UNIVERSITY OF GOTHENBURG

A Compositional Bayesian Semantics for Natural Language

Jean-Philippe Bernardy
Rasmus Blanck
Stergios Chatzikyriakidis
Shalom Lappin

Centre for Linguistic Theory and Studies in Probability (CLASP)
University of Gothenburg

CLASP Seminar, 3 October 2018

Introduction

Classical model-theoretic semantics interprets declarative sentences in terms of truth conditions. But:

- this excludes vagueness from semantic interpretation;
- this does not provide a natural framework for explaining semantic learning.

Introduction

Classical model-theoretic semantics interprets declarative sentences in terms of truth conditions. But:

- this excludes vagueness from semantic interpretation;
- this does not provide a natural framework for explaining semantic learning.

Several theories of probabilistic semantics for natural language have been proposed; but all suffer from some shortcomings.

Our Approach

- a compositional Bayesian semantics, interpreting declarative sentences by assigning them probability conditions
- the conditional probability of a sentence is the likelihood that an idealised speaker would accept the assertion
- assessing the probability of a sentence is an instance of evaluating the application of a classifier to a new argument
- straightforward treatments of vagueness in predication, gradable predicates, comparatives, generalised quantifiers, and probabilistic inferences across several property dimensions
- a prototype that offers a proof of concept for our approach

Implementation

Our semantics draws inspiration from (i) Montague semantics, (ii) vector space models, and (iii) Bayesian inference, and

- interprets sentences as probabilistic programs;
- uses the precise semantics for probabilistic programming provided by Borgström et al. (2013);
- uses Markov Chain Monte Carlo (MCMC) sampling to estimate probabilities, as described by Goodman et al. (2008);
- is encoded as a Haskell library, with calls into the WebPPL language of Goodman and Stuhlmüller (2014);
- code is available at `https://github.com/GU-CLASP/CompositionalBayesianSemantics`.

Implementation (cont.)

Following Montague, our semantics assumes an assignment from syntactic categories to types. These assignments are given in Haskell as follows:

```
type Pred = Ind -> Prop
type Measure = Ind -> Scalar
type AP = Measure
type CN = Ind -> Prop
type VP = Ind -> Prop
type NP = VP -> Prop
type Quant = CN -> NP
```

Implementation (cont.)

- Individuals and properties are represented as vectors in a multidimensional vector space.
- The distribution of individuals is a multivariate normal distribution of dimension k, with a zero mean and a unit covariance matrix.
- Predicates are parameterised by a bias b and a vector d.
- An individual x satisfies a predicate if the expression $b + d \cdot x > 0$ is true.

The Simplest Code Example

```
modelSimplest = do
  p <- newPred
  x <- newInd
  return (p x)
```

The Simplest Code Example

```
modelSimplest = do
  p <- newPred
  x <- newInd
  return (p x)
```

In the absence of further information, an arbitrary predicate has an even chance to hold of an arbitrary individual. Running the model gives the following approximate result:

```
true : 0.456     false : 0.544
```

A Simple Code Example

We may make assumptions about predicates and individuals, using the `observe` primitive.

```
modelSimple = do
   p <- newPred
   x <- newInd
   observe (p x)
   return (p x)
```

A Simple Code Example

We may make assumptions about predicates and individuals, using the `observe` primitive.

```
modelSimple = do
  p <- newPred
  x <- newInd
  observe (p x)
  return (p x)
```

Even when using our approximating implementation, evaluating the above model yields certainty.

```
true : 1
```

Comparatives

We support scalar predicates and comparatives. The expression $b + d \cdot x$ can be interpreted as a degree to which the individual x satisfies the property characterised by (b, d).

```
modelTall :: P Scalar
modelTall = do
  tall <- newMeasure
  john <- newInd
  mary <- newInd
  observe (more tall john mary)

  return (is tall john)
```

Comparatives

We support scalar predicates and comparatives. The expression $b + d \cdot x$ can be interpreted as a degree to which the individual x satisfies the property characterised by (b, d).

```
modelTall :: P Scalar
modelTall = do
  tall <- newMeasure
  john <- newInd
  mary <- newInd
  observe (more tall john mary)

  return (is tall john)
```

For the above example, we get:

```
true : 0.552    false : 0.448
```

Vague predicates

We support vague predication, by adding an uncertainty to each measure we make for the predicate in question. This is implemented through a Gaussian error with a given std. dev. σ for each measure.

```
modelTall :: P Prop
modelTall = do
  tall <- vague 3 <$> newMeasure
  john <- newInd
  mary <- newInd
  observe (more tall john mary)

  return (is tall john)
```

In this situation the tallness of John is more uncertain than before:

```
true : 0.488    false : 0.512
```

Generalised Quantifiers in General

On a standard reading, a generalised quantifier like “most" can be seen as a constraint on a ratio between the cardinality of sets.

$$most(cn, vp) = \frac{\#\{x : cn(x) \wedge vp(x)\}}{\#\{x : cn(x)\}} > \theta$$

for a suitable threshold θ.

Generalised Quantifiers in General

On a standard reading, a generalised quantifier like “most" can be seen as a constraint on a ratio between the cardinality of sets.

$$most(cn, vp) = \frac{\#\{x : cn(x) \wedge vp(x)\}}{\#\{x : cn(x)\}} > \theta$$

for a suitable threshold θ. In a probabilistic framework, we posit that the expected value of $vp(x)$ given that $cn(x)$ holds should be greater than θ.

$$most(cn, vp) = E(\mathbf{1}(vp(x))|cn(x)) > \theta$$

Generalised Quantifiers in General (cont.)

The expected value can be given a definite symbolic form:

$$most(cn, vp) = \frac{\int_{Ind} f_{\mathcal{N}}(x)\mathbf{1}(cn(x) \wedge vp(x))dx}{\int_{Ind} f_{\mathcal{N}}(x)\mathbf{1}(cn(x))dx} > \theta$$

where $f_{\mathcal{N}}$ denotes the density of the multivariate Gaussian distribution for individuals.

Generalised Quantifiers in General (cont.)

The expected value can be given a definite symbolic form:

$$most(cn, vp) = \frac{\int_{Ind} f_{\mathcal{N}}(x)\mathbf{1}(cn(x) \wedge vp(x))dx}{\int_{Ind} f_{\mathcal{N}}(x)\mathbf{1}(cn(x))dx} > \theta$$

where $f_{\mathcal{N}}$ denotes the density of the multivariate Gaussian distribution for individuals.
We implement the this in WebPPL by creating a probabilistic program p, which samples over all individuals x which satisfy cn, and we evaluate $vp(x)$. The compound statement is satisfied if the expected value of the program p, itself evaluated using an inner MCMC sampling procedure, is larger than θ.

Generalised Quantifiers and Chairs

On this basis, we make inferences of the following kind. “If many chairs have four legs, then it is likely that any given chair has four legs". We model this sentence as follows:

```
chairExample1 = do
  chair <- newPred
  fourlegs <- newPred
  observe (many chair fourlegs)
  x <- newIndSuch [chair]
  return (fourlegs x)
```

Generalised Quantifiers and Chairs

On this basis, we make inferences of the following kind. “If many chairs have four legs, then it is likely that any given chair has four legs". We model this sentence as follows:

```
chairExample1 = do
  chair <- newPred
  fourlegs <- newPred
  observe (many chair fourlegs)
  x <- newIndSuch [chair]
  return (fourlegs x)
```

Our model yields:

```
true : 0.821     false : 0.179
```

Generalised Quantifiers and Chairs (cont.)

The models that we are building implement generalised quantifiers through correlation of predicates, so we get 'inverse' correlation as well. In the absence of further information, and given an individual x with four legs, we will predict a high probability for $chair(x)$.

```
chairExample2 :: P Prop
chairExample2 = do
  chair <- newPred
  fourlegs <- newPred
  observe (many chair fourlegs)
  x <- newIndSuch [fourlegs]
  return (chair x)
```

Generalised Quantifiers and Chairs (cont.)

The models that we are building implement generalised quantifiers through correlation of predicates, so we get 'inverse' correlation as well. In the absence of further information, and given an individual x with four legs, we will predict a high probability for $chair(x)$.

```
chairExample2 :: P Prop
chairExample2 = do
  chair <- newPred
  fourlegs <- newPred
  observe (many chair fourlegs)
  x <- newIndSuch [fourlegs]
  return (chair x)
```

The model yields:

```
true : 0.653    false : 0.347
```

Generalised Quantifiers and Chairs (cont.)

The model's assumptions can be augmented with the hypothesis that most individuals are not chairs. This will lower the probability of being a chair appropriately.

```
chairExample3 :: P Prop
chairExample3 = do
  chair <- newPred
  fourlegs <- newPred
  observe (many chair fourlegs)
  observe (most anything (not' . chair))
  x <- newIndSuch [fourlegs]
  return (chair x)
```

Generalised Quantifiers and Chairs (cont.)

The model's assumptions can be augmented with the hypothesis that most individuals are not chairs. This will lower the probability of being a chair appropriately.

```
chairExample3 :: P Prop
chairExample3 = do
  chair <- newPred
  fourlegs <- newPred
  observe (many chair fourlegs)
  observe (most anything (not' . chair))
  x <- newIndSuch [fourlegs]
  return (chair x)
```

The model yields:

```
true : 0.221    false : 0.779
```

An Example Inference

Assume that

1. most animals do not fly;
2. most birds fly;
3. every bird is an animal.

Can we conclude that “most animals are not birds”?

An Example Inference (cont.)

We model the example as follows:

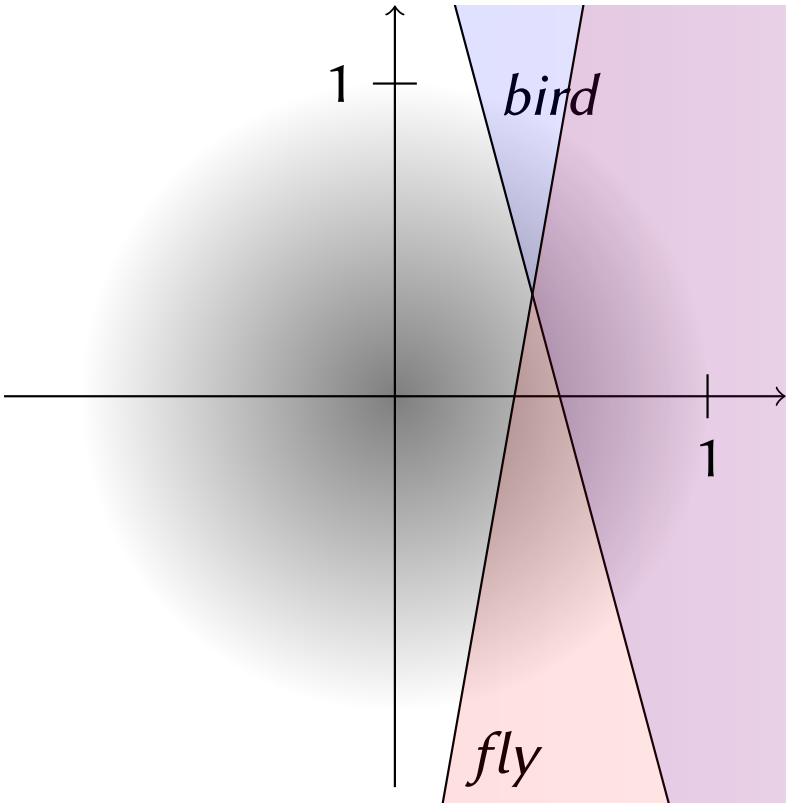
```
birdExample = do
  animal <- newPred
  bird <- newPred
  fly <- newPred

  observe (most animal (not' . fly))
  observe (most bird fly)
  observe (every bird animal)
  return (most animal (not' . bird))
```

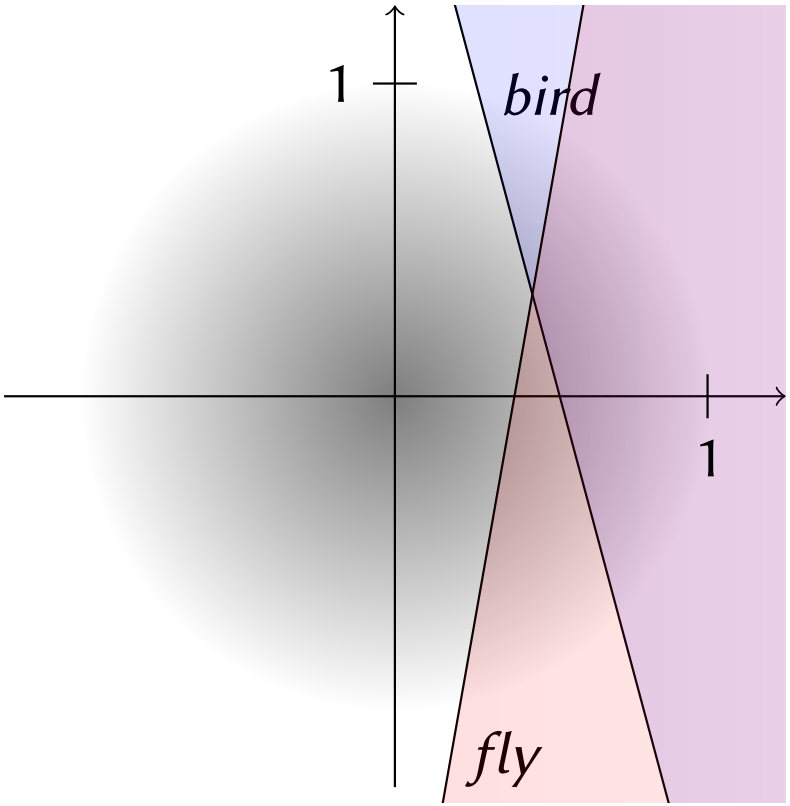
Our implementation concludes that "most animals are not birds" with overwhelming probability:

```
true : 0.941    false : 0.059
```

An Example Inference (cont.)

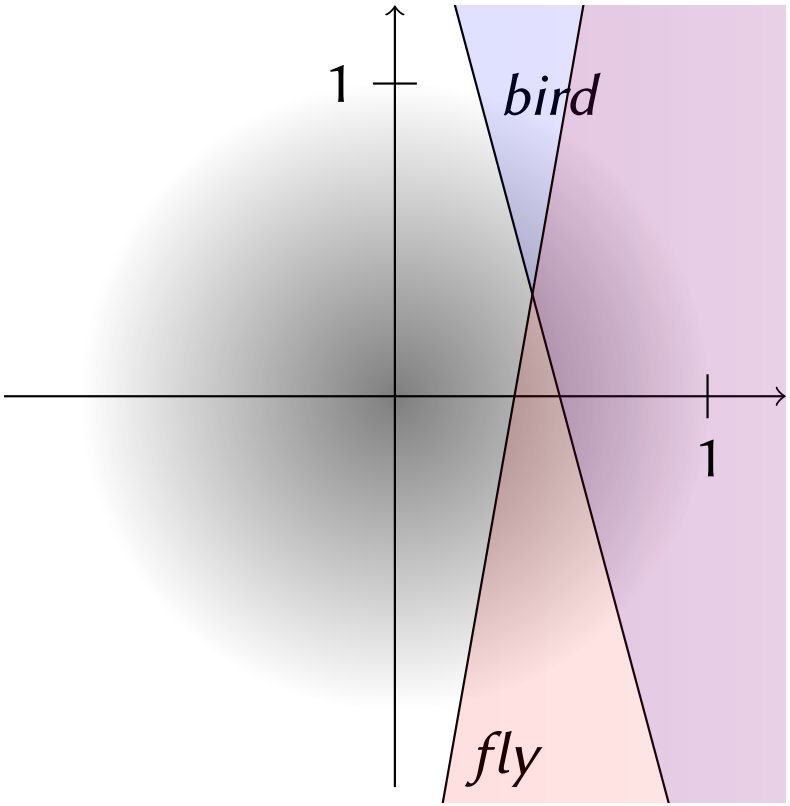


An Example Inference (cont.)



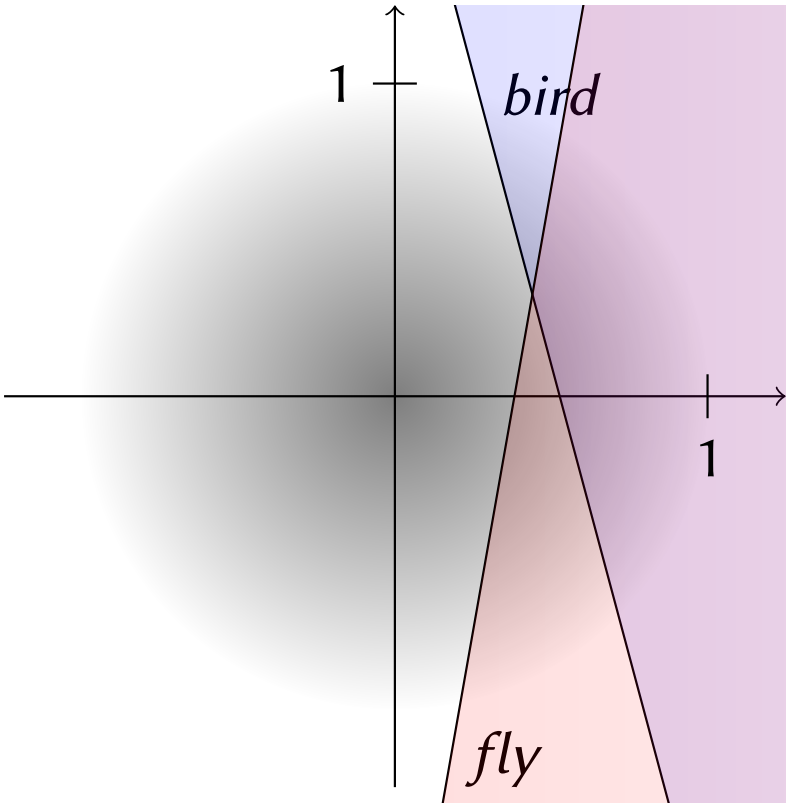
- We may assume that "animal" holds for every individual.

An Example Inference (cont.)



- We may assume that “animal” holds for every individual.
- “Most animals don’t fly” implies that “fly” has a large bias.

An Example Inference (cont.)



- We may assume that “animal” holds for every individual.
- “Most animals don’t fly” implies that “fly” has a large bias.
- “Most birds fly” can be satisfied only if “fly” is highly correlated with “bird”, and if the bias of “bird” is even greater than that of “fly”.

Semantic Learning

Our model can adapt to new observations, giving rise to learning.
Consider the data taken from `https://en.wikipedia.org/wiki/Naive_Bayes_classifier`.

Person	weight (lbs)
male	180
male	190
male	170
male	165
female	100
female	150
female	130
female	150

Semantic Learning

Our model can adapt to new observations, giving rise to learning.
Consider the data taken from `https://en.wikipedia.org/wiki/Naive_Bayes_classifier`.

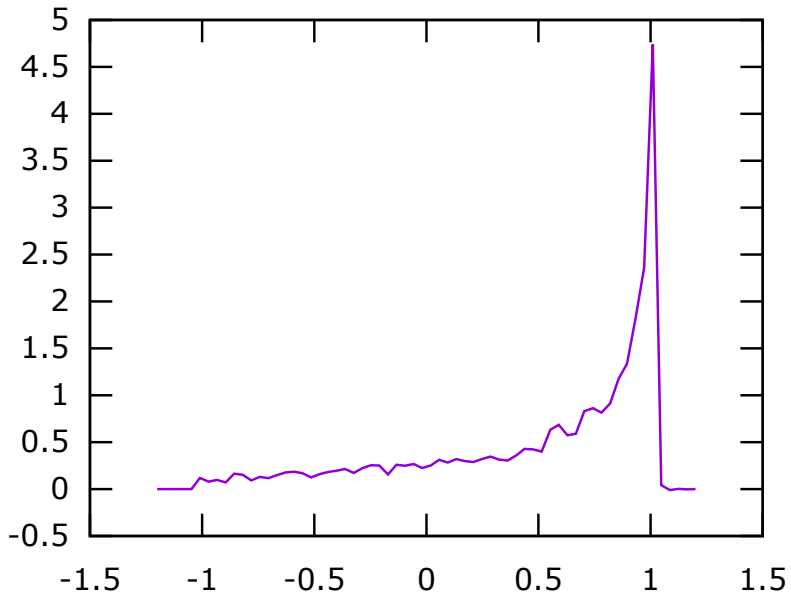
Person	weight (lbs)
male	180
male	190
male	170
male	165
female	100
female	150
female	130
female	150

When given the person and weight data, our model predicts that an individual of weight 190 is male with the following probabilities.

```
true : 0.57805    false : 0.42195
```

Semantic Learning (cont.)

By directly measuring the cosine of the angle between the weight and male vector, we get the following distribution, indicating a strong correlation:



Future Work

In future work we expect to

- extend the syntactic and semantic coverage of our framework;
- improve our modelling and sampling mechanisms to accommodate large scale applications more efficiently and robustly;
- develop our Bayesian learning theory to handle more complex cases of classifier acquisition.

Conclusions

- a compositional Bayesian semantics, interpreting declarative sentences by assigning them probability conditions
- the conditional probability of a sentence is the likelihood that an idealised speaker would accept the assertion
- assessing the probability of a sentence is an instance of evaluating the application of a classifier to a new argument
- straightforward treatments of vagueness in predication, gradable predicates, comparatives, generalised quantifiers, and probabilistic inferences across several property dimensions

References

- Borgström, Johannes et al. (2013). "Measure Transformer Semantics for Bayesian Machine Learning". In: *Logical Methods in Computer Science* 9, pp. 1–39.
- Goodman, Noah D. and Andreas Stuhlmüller (2014). *The Design and Implementation of Probabilistic Programming Languages*. http://dippl.org. Accessed: 2018-4-17.
- Goodman, Noah D. et al. (2008). "Church: a Language for Generative Models". In: *Proceedings of the 24th Conference Uncertainty in Artificial Intelligence (UAI)*, pp. 220–229.

Thank you!