Mapping the Internet
Modelling Entity Interactions in Complex Heterogeneous Networks
Ing. Šimon Mandlík, Supervisor: doc. Ing. Tomáš Pevný, Ph.D

Motivation
In many domains, application of standard machine learning methods on modern data sources is still hindered due to:
• Unrealistic assumption about independence and identical distribution of input data
• Unknown set of informative features to represent samples
• Heterogeneous or hierarchical nature of the data
• Missing data on various levels of abstraction
• Insufficient scalability
• Unsatisfactory explainability and interpretability

HMill framework
• Hierarchical Multi-instance Learning Library
• general-purpose, unified framework for sample representation and model definition
• high modelling flexibility and overall versatility

Classical machine-learning approach:
extract features → \( x = (x_1, x_2, x_3, x_4) \) → send to classifier → \( f(x) = \begin{cases} 
Iris\ setosa \\
Iris\ virginica \\
Iris\ versicolor 
\end{cases} \)

HMill approach:

Array nodes
For modelling lowest-level raw observations
\[
\begin{align*}
\varphi_1 &= FALSE \quad h_{\text{oh}} \text{ an}( ) \\
\varphi_2 &= \text{“Anna“} \quad h_{\text{n-gram}} \text{ an}( ) \\
\varphi_3 &= \begin{pmatrix} 3.14 \\ 0 \\ 42 \end{pmatrix} \quad h_1 \text{ an}( ) \\
\varphi_4 &= \bullet \quad h_{\text{oh}} \text{ an}( )
\end{align*}
\]

Bag nodes
For modelling compact sets of probability measures
\[
\text{bn}(\{x_1, x_2\}) \quad f_{\tau} \\
\text{bn}(\{x_1, x_2\}) \quad f_{\tau} \\
\text{bn}(\{x_1, x_2\}) \quad f_{\tau} \\
\text{bn}(\{x_1, x_2\}) \quad f_{\tau}
\]

Product nodes
For modelling Cartesian products
\[
\text{pn}(\{x_1, x_2\}) \quad f_1 \\
\text{pn}(\{x_1, x_2\}) \quad f_2 \\
\text{pn}(\{x_1, x_2\}) \quad f_3 \\
\text{pn}(\{x_1, x_2\}) \quad f_4
\]

HMill traits
• theoretically justified (extension of the UA theorem)
• efficient batching and gradient computation
• elegant dealing missing data
• convenient sampling techniques for large inputs

Real-world use cases
• framework tested on three completely different tasks
• cybersecurity domain — very relevant and difficult for ML
• baseline models achieved comparable or better performance than specialized methods on all three tasks

Use case: Classifying IoT device over network
• classifying the type of IoT device
• based on measurements obtainable by network scanning
• structured, hierarchical and heterogeneous data
• some items are missing
• input: JSON/XML documents
• Avast data
• HMill performs better on the provided dataset than a specialized method

Use case: Detecting malware with behavioral graphs
• detecting malicious binary files
• based on the behavior in Windows OS
• input: snapshot of the OS represented as a graph
• nodes represent files and processes
• edges represent interaction between files and processes
• data obtained from Avast
• HMill more accurate than methods ignoring relations

Use case: Harmful domain detection from relations
• detecting harmful domains
• input: binary relations
• example: domain D in relation with binary B, because B connected to D
• Cisco cooperation
• HMill performs better on the provided dataset than a specialized method

Conclusion
• HMill offers high versatility with no performance compromises
• excels at automated, Auto ML style approach to learning from real-world data
• out-of-the-box availability and little to no preprocessing needed enable application to many problems
• implementation available at https://github.com/pevnak/Mill.jl