

Learning FOL

Alexander Pluska

January 10, 2023

1 Multi-sorted first-order logic

1.1 Syntax

We consider a countably infinite set of sorts S, T, \dots and for each sort a countably infinite set of variables x, y, \dots . Furthermore for each $n+1$ -tuple of sorts S_0, \dots, S_n we define a countably infinite set of function symbols f, g, \dots , we say that those functions are of sort $S_0 \times \dots \times S_{n-1} \rightarrow S_n$. A *term* of sort S is defined inductively as follows:

- A variable x of sort S is a term of sort S .
- Given a function f of sort $S_0 \times \dots \times S_{n-1} \rightarrow S_n$ and n terms t_0, \dots, t_{n-1} of sorts S_0, \dots, S_{n-1} , $f(t_0, \dots, t_{n-1})$ is a term of sort S_n .

Given two terms t, u of the same sort S , $t = u$ is an equation. Given equations $t_0 = u_0, \dots, t_m = u_m$ and $v_0 = w_0, \dots, v_n = w_n$

$$t_0 = u_0, \dots, t_m = u_m \vdash v_0 = w_0, \dots, v_n = w_n$$

is a sequent. A set of sequents is a *theory*.

1.2 Semantics

An *interpretation* is an assignment I that maps each sort to a set, each variable x of sort S to an element $I(x) \in I(S)$ and each function f of sort $S_0 \times \dots \times S_{n-1} \rightarrow S_n$ to a function $I(f) : I(S_0) \times \dots \times I(S_{n-1}) \rightarrow I(S_n)$. As usual t is canonically extended to terms. We write t^I for $I(t)$.

We say that an interpretation I *satisfies* an equation $t = u$ if $t^I = u^I$. We say that an interpretation I *satisfies* a sequent $t_0 = u_0, \dots, t_m = u_m \vdash v_0 = w_0, \dots, v_n = w_n$ if it does not satisfy one of $t_0^I = u_0^I, \dots, t_m^I = u_m^I$ or it satisfies all of $v_0^I = w_0^I, \dots, v_n^I = w_n^I$. We say that an interpretation I

satisfies a theory \mathcal{T} if it satisfies all sequents in \mathcal{T} . A sequent is *valid* in a theory \mathcal{T} if every interpretation that satisfies \mathcal{T} satisfies the sequent. A theory is *consistent* if it has a model.

1.3 Logic embedding

A logic embedding is an assignment E of sorts to Euclidean spaces \mathbb{R}^n equipped with a random variable $X : \Omega \rightarrow \mathbb{R}^n$ and functions $f : S_0 \times \cdots \times S_{n-1} \rightarrow S_n$ to differentiable functions $E(f) : E(S_0) \times \cdots \times E(S_{n-1}) \rightarrow E(S_n)$. We can extend f to terms inductively.

Given a logic embedding the error of an equation $t = u$ named e is

$$\mathcal{E}(E, e) = \min(1, \|t^E - u^E\|_\infty),$$

the error of a sequent $e_0, \dots, e_m \vdash f_0, \dots, f_n$ named Γ is

$$\mathcal{E}(E, \Gamma) = \min \left(1 - \max_i (e_i), \min_j (f_j) \right),$$

the error of a theory \mathcal{T} is

$$\mathcal{E}(E, \mathcal{T}) = \sum_{\Gamma \in \mathcal{T}} \mathcal{E}(E, \Gamma).$$

1.4 Learning the embedding

The following notion is inspired by graph embeddings. Given a theory the first question we must answer is what the dimensions of different sorts are including random variables and what function spaces we consider for each function.

For the functions we shall consider MLPs with ReLU activation. The random variables will be binomially distributed over the embeddings of terms of the sort. We will set dimensions and initial parameters manually, however, we plan on experimenting with different regimes and in particular online-adjustment of these in the future.

For optimization we minimize the error of the theory. We use the Adam optimizer with a learning rate of 0.001 and a batch size of 100. We use a learning rate scheduler to decrease the learning rate by a factor of 0.1 every 1000 epochs.

1.5 Using the embedding

We utilize superposition. Superposition is usually defined on clauses, i.e. disjunctions of literals $\neg A_0 \vee \dots \vee \neg A_{m-1} \vee B_0 \vee \dots \vee B_{n-1}$. (Classically) equivalently we instead consider sequents $A_0, \dots, A_{m-1} \vdash B_0, \dots, B_{n-1}$ of atoms, which can be interpreted as $\bigwedge A_i \rightarrow \bigvee B_i$. Instead of $A_0, \dots, A_{m-1} \vdash B_0, \dots, B_{n-1}$ we write $\mathbf{A} \vdash \mathbf{B}$. A simple superposition calculus comprises the following rules:

Superposition (Sup):

$$\frac{\mathbf{A} \vdash \mathbf{B} \vee l = r \quad \mathbf{A}' \vdash \mathbf{B}' \vee s[l'] = t}{(\mathbf{A} \wedge \mathbf{A}' \vdash \mathbf{B} \vee \mathbf{B}' \vee s[r] = t)\theta} \quad \frac{\mathbf{A} \vdash \mathbf{B} \vee l = r \quad s[l'] = t \wedge \mathbf{A}' \vdash \mathbf{B}'}{(s[r] = t \wedge \mathbf{A} \wedge \mathbf{A}' \vdash \mathbf{B} \vee \mathbf{B}')\theta}$$

where θ is the mgu of l and l' and l' is not a variable.

Equality Resolution (ER) and Equality Factoring (EF):

$$\frac{l = l' \wedge \mathbf{A} \vdash \mathbf{B}}{(\mathbf{A} \vdash \mathbf{B})\theta} \quad \frac{\mathbf{A} \vdash \mathbf{B} \vee l = r \vee l' = r'}{(r = r' \wedge \mathbf{A} \vdash \mathbf{B} \vee l = r')\theta}$$

where θ is the mgu of l and l' .

Binary Resolution (BR), Positive and Negative Factoring (Fact).

$$\frac{\mathbf{A} \vdash \mathbf{B} \wedge P \quad P' \wedge \mathbf{A}' \vdash \mathbf{B}'}{(\mathbf{A} \wedge \mathbf{A}' \vdash \mathbf{B} \wedge \mathbf{B}')\theta} \quad \frac{\mathbf{A} \vdash \mathbf{B} \vee P \vee P'}{(\mathbf{A} \vdash \mathbf{B} \vee P)\theta} \quad \frac{P \wedge P' \wedge \mathbf{A} \vdash \mathbf{B}}{(P \wedge \mathbf{A} \vdash \mathbf{B})\theta}$$

where θ is the mgu of P and P' .

Superposition operates by saturation, i.e. formulas are added by the above rules until a refutation is found (i.e. the empty) or no new formulas can be added. With this superposition is semi-complete for first-order logic. Usually this process will not terminate however and a elimination ordering is utilized to guide proof search. In our case we will guide proof search a bit differently. Assume we are given a logic embedding \mathcal{E} . We perform A^* search for the empty sequent where the heuristic is determined by the error of the sequent.

1.6 Extensions

At the current stage there is one learning and one search step. Instead learning and search could be alternated. Since terms are naturally tree-like structures, an embedding into hyperbolic space instead of Euclidean could be considered.