Probabilistic Soft Logic (PSL) Tutorial

Installation and Prerequisites

PSL requires Java version 8 or newer. Make sure Java is installed before starting.

Example installation on macOS:
brew install java

Writing PSL Rules

PSL uses weighted logical rules to guide inference.

• Higher weights mean stronger influence.
• Rules follow the format: body -> head
• Multiple conditions use the symbol &
• Adding ^2 uses squared hinge-loss.
• Rules ending with a period without a weight are hard constraints.

Example:

1.0: Knows(A,B) & Likes(A,X) -> Likes(B,X) ^2
0.5: ~Likes(A,B) -> ~Likes(B,A) ^2
Knows(A,B) = Knows(B,A).

Project Setup

PSL source code:
https://github.com/linqs/psl

Example projects:
https://github.com/linqs/psl-examples

Typical project files include:

• model.psl – rules and predicates
• model.data – data configuration

Predicates and Data

Predicates can be either closed or open.

Closed predicates:

• Fully observed
• Do not change during inference

Open predicates:
• Values are inferred by PSL

Observation files:

• Stored as tab-separated text
• May include truth values between 0.0 and 1.0

Example configuration:

predicates:
Knows/2: closed
Likes/2: open

observations:
Knows: knows_obs.txt
Likes: likes_obs.txt

targets:
Likes: likes_targets.txt

truth:
Likes: likes_truth.txt

Running Inference

PSL performs Most Probable Explanation (MPE) inference.

Example command:

java -jar psl-cli.jar --infer --model model.psl --data model.data --output results/

Example Results

Example observations:

knows_obs.txt
Alice Bob
Bob Carol

likes_obs.txt
Alice Pizza 1.0
Alice Sushi 0.8

Example inferred results:
Bob Pizza 0.91

Bob Sushi 0.74
Carol Pizza 0.63

Summary

Probabilistic Soft Logic combines logical rules with probabilistic inference.

• Uses weighted rules
• Supports uncertainty
• Works with relational data
• Produces truth values between 0 and 1