but instead uses a linear transformation model for classification and learning.

1 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 (TTR, Cooper (2012a); Cooper and Ginzburg (2015)). This theory, Probabilistic Type Theory with Records (ProbTTR) assigns probability values, rather than Boolean truth-values, to type judgements.

TTR has been used previously for natural language semantics (see, for example, Cooper (2005) and Cooper (2012a)), and to analyse semantic coordination and learning (for example, (Larsson and Cooper, 2009; Cooper and Larsson, 2009)). It has also been applied to the analysis of interaction in dialogue (for example, Ginzburg (2012) and Breitholtz (2020)), in modelling robotic states and spatial cognition (for example, Dobnik et al. (2013)), and to the problem of learning perceptual meaning from interaction (Larsson, 2015). We believe that a probabilistic version of TTR could be useful in all these domains.

Two main considerations motivated recasting TTR in probabilistic terms. First, a probabilistic type theory offers a natural framework for capturing the gradience of semantic judgements. This allows it to serve as the basis for an account of vagueness in interpretation, as shown by Fernández and Larsson (2014). Second, and this is the focus of the present paper, such a theory lends itself to developing a model of semantic classification and learning that can be straightforwardly integrated into more general probabilistic explanations of learning and inference.

This paper presents an account of probabilistic classification (inference) and learning in ProbTTR based on a linear transformation model. Recent work (Larsson, 2020; Larsson and Cooper, 2021; Larsson et al., 2021; Larsson, 2021) has developed and used a Bayesian account of classification and a learning theory with a frequentist flavour. Below in Section 2, we first introduce TTR and ProbTTR, and explain briefly how a Naive Bayes classifier can be formulated in ProbTTR. We then review earlier work on semantic classification and learning using ProbTTR, and introduce a simple language game (the fruit recognition game) that has been used as an example there. In Section 3, we note some drawbacks of the frequentist account of classification and learning, motivating the exploration of alternative accounts. The main contribution of this paper is the account of semantic classification and learning using a linear transformation model presented in Section 4. We show how classification (Section 4.2) and learning (Section 4.3) is handled in this account, again taking the fruit recognition game as our example. In Section 4, we provide conclusions and point towards future work.

2 Background

This section reviews the background needed to follow the rest of the paper: TTR, Probabilistic TTR fundamentals, and Bayes nets and Naive Bayes classifiers.

2.1 TTR: A brief introduction

We will be formulating our account in a Type Theory with Records (TTR). We can here only give

Proceedings of the Conference on Reasoning and Interaction (ReInAct), pages 14–22
October 4-6, 2021.

a brief and partial introduction to TTR; see also Cooper (2005) and Cooper (2012b). To begin with, $s : T$ is a judgment that some s is of type T. One *basic type* in TTR is Ind, the type of an individual; another basic type is Real, the type of real numbers.

Next, we introduce *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 in Figure 1 is of the record type to the right.

In Figure 1, $\ell_1, \ldots \ell_n$ are *labels* which can be used elsewhere to refer to the values associated with them. A sample record and record type is shown in Figure 2.

Types constructed with predicates may be *dependent*. This is represented by the fact that arguments to the predicate may be represented by labels used on the left of the ':' elsewhere in the record type. In Figure 2, the type of c_{man} is dependent on ref (as is c_{run}).

If r is a record and ℓ is a label in r, we can use a *path* $r.\ell$ to refer to the value of ℓ in r. Similarly, if T is a record type and ℓ is a label in T, $T.\ell$ refers to the type of ℓ in T. Records (and record types) can be nested, so that the value of a label is itself a record (or record type). As can be seen in Figure 2, types can be constructed from predicates, e.g., "run" or "man". Such types are called *ptypes* and correspond roughly to propositions in first order logic.

2.2 Probabilistic TTR fundamentals

In ProbTTR (as in TTR generally), situations are understood in a sense similar to that of Barwise and Perry (1983). It is also assumed that agents can individuate situations, and that they have a way of judging situations to be of situation types.

The core of ProbTTR is the notion of a probabilistic judgement, where a situation s is judged to be of a type T with some probability.

(1) $p(s : T) = r$, where $r \in$ [0,1]

Such a judgement expresses a subjective probability in that it encodes an agent's take on the likelihood that a situation is of that type.

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 in (2).

(2)
$$\begin{bmatrix} \text{sit} & : & \textit{Sit} \\ \text{sit-type} & : & \textit{Type} \\ \text{prob} & : & [0,1] \end{bmatrix}$$

A probabilistic Austinian proposition φ of this type corresponds to the judgement that φ.sit is of type φ.sit-type with probability φ.prob.

(3) $p_{\mathfrak{J}}(\varphi.\text{sit}{:}\varphi.\text{sit-type}) = \varphi.\text{prob}$

We assume that agents track observed situations and their types, modelled as probabilistic Austinian propositions.

We use $p(T_1||T_2)$ to represent the probability that an agent assigns to some situation s being of type T_1, given that s is of type T_2. Note that $p(T_1||T_2)$, the conditional probability for some s of $s : T_1$ given that $s : T_2$, is different from $p(T_1|T_2)$, the probability of there being something of type T_1 given that there is something of type T_2. We refer to the former as the *bound variable* conditional probability, and the latter as the *existential* conditional probability.

2.3 Bayesian nets and the Naive Bayes classifier

A Bayesian Network is a Directed Acyclic Graph (DAG). The nodes of the DAG are random variables, each of whose values is the probability of one of the set of possible states that the variable denotes. Its directed edges express dependency relations among the variables. When the values of all the variables are specified, the graph describes a complete joint probability distribution (JPD) for its random variables. Bayesian Networks provide graphical models for probabilistic learning and inference (Pearl (1990); Halpern (2003)).

A standard Naive Bayes model is a special case of a Bayesian network. More precisely, it 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. Figure 2 illustrates the relation between evidence types and class types in a Naive Bayes classifier.

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

(4)
$$p(c) = \sum_{e_1,\ldots,e_n} p(c \mid e_1, \ldots, e_n) p(e_1) \ldots p(e_n)$$

where c_1 is the value of C, e_i is the value of E_i $(1 \leq i \leq n)$ and

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

Figure 1: Schema of record and record type

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

Figure 2: Sample record and record type

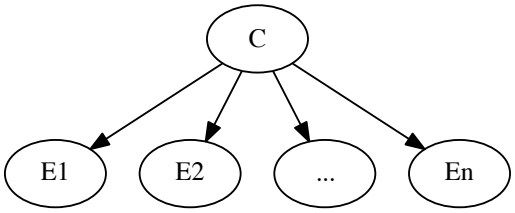

Figure 3: Evidence and Class in a Naive Bayes classifier

(5) $p(c \mid e_1, \dots, e_n) =$

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

2.4 Random variables in TTR

Larsson and Cooper (2021) introduce a type theoretic counterpart of a random variable in Bayesian inference. To represent a single (discrete) random variable with a range of possible (mutually exclusive) values, ProbTTR uses a *variable type* V whose range is a set of *value types* $\mathfrak{R}(V) = \{A_1, \dots, A_n\}$ such that the following conditions hold.

(6) a. $A_j \sqsubseteq V$ 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, $p(s : V) \in \{0, 1\}$ and $p(s : V) = \sum_{T \in \mathfrak{R}(V)} p_{\mathfrak{J}}(s : T)$

2.5 Representing probability distibutions

For a situation s, a probability distribution over the m value types $A_j \in \mathfrak{R}(\mathbb{A}), 1 \leq j \leq m$ belonging to a variable type $\mathbb{A}$ can be written (as above) as a set of probabilistic Austinian propositions, e.g.

(7) $\{ \begin{bmatrix} \text{sit} = s \\ \text{sit-type} = A_j \\ \text{prob} = p(s : A_j) \end{bmatrix} \mid A_j \in \mathfrak{R}(\mathbb{A}) \}$

However, we will also have use for a vector representation of probability distributions, which is also more compact. If we assume $\mathfrak{R}(\mathbb{A})$ is an ordered set $\{A_1, \dots A_m\}$, we can define probability distribution $d_{\mathbb{A}}(s)$:

(8) $d_{\mathbb{A}}(s) = \langle p_1, \dots, p_m \rangle$ where $p_j = p(s : A_j)$ for $A_j \in \mathfrak{R}(\mathbb{A}), 1 \leq i \leq m$

We also write $d_{\mathbb{A}}(s)_j$ for $p(s : A_j)$. This means we can reformulate (11) above:

(9) $d_{\mathbb{C}^\kappa}(s) = \langle p(s : C_1), \dots, p(s : C_{|\mathfrak{R}(\mathbb{C}^\kappa)|}) \rangle$

2.6 A ProbTTR Naive Bayes classifier

Corresponding to the evidence, class variables, and their value types, we associate with a ProbTTR Naive Bayes classifier κ:

(10) a. a collection of n evidence variable types $\mathbb{E}^\kappa_1, \dots, \mathbb{E}^\kappa_n$

b. n associated sets of evidence value types $\mathfrak{R}(\mathbb{E}^\kappa_1), \dots, \mathfrak{R}(\mathbb{E}^\kappa_n)$

c. a class variable type $\mathbb{C}^\kappa$, e.g. *Fruit*, and

d. an associated set of class value types $\mathfrak{R}(\mathbb{C}^\kappa)$

To classify a situation s using a classifier κ, the evidence is acquired by observing and classifying s with respect to the evidence types.

Larsson and Cooper (2021) define a ProbTTR Bayes classifier κ as a function from a situation s (of the meet type of the evidence variable types $\mathbb{E}^\kappa_1, \ldots, \mathbb{E}^\kappa_n$) 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 $\mathbb{E}^\kappa_1, \ldots, \mathbb{E}^\kappa_n$. Formally, a ProbTTR Naïve Bayes classifier is a function

$$(11)\quad \kappa : \mathbb{E}^\kappa_1 \wedge \ldots \wedge \mathbb{E}^\kappa_n \rightarrow \text{Set}(\begin{bmatrix} \text{sit} & : & \textit{Sit} \\ \text{sit-type} & : & \textit{Type} \\ \text{prob} & : & [0,1] \end{bmatrix})$$

such that if[1] $s : \mathbb{E}^\kappa_1 \wedge \ldots \wedge \mathbb{E}^\kappa_n$, then

$$(12)\quad \kappa(s) = \{ \begin{bmatrix} \text{sit} = s \\ \text{sit-type} = C \\ \text{prob} = p(s : C) \end{bmatrix} \mid C \in \mathfrak{R}(\mathbb{C}^\kappa) \}$$

or equivalently,

$$(13)\quad \kappa(s) = \{ \begin{bmatrix} \text{sit} = s \\ \text{sit-type} = C \\ \text{prob} = d_{\mathbb{C}^\kappa}(s)_C \end{bmatrix} \mid C \in \mathfrak{R}(\mathbb{C}^\kappa) \}$$

2.7 The fruit recognition game

Larsson and Cooper (2021) illustrate semantic classification and learning using a Naive Bayes classifier in ProbTTR using the *Apple Recognition Game*. In this game a teacher shows a learning agent fruits. The agent makes a guess, the teacher provides the correct answer, and the agent learns from these observations.

We will use shorthands *Apple* and *Pear* for the types corresponding to an object being an apple or a pear, respectively[2]. Furthermore, we will assume that the objects in the Apple Recognition Game have one of two shapes (a-shape or p-shape, corresponding to types *Ashape* and *Pshape*= and one of two colours (green or red, corresponding to types *Green* and *Red*).

The class variable type is *Fruit*, with value types $\mathfrak{R}(\textit{Fruit}) = \{\textit{Apple}, \textit{Pear}\}$. The evidence variable types are (i) *Col*(our), with value types $\mathfrak{R}(\textit{Col}) = \{\textit{Green}, \textit{Red}\}$, and (ii) Shape, with value types $\mathfrak{R}(\textit{Shape}) = \{\textit{Ashape}, \textit{Pshape}\}$. Figure 4 shows the evidence and class types of the Apple Recognition Game in a simple Bayesian Network.

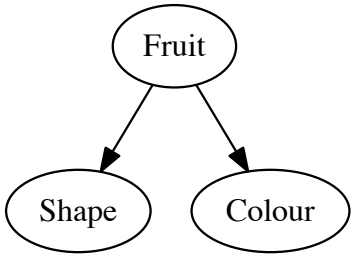

Figure 4: Bayesian Network for the Apple Recognition Game

For a situation s the classifier FruitC(s) returns a probability distribution over the value types in $\mathfrak{R}(\textit{Fruit})$.

$$(14)\quad \text{FruitC}(s) = \{ \begin{bmatrix} \text{sit} = s \\ \text{sit-type} = F \\ \text{prob} = p^{\text{FruitC}}(s : F) \end{bmatrix} \mid F \in \mathfrak{R}(\textit{Fruit}) \}$$

2.8 A frequentist model of semantic classification and learning

In Larsson et al. (2021), an account of semantic classification and learning with a frequentist flavour (but also with some differences to regular frequentist learning acccounts) is presented, under the assumption that we can compute conditional probabilities $p(C_j || E_1 \ldots E_n)$ of a class value types C_j given evidence value types $E_1 \ldots E_n$.

In general, for $C_j \in \mathfrak{R}(\mathbb{C}^\kappa)$, we have

$$(15)\quad p(s : C_j) = \sum_{\substack{E_1 \in \mathfrak{R}(\mathbb{E}^\kappa_1) \\ \ldots \\ E_n \in \mathfrak{R}(\mathbb{E}^\kappa_n)}} p(C_j || E_1 \ldots E_n) p(s : E_1) \ldots p(s : E_n)$$

The non-conditional probabilities $p(s : E_1) \ldots p(s : E_n)$ are derived from the agents' take on the particular situation s being classified, coming for example from perceptual sensors that are directed at s.

[1] Recall that for any s, $p_{\mathfrak{I}}(s : V) \in \{0, 1\}$ for any variable type V. Therefore, any type judgement regarding a variable type, such as that involved in the classifier function, can be regarded as categorical.

[2] For details, see Larsson and Cooper (2021).

For the model of semantic classification that uses conditional probabilities, a central question is of course how to estimate conditional probabilities, of the form $p(C||E_1 \wedge \ldots \wedge E_n)$ (where $C \in \mathfrak{R}(\mathbb{C}), E_i \in \mathfrak{R}(\mathbb{E}_i), 1 \leq i \leq n$). Using Bayes rule and marginalising over the class value types, we get for a Naive Bayes classifier:

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

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

To estimate the likelihoods $p(E_i||C)$ and priors $p(C')$, Larsson et al. (2021) use a version of counting previous instances of C and E_i:

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

The account in (ibid.) is 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 $p_{\mathfrak{J}}(s : T)$ on the basis of A's previous judgements $\mathfrak{J}$. For all combinations of evidence value types $E_1, \ldots, E_n$ and class value types C, the account in (ibid.) computes the conditional probability of the evidence value types given the class value type as in (17):

(17)

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

Note that the recorded judgements concerning the class types $C \in \mathfrak{R}(\mathbb{C})$ are here assumed to be derived mainly from a tutor's explicit judgements, which are thus assumed to provide the ground truth.

The account in (ibid.) also computes the prior of the class value type as in (18). $\mathfrak{p}_{\mathfrak{J}}(T)$ represents the prior probability that an arbitrary situation is of type T given $\mathfrak{J}$.

(18)

$$\mathfrak{p}_{\mathfrak{J}}(T) = \frac{||\ T\ ||_{\mathfrak{J}}}{\text{P}(\mathfrak{J})} \text{ if } \text{P}(\mathfrak{J}) > 0, \text{ otherwise } 0$$

where $\text{P}(\mathfrak{J})$ is the cardinality of situations in $\mathfrak{J}$, i.e. the total number of situations in $\mathfrak{J}$.

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

3 Drawbacks of the frequentist account

While conceptually simple, the above account, as any frequentist model, has some drawbacks. Some are well known, such as (problem P1) assigning probability 0 to judgements concerning unseen types, and (P2) putting equal weight on old and recent observations, thereby risking that classifiers for types that have a large amount of related judgements in $\mathfrak{J}$ may change only very slowly in light of new observations. Also, (P3) the account may be computationally unwieldy in real life settings since conditional probabilities are computed from scratch from $\mathfrak{J}$ on every instance of classification.

Other drawbacks are more specifically related to our goal of modelling semantic coordination in dialogue, where both definitions (or explications) and examples can affect meanings but in different ways (Myrendal, 2019; Larsson and Myrendal, 2017; Larsson, 2021). With respect to the problem (P4) of combining evidence from examples and definitions (as described in Larsson (2021)), the frequentist model does not provide a theoretically satisfying way of doing this. While a definition may be useful until examples have been observed, at some point the observed examples may override a definition. In the account proposed in (Larsson, 2021), definitions affect the corresponding classifier only in the short run, and effects of proposed definitions are overwritten as soon as an observation of an instance of the defined concept has been made. A more flexible trade-off between definitions and examples (observations) would probably be desirable in this context.

Finally, (P5) the frequentist model has little to say about the relation between the learning agents' own judgement and the judgement given by the teacher with respect to how much weight is put on these relative to each other when learning from interaction. Does the agent completely trust the tutor, or does it weigh in other factors when learning from tutor input?

While there may be ways of addressing at least some of these problems within the frequentist account[3], we will here explore an alternative account

[3]One might argue that the interactive learning setting already addresses P1 to the extent that tutor input can override the agent's judgement concerning unseen types. To address P2, the frequentist model could be amended with exponential decay over $\mathfrak{J}$. To address P3, some method of caching conditional probabilities and priors computed from $\mathfrak{J}$, and updating them only when needed, might be devised. To address P4, one could let a definition lead to adding some relatively high number N of "fake" observations in line with the definition to $\mathfrak{J}$.

that seems to address all these problems without the need for ad hoc solutions.

4 Semantic classification and learning using a linear transformation model

In this section we present a model where the probabilities given in $\mathfrak{J}$ are used to compute a linear transformation model Θ that generalises over $\mathfrak{J}$ and which is used to compute the conditional probabilities used in classification. Such a model can be made more computationally efficient than the frequentist model, and is also compatible with the way probabilistic inference and learning is encoded in neural network models.

4.1 Modelling how Evidence is determined by Class

Like before we assume that evidence variables are determined independently by the class variable (in the fruit recognition game game, *Col(our)* and *Shape* are determined independently by the Fruit variable).

Following standard practice in deep learning models, a probability distribution over the values (class value types) of the class variable type $\mathbb{C}$ is mapped to probability distributions over the evidence value types corresponding to each evidence variable type $\mathbb{E}_i$ (in the fruit recognition game, fruit type is mapped to a colour distribution and a shape distribution) using a linear transformation, represented by a matrix $\Theta_{\mathbb{C}\to\mathbb{E}_i}$ (in the apple game, $\Theta_{Fruit\to Col}$ and $\Theta_{Fruit\to Shape}$) followed by a softmax. Let us call Θ the combined parameters of such linear transformations. For a classifier κ, a subset Θ^κ of the parameters can be used.

For example, in the apple game, we assume that the probability distribution over the variable value types in $\mathfrak{R}(Col)$ are estimated thus:

$$\hat{d}_{Col}(s) = \mathsf{softmax}(\Theta^\kappa_{Fruit\to Col} d_{Fruit}(s)) \quad (20)$$

In general,

$$\hat{d}_{\mathbb{E}_i}(s) = \mathsf{softmax}(\Theta^\kappa_{\mathbb{C}\to\mathbb{E}_i} d_{\mathbb{C}}(s)) \quad (21)$$

Even more generally, with an arbitrary bayesian network, we take into account all edges to the variable $\mathbb{E}$ (giving a finite set of unobserved variables ranged by $\mathbb{I}$ below). We assume simultaneously a parameter matrix $\Theta^\kappa_{\mathbb{IE}}$ for each such edge:

$$\hat{d}_{\mathbb{E}}(s) = \mathsf{softmax}(\textstyle\sum_{(I\to E)\in net} \Theta^\kappa_{\mathbb{IE}} d_{\mathbb{I}}(s)) \quad (22)$$

It follows that for $E_j \in \mathfrak{R}(\mathbb{E}_i)$, an estimation of the probability of a situation having evidence value type E_j is:

$$\hat{p}(s : E_j) = \hat{d}_{\mathbb{E}_i}(s)_j = \mathsf{softmax}(\Theta^\kappa_{\mathbb{C}\to\mathbb{E}_i} d_{\mathbb{C}}(s))_j \quad (23)$$

Expanding the definition of softmax, we get:

$$\hat{p}(s : E_j) = \frac{e^{(\Theta^\kappa_{\mathbb{C}\to\mathbb{E}_i} d_{\mathbb{C}}(s))_j}}{\sum_{E_k\in\mathfrak{R}(\mathbb{E}_i^\kappa)} e^{(\Theta^\kappa_{\mathbb{C}\to\mathbb{E}_i} d_{\mathbb{C}}(s))_k}} \quad (24)$$

or equivalently (dot, products and column vectors)

$$\hat{p}(s : E_j) = \frac{e^{(\Theta^\kappa_{\mathbb{C}\to E_j}\cdot d_{\mathbb{C}(s)})}}{\sum_{E_k\in\mathfrak{R}(\mathbb{E}_i^\kappa)} e^{(\Theta^\kappa_{\mathbb{C}\to E_k}\cdot d_{\mathbb{C}(s)})}} \quad (25)$$

Note that softmax is here overloaded to be used for vectors of probabilities as well as for individual probabilities.

We also define for any distribution $d_{\mathbb{A}}$ over (variable value types of) variable type $\mathbb{A}$:

$$\check{d}_{\mathbb{B}}(d_{\mathbb{A}}) = \mathsf{softmax}(\Theta_{\mathbb{A}\to\mathbb{B}} d_{\mathbb{A}}) \quad (26)$$

so that e.g.

$$\hat{d}_{\mathbb{E}}(s) = \check{d}_{\mathbb{E}}(d_{\mathbb{C}}(s)) = \mathsf{softmax}(\Theta^\kappa_{\mathbb{C}\to\mathbb{E}} d_{\mathbb{C}}(s)) \quad (27)$$

4.2 Classification using a transformation model

When we use a transformation model for classification, the idea is to evaluate the likelihood of a distribution $\hat{d}_{\mathbb{C}}(s)$ which according to the model Θ accounts for the observed evidence $d_{\mathbb{E}_i}(s)$[4]. This means we need to represent meta-level probabilities of a probability distribution given another probability distribution.

When classifying fruits in the Apple game, we want to estimate the probability of the class value types given the observed distribution over the evidence value types. The probability for a particular distribution $d_{\mathbb{C}}(s)$ is estimated using Bayesian marginalisation:

By manipulation of N, the relative importance of definitions relative to observations can be regulated. To address P5, judgements $s : C$ regarding class value types C that are added to $\mathfrak{J}$ could be made to reflect a combination of teacher judgement and other factors, including the agents' own estimation.

[4] It is also possible to not decide on one distribution, but to keep a distribution over distributions over the class variable.

(28) $\hat{p}(d_{\mathbb{C}}(s)|d_{\mathbb{E}_i}(s)) \propto p(d_{\mathbb{E}_i}(s)|d_{\mathbb{C}}(s)) \times prior(d_{\mathbb{C}}(s))$

If we want to find the distribution $d_{\mathbb{C}(s)}$ that maximises the observed evidence in light of the model, for a single evidence variable type $\mathbb{E}_i$ we want to find

(29) $\text{argmax}_{z\in[0,1]^{|\mathfrak{R}(\mathbb{C})|}}\hat{p}(d_{\mathbb{E}_i(s)}|z)\text{prior}(z)$

and for evidence variable types $\mathbb{E}_1^\kappa, \ldots \mathbb{E}_n^\kappa$ we want to find

(30) $d_{\mathbb{C}^\kappa}(s) = \text{argmax}_{z\in[0,1]^{|\mathfrak{R}(\mathbb{C})|}} \hat{p}(d_{\mathbb{E}_1}(s)|z)\ldots\hat{p}(d_{\mathbb{E}_n}(s)|z)\text{prior}(z)$

where z is ranging over the space of distributions over $\mathbb{C}$ value types . If we have $k = |\mathfrak{R}(\mathbb{C})|$ possible value types, this space is contained in $[0,1]^k$. To find z we need a numerical method, e.g. gradient descent.

To classify a situation s with respect to each $C_j \in \mathfrak{R}(\mathbb{C})$,

(31) $\hat{p}(s : C_j) = \hat{d}_{\mathbb{C}}(s)_j$

In the fruit game, for each $C_j \in \mathfrak{R}(\textit{Fruit})$,

(32) $\hat{p}(s : C_j) = \hat{d}_{\mathbb{C}}(s)_j$ = $\text{argmax}_{z\in[0,1]^2}\hat{p}(d_{\textit{Col}}(s)|z)\hat{p}(d_{\textit{Shp}}(s)|z)\text{prior}(z)$

Conditional probabilities Instead of estimating the conditional probability of an evidence value type given a class value type, as in the frequentist model, we here estimate the conditional probability of a distribution over evidence value types given a distribution over class value types belonging to the class variable type.

The probability of an observed probability distribution $d_{\mathbb{E}_i}(s)$ over evidence value types $E_j \in \mathbb{E}_i$ for a situation s given a distribution $d_{\mathbb{C}}(s)$ over the class value types for s can be estimated as:

(33) $\hat{p}(d_{\mathbb{E}_i}(s)|d_{\mathbb{C}}(s)) = e^{-H(d_{\mathbb{E}_i}(s),\check{d}_{\mathbb{E}_i}(d_{\mathbb{C}}(s)))}$

where $H(d_{\mathbb{E}_i}(s), \check{d}_{\mathbb{E}_i}(d_{\mathbb{C}}(s)))$ is the cross entropy between the observed distribution over the evidence $d_{\mathbb{E}_i}(s)$ and the distribution $\check{d}_{\mathbb{E}_i}(d_{\mathbb{C}}(s))$ over the evidence variable type $\mathbb{E}_i$ as predicted by the model $\Theta^\kappa_{\mathbb{C}\to\mathbb{E}_i}$ based on a (hypothetical) distribution over the class variable.

Probability Density Functions In reality, $d_{\mathbb{C}}(s)$ is a continuous variable (since it is a probability distribution), so $p(d_{\mathbb{C}}(s)) = 0$. Basically, since there are uncountably many possible probability distributions, the probability of any one of them is zero.

However, the same kind formula works for Probability Density Functions (PDFs) which give probability distributions over a continuous variable. Writing f for PDF, we have:

(34) $f_{d_{\mathbb{E}_i}(s)}(d_{\mathbb{C}}(s)) \propto p(d_{\mathbb{E}_i}(s)|d_{\mathbb{C}}(s)) \times f_{prior}(d_{\mathbb{C}}(s))$

corresponding to (28), repeated here as (35)):

(35) $\hat{p}(d_{\mathbb{C}}(s)|d_{\mathbb{E}_i}(s)) \propto p(d_{\mathbb{E}_i}(s)|d_{\mathbb{C}}(s)) \times prior(d_{\mathbb{C}}(s))$

As before, when classifying we want the the distribution $\hat{d}_{\mathbb{C}}(s)$ that maximises the probability that the model Θ accounts for the evidence. For a single evidence variable, this is $d_{\mathbb{E}_i}(s)$.

(36) $\hat{d}_{\mathbb{C}^\kappa}(s) = argmax_z \hat{p}(d_{\mathbb{E}_i}(s)|z) f_{prior}(z)$

For n evidence variables:

(37) $\hat{d}_{\mathbb{C}^\kappa}(s) = argmax_z \hat{p}(d_{\mathbb{E}_1}(s)|z)\cdots\hat{p}(d_{\mathbb{E}_n}(s)|z) f_{prior}(z)$

corresponding to (30), repeated here as (38):

(38) $d_{\mathbb{C}^\kappa}(s) = \text{argmax}_{z\in[0,1]^{|\mathfrak{R}(\mathbb{C})|}} \hat{p}(d_{\mathbb{E}_1(s)}|z)\ldots\hat{p}(d_{\mathbb{E}_n(s)}|z)\text{prior}(z)$

Priors for $d_{\mathbb{C}}$ There are many ways to give a prior for $d_{\mathbb{C}}$. We know that (1) it must be a function of Θ and (2) must be a probability distribution. One way to satisfy these requirements is to follow the same recipe as for evidence (but with no dependency). According to this recipe, we have the formula:

$$\hat{d}_{\mathbb{C}}(s) = \mathsf{softmax}(\Theta^\kappa_{\mathbb{C}})$$

This way, there is a functional dependency from Θ to $\hat{d}_{\mathbb{C}}(s)$, and therefore any prior density function on Θ yields another density function on $\hat{d}_{\mathbb{C}}(s)$, called hereafter $f_{prior}(\hat{d}_{\mathbb{C}}(s))$[5].

Note that here, $\Theta^\kappa_{\mathbb{C}}$ is a vector, not a matrix. The priors of each element in Θ can be an independent uniform distribution over reals.

[5]Unfortunately, because softmax is not a bijective function, there is no simple formula connecting these PDFs.

4.3 Learning

It remains to see how Θ gets updated by any learning event j. To do so, one uses Bayesian reasoning again. We start by evaluating the probability of a learning event in the form of a newly observed situation s associated with j actually occurring, given a fixed value of Θ. As in the frequentist account of learning, we assume that our agent A has stored in $\mathfrak{J}$ probabilistic judgements providing probability distributions for s over the class and evidence variables (or in the general case of a Bayes net, all evidence variables and unobserved variables).

We want to assign a high probability if they match the prediction and low otherwise. Following standard practice in information theory, we assign it the (inverse exponential of) the cross entropy of each characteristic's observed distribution, with the predicted distribution.

When learning from a tutor, as in the fruit recognition game, the learning agent computes the cross entropy between the predicted (estimated) distribution $\hat{d}_{\mathbb{C}}(s)$ and the distribution based on the teacher's input $d_{\mathbb{C}}(s)$, which is here treated as ground truth. By contrast, in the frequentist account, the predicted d_C played no role in learning (although it did affect the learning agent's guess).

Using $J^{\kappa}(s)$ as a shorthand for the probabilistic judgements concerning a situation s (with respect to an evidence variable $\mathbb{E}_i$ and a class variable C used by a classifier κ) encoded in $\mathfrak{J}$ (concretely, the observed probability distributions for s over $\mathbb{E}_i$ and $\mathbb{C}$), we can compute the conditional probability of these judgements given a classifier parameter matrix Θ^{κ} thus:

(39)

$$p(J^{\kappa}(s)|\Theta^{\kappa}) = p(d_{\mathbb{E}_i}(s), d_{\mathbb{C}}(s)|\Theta^{\kappa}) =$$

$$e^{-H(d_{\mathbb{C}}(s),\hat{d}_{\mathbb{C}}(s))} \times \prod_i e^{-H(d_{\mathbb{E}_i}(s),\hat{d}_{\mathbb{E}_i}(s))}$$

Using the same kind of Bayesian reasoning as always, we can marginalise:

$$p(\Theta|J(s)) \propto p(J(s)|\Theta)p(\Theta)$$

A benefit of this model that the estimation for various probabilities depend only on Θ. This means that the agent needs not remember the whole history J, only the distribution of Θ (over all $\Xi \in Parameters$). (Yet one can consider several learning events jointly when performing a Bayesian update.)

In practice, an actual agent will only work with an approximation of this distribution. For example, a neural net may remember just a single Θ, and instead of a Bayesian update it takes a gradient of $p(J(s)|\Theta)$ wrt. Θ and update it accordingly:

$$\Theta := \Theta - \alpha \frac{dp(J(s)|\Theta)}{d\Theta}$$

Insofar as the agent updates parameters directly, rather than updating the judgements history $\mathfrak{J}$ and using it to compute classifier parameters, this account addresses problem P2 noted above. Furthermore, the fact that the proposed model has an explicit learning factor is key to addressing some of the other problems noted above. Since the learning factor explicitly addresses the impact of new examples compared to previous observations, it enables us to address problem P3. To address problem P4, definitions can be associated with a higher learning factor than examples, to model the hypothesis that definitions have a much larger potential impact on an agents' take on a meaning compared to an example. Also, we could possibly use the learning factor α to model how much the teacher's judgement is prioritised over the agent's own judgement estimation (based on perception of the situation), thereby addressing problem P5.

5 Conclusions and future work

Previous work proposed a frequentist Bayesian account of semantic classification and learning formulated in terms of a Probabilistic Type Theory with Records. We observed some problems with this approach, including accounting for the effect of definitions as opposed to examples in learning meanings from interaction, and proposed an alternative account of learning that keeps the broadly Bayesian model of classification, but where classification is based on a linear transformation model. We argued that the account proposed here can address some of the problems of the frequentist account.

In future work, we wish to implement both the frequentist model (including some amendments to address observed problems) and the linear transformation model, and evaluate and compare them practically with respect to the problems P1-P5 noted above.

Acknowledgments

This work was supported by grant 2014-39 from the Swedish Research Council (VR) for the establishment of the Centre for Linguistic Theory and Studies in Probability (CLASP) at the University of Gothenburg.

References

Jon Barwise and John Perry. 1983. *Situations and Attitudes*. Bradford Books. MIT Press, Cambridge, Mass.

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

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

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

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

Robin Cooper, Simon Dobnik, Shalom Lappin, and Staffan Larsson. 2014. A probabilistic rich type theory for semantic interpretation. In *Proceedings of the EACL 2014 Workshop on Type Theory and Natural Language Semantics (TTNLS)*, pages 72–79. Gothenburg, Association of Computational Linguistics.

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

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

Robin Cooper and Staffan Larsson. 2009. Compositional and ontological semantics in learning from corrective feedback and explicit definition. In *Proceedings of DiaHolmia: 2009 Workshop on the Semantics and Pragmatics of Dialogue*, pages 59–66. Department of Speech, Music and Hearing, KTH.

Simon Dobnik, Robin Cooper, and Staffan Larsson. 2013. Modelling language, action, and perception in Type Theory with Records. In Denys Duchier and Yannick Parmentier, editors, *Constraint Solving and Language Processing - 7th International Workshop on Constraint Solving and Language Processing, CSLP 2012, Orleans, France, September 13-14, 2012. Revised Selected Papers*, number 8114 in Publications on Logic, Language and Information (FoLLI). Springer, Berlin, Heidelberg.

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

Jonathan Ginzburg. 2012. *The Interactive Stance: Meaning for Conversation*. Oxford University Press, Oxford.

J. Halpern. 2003. *Reasoning About Uncertainty*. MIT Press, Cambridge MA.

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

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

Staffan Larsson. 2021. The role of definitions in coordinating on perceptual meanings. In *Proceedings of the Workshop on the Semantics and Pragmatics of Dialogue (SemDial 2021)*.

Staffan Larsson, Jean-Philippe Bernardy, and Robin Cooper. 2021. Semantic learning in a probabilistic type theory with records. In *Proceedings of Workshop on Computing Semantics with Types, Frames and Related Structures 2021*.

Staffan Larsson and Robin Cooper. 2009. Towards a formal view of corrective feedback. In *Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition*, pages 1–9. EACL.

Staffan Larsson and Robin Cooper. 2021. Bayesian classification and inference in a probabilistic type theory with records. In *Proceedings of NALOMA 2021*.

Staffan Larsson and Jenny Myrendal. 2017. Dialogue acts and updates for semantic coordination. *SEMDIAL 2017 SaarDial*, page 59.

Jenny Myrendal. 2019. Negotiating meanings online: Disagreements about word meaning in discussion forum communication. *Discourse Studies*, 21(3):317–339.

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