My Favorite Algorithm

Fabian Hadiji

Lehrstuhl für Künstliche Intelligenz, TU Dortmund

DSDay7, Berlin, October 30, 2014

Joint work with Kristian Kersting
My Favorite Algorithm?

Desirable Properties:
My Favorite Algorithm?

Desirable Properties:
- Should be widely applicable!
My Favorite Algorithm?

Desirable Properties:
- Should be widely applicable!
- Should scale well!

Label Propagation
[Zhu and Ghahramani, 2002, Zhu et al., 2003]
My Favorite Algorithm?

Desirable Properties:

- Should be widely applicable!
- Should scale well!
- Should have theoretic guarantees!
My Favorite Algorithm?

Desirable Properties:
- Should be widely applicable!
- Should scale well!
- Should have theoretic guarantees!
- Should need only a few lines of code!
My Favorite Algorithm?

Desirable Properties:
- Should be widely applicable!
- Should scale well!
- Should have theoretic guarantees!
- Should need only a few lines of code!

Label Propagation
[Zhu and Ghahramani, 2002, Zhu et al., 2003]
Label Propagation — Intuition

- Set of nodes
Label Propagation — Intuition

- Set of nodes
- Set of known labels

\[
\text{Similarity function} \quad \text{e.g. exp}\left( -\sum \left( \frac{x_{id} - x_{jd}}{\sigma^2} \right)^2 \right)
\]

Iteratively propagate labels

Read off labels
Label Propagation — Intuition

- Set of nodes
- Set of known labels
- Similarity function
  - e.g. \( \exp \left( - \sum_d \frac{(x_{id} - x_{jd})^2}{\sigma_d^2} \right) \)
Label Propagation — Intuition

- Set of nodes
- Set of known labels
- Similarity function:
  \[ \text{e.g. } \exp \left( -\sum_d \frac{(x_{id} - x_{jd})^2}{\sigma_d^2} \right) \]
- Iteratively propagate labels

\[
\begin{array}{c|c}
\text{Node} & \text{Label} \\
\hline
0.67/0.00 & 0.00/0.00 \\
0.00/0.33 & 0.00/0.00 \\
0.77/0.06 & 0.33/0.17 \\
0.11/0.50 & 0.29/0.71 \\
0.86/0.14 & 0.57/0.43 \\
0.29/0.71 & 0.00/0.00 \\
\end{array}
\]
Label Propagation — Intuition

- Set of nodes
- Set of known labels
- Similarity function
  - e.g. \( \exp \left( - \sum_d \frac{(x_{id} - x_{jd})^2}{\sigma_d^2} \right) \)
- Iteratively propagate labels
Label Propagation — Intuition

- Set of nodes
- Set of known labels
- Similarity function
  - e.g. \( \exp \left( - \sum_d \frac{(x_{id} - x_{jd})^2}{\sigma_d^2} \right) \)
- Iteratively propagate labels
Label Propagation — Intuition

- Set of nodes
- Set of known labels
- Similarity function
  - e.g. $\exp\left(-\sum_d \frac{(x_{id} - x_{jd})^2}{\sigma_d^2}\right)$
- Iteratively propagate labels

![Diagram showing label propagation with nodes and edge weights](image-url)
Label Propagation — Intuition

- Set of nodes
- Set of known labels
- Similarity function
  - e.g. \( \exp \left( - \sum_d \frac{(x_{id} - x_{jd})^2}{\sigma_d^2} \right) \)
- Iteratively propagate labels
- Read off labels
# W is similarity matrix
# Y is label matrix
W = preprocess(W, Y)
Y_old = Y.copy()
iters = 0
while True:
    Y = W * Y
    max_diff = np.abs(Y-Y_old).max()
    iters += 1
    if max_diff < th:
        break
    Y_old = Y.copy()
GeoDBLP
DBLP enriched with geo-locations

- DBLP\(^1\) is a bibliography database with \(\approx 1.5\)M authors and 
  \(\approx 2.8\)M papers

\(^1\)http://dblp.uni-trier.de/
GeoDBLP
DBLP enriched with geo-locations

- DBLP\(^1\) is a bibliography database with \(\approx 1.5\)M authors and \(\approx 2.8\)M papers
- Unfortunately, DBLP does not contain affiliations/geo-locations

\(^1\)http://dblp.uni-trier.de/
GeoDBLP
DBLP enriched with geo-locations

- DBLP\(^1\) is a bibliography database with \(\approx 1.5M\) authors and \(\approx 2.8M\) papers
- Unfortunately, DBLP does not contain affiliations/geo-locations
- Obtaining seed affiliations/geo-locations is possible but challenging

\(^1\)http://dblp.uni-trier.de/
GeoDBLP
DBLP enriched with geo-locations

- DBLP\(^1\) is a bibliography database with \(\approx 1.5\)M authors and \(\approx 2.8\)M papers
- Unfortunately, DBLP does not contain affiliations/geo-locations
- Obtaining seed affiliations/geo-locations is possible but challenging
- Application: label \(\approx 5\) million author-paper-pairs form DBLP with one of \(\approx 4.5\) thousand cities

\(^1\)http://dblp.uni-trier.de/
GeoDBLP
DBLP enriched with geo-locations

- DBLP\(^1\) is a bibliography database with \(\approx 1.5\)M authors and \(\approx 2.8\)M papers
- Unfortunately, DBLP does not contain affiliations/geo-locations
- Obtaining seed affiliations/geo-locations is possible but challenging
- Application: label \(\approx 5\) million author-paper-pairs form DBLP with one of \(\approx 4.5\) thousand cities

\[
W \cdot Y = \begin{pmatrix}
  w_{11} & \ldots & w_{1n} \\
  \vdots & \ddots & \vdots \\
  w_{n1} & \ldots & w_{nn}
\end{pmatrix}
\begin{pmatrix}
  y_{11} & \ldots & y_{1k} \\
  \vdots & \ddots & \vdots \\
  y_{n1} & \ldots & y_{nk}
\end{pmatrix}
\]

\(5\text{M} \times 5\text{M} \times 5\text{M} \times 4.5\text{k}\)

\(^1\)http://dblp.uni-trier.de/
Large-scale Label Propagation

- **Problem**: Impossible to store dense affinity matrix in RAM.
- **Solution**: Use similarity function based on relational formulas [Hadiji et al., 2013]. E.g.:

\[ w_{ij} += \lambda_d \text{ if } \text{author}(i) = \text{author}(j) \land \text{year}(i) = \text{year}(j) \]
Large-scale Label Propagation

- **Problem**: Impossible to store dense affinity matrix in RAM.
- **Solution**: Use similarity function based on relational formulas [Hadiji et al., 2013]. E.g.:

  \[ w_{ij} = \lambda_d \text{ if } \text{author}(i) = \text{author}(j) \land \text{year}(i) = \text{year}(j) \]

- **Problem**: LP often suffers from slow convergence
- **Solution**: Bootstrapping to speed up convergence [Hadiji and Kersting, 2013]
Propagated Data

Initial Data
Propagated Data

Completed Data
Thank You

Questions?

www.hadiji.com  |  @fabianhadiji
