Towards Confluence: Model Building for First Order Abelian Logic using Backpropagation and Stochastic Gradient Descent

Daoud Clarke and Bill Keller

August 27, 2015

1 Introduction

In this paper we apply techniques used for learning neural networks to build logical models. Specifically, we use backpropagation and stochastic gradient descent to learn models for a new logic, first order Abelian logic, which we also describe. The motivation for this work is natural language semantics in which words are often represented as vectors. The models we learn represent predicates as vectors, or more generally, tensors.

2 First order Abelian logic

First order Abelian logic is intended to extend the predicate logic of Meyer and Slaney (1989) with first order quantifiers, however we leave the proof of a formal relationship between their logic and ours to further work. Moreover, we use vector spaces for models instead of the more general Abelian groups they use, since the continuity properties are more suited to the techniques we wish to apply.

The language of first order Abelian logic is that of standard first order logic; for simplicity we exclude equality, functions and free variables. We denote variables $x_1, x_2, \ldots$. Its semantics are as follows:

- The interpretation of an n place predicate symbol P is a tensor T_P of order n; zero-place predicates (constants) are zero order tensors (scalars).
- The interpretation of a formula is an *assigned tensor*, i.e. a pair $\langle T, A\rangle$, where T is a tensor of order n and A is an n-tuple of integers describing variable assignment.
- The interpretation of a formula $P(x_{i_1}, x_{i_2}, \ldots x_{i_n})$ where P is an n-place predicate symbol is the assigned tensor $\langle T_P, \langle i_1, i_2, \ldots i_n\rangle\rangle$. For example the formula $loves(x_2, x_3)$ would have the assigned tensor $\langle T_{loves}, \langle 2, 3\rangle\rangle$.

- The interpretation of $\alpha \circ \beta$, where $\circ$ is either $\wedge$, $\vee$ or $\rightarrow$ is $[\![\alpha]\!] \bullet [\![\beta]\!]$, where $\bullet$ is the *assigned tensor unification* operation corresponding to $\circ$, described below, and $[\![\alpha]\!]$ denotes the interpretation of the formula α.

- The interpretation of $\forall x_i\ \alpha$ (or $\exists x_i\ \alpha$) where $[\![\alpha]\!] = \langle T, A\rangle$ and T is of order n is an assigned tensor $\langle T', A'\rangle$ where T' is of order $n-1$ and consists of the component-wise minimum (component-wise maximum) of all slices of the tensor in dimension j, where j is the index of A in which i occurs and A' is A with i removed. For example, suppose the predicate *loves* is described by the order 2 tensor

$$T_{loves} = \begin{pmatrix} 0.1 & 0 & -0.2 \\ 0.2 & 0.5 & 0 \\ -0.3 & 0.1 & 0 \end{pmatrix}$$

Then $[\![loves(x_3, x_2)]\!] = \langle T_{loves}, \langle 3, 2\rangle\rangle$ and $[\![\exists x_2\ loves(x_3, x_2)]\!]$ is $\langle T'_{loves}, \langle 3\rangle\rangle$, where

$$T'_{loves} = \begin{pmatrix} 0.1 \\ 0.2 \\ -0.3 \end{pmatrix} \sqcup \begin{pmatrix} 0 \\ 0.5 \\ 0.1 \end{pmatrix} \sqcup \begin{pmatrix} -0.2 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0.1 \\ 0.5 \\ 0.1 \end{pmatrix}$$

and $\sqcup$ denotes the component-wise maximum. Note that we perform the operation over the *columns* of the tensor since the quantification is over x_2 and 2 occurs in the second position of A; if the quantification had been over x_3 we would have used the rows.

2.1 Assigned tensor unification

We define the operations $\sqcup$, $\sqcap$ and $-$ between assigned tensors, these are used in interpretations in place of the logical symbols $\vee$, $\wedge$ and $\leftarrow$ respectively. Let $\langle T, A\rangle$ and $\langle S, B\rangle$ be two assigned tensors. We define $\langle U, C\rangle = \langle T, A\rangle \bullet \langle S, B\rangle$ where $\bullet$ is one of $\sqcup$, $\sqcap$ and $-$. Then C is the sorted union of A and B in ascending order, and U is a tensor of order $|C|$. Let $I = \langle i_1, i_2 \ldots i_m\rangle$ be an indexing of U. The components of U are

$$U_I = T_J \bullet S_K$$

where J and K are indexes defined such that $j_a = i_c$ and $k_b = i_c$ where c is the index in C of the ath element of A and b is its index in B.

For example, assume *bill* and *run* have interpretations

$$T_{bill} = \begin{pmatrix} 0.3 \\ -0.2 \\ -0.1 \end{pmatrix} \quad T_{run} = \begin{pmatrix} 0.1 \\ 0.1 \\ -0.2 \end{pmatrix}$$

Then $bill(x_1) \wedge run(x_2)$ has the interpretation

$$\langle T_{bill}, \langle 1\rangle\rangle \sqcap \langle T_{run}\langle 2\rangle\rangle = \langle U, \langle 1, 2\rangle\rangle$$

where

$$U = \begin{pmatrix} 0.3 \wedge 0.1 & -0.2 \wedge 0.1 & -0.1 \wedge 0.1 \\ 0.3 \wedge 0.1 & -0.2 \wedge 0.1 & -0.1 \wedge 0.1 \\ 0.3 \wedge -0.2 & -0.2 \wedge -0.2 & -0.1 \wedge -0.2 \end{pmatrix} = \begin{pmatrix} 0.1 & -0.2 & -0.1 \\ 0.1 & -0.2 & -0.1 \\ -0.2 & -0.2 & -0.2 \end{pmatrix}$$

However $\mathit{bill}(x_1) \wedge \mathit{run}(x_1)$ has the interpretation $\langle U', \langle 1 \rangle \rangle$ where

$$U' = \begin{pmatrix} 0.3 \wedge 0.1 \\ -0.2 \wedge 0.1 \\ -0.1 \wedge -0.2 \end{pmatrix} = \begin{pmatrix} 0.1 \\ -0.2 \\ -0.2 \end{pmatrix}$$

References

R. K. Meyer and J. K. Slaney. Abelian logic (from A to Z). In G. Priest, R. Routley, and J. Norman, editors, *Paraconsistent Logic: Essays on the Inconsistent*, pages 245–288. Philosophia Verlag, 1989.