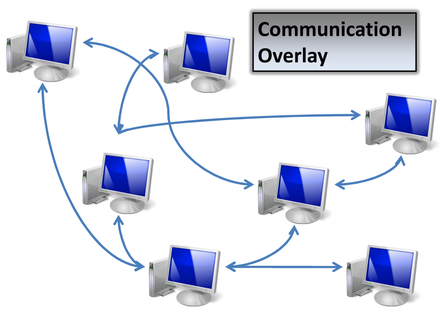
Lightning Fast Asynchronous Distributed K-Means Clustering

Árpád Berta István Hegedűs Róbert Ormándi

Research Group on Artificial Intelligence, Hungarian Academy of Sciences and University of Szeged, Szeged, Hungary

This work was supported by the European Union and the European Social Fund through project FuturICT.hu (grant no.: TAMOP-4.2.2.C-11/1/KONV-2012-0013).

Fully Distributed K-Means

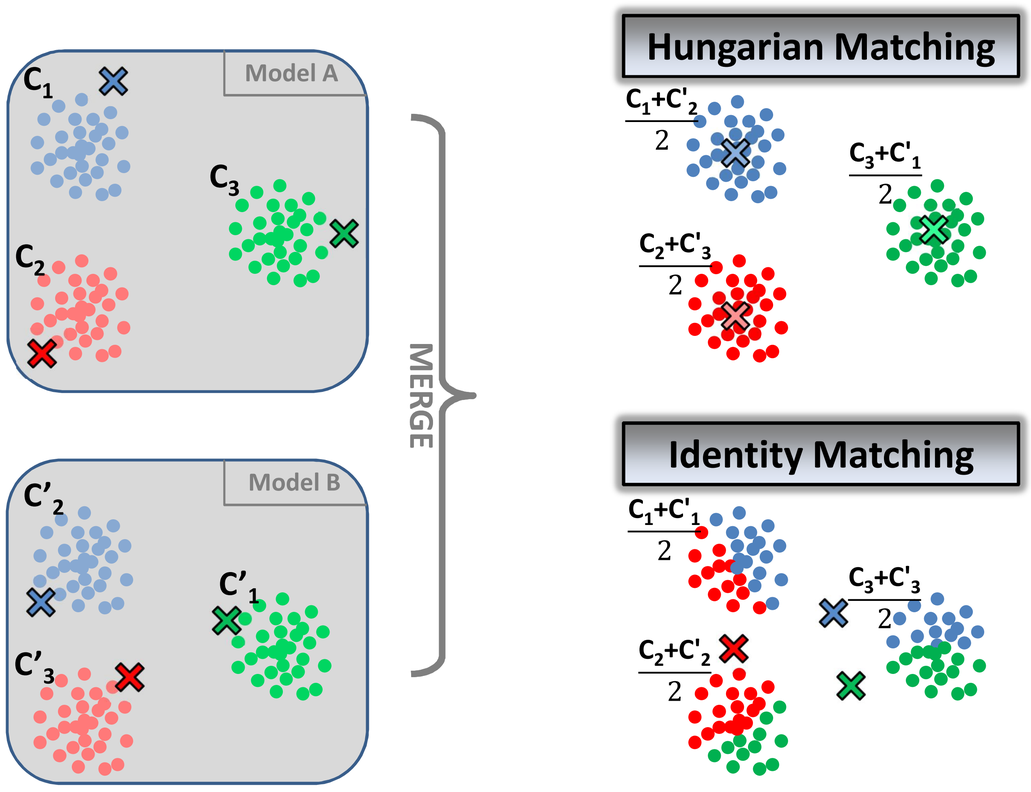


- A huge number of individual computational units (nodes)
- There is no central control
- Nodes communicate by messaging
- Every node runs the same algorithm (GoLF)
- Every node has only one data point locally
- Nodes cooperatively solve machine learning tasks

Various Merging Strategies

Merging strategies can speed up the convergence

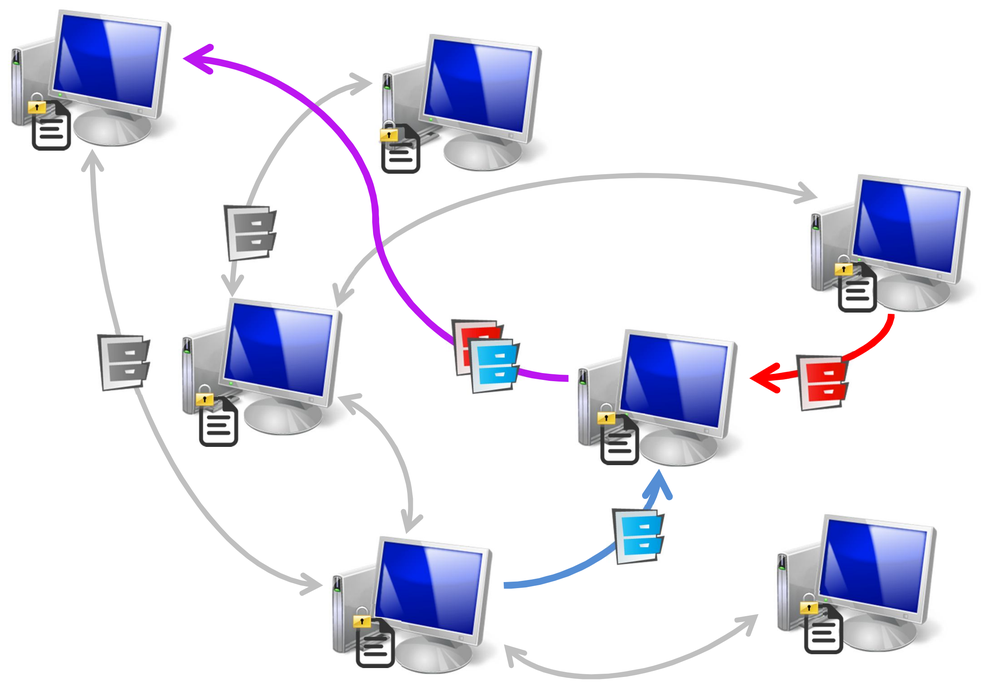
- Nodes can merge previously received models (e.g. by centroid averaging)
- **Identity Matching:** averaging centroids that having the same indices
- **Hungarian Matching:** averaging centroids by minimizing the sum of distances (solving the assignment problem by Hungarian method)



Gossip Learning Framework (GoLF)

Algorithm 1 The Learning Framework

1: $cModel \leftarrow$ initModel()
2: **loop**
3: wait(Δ)
4: $p \leftarrow$ getRandomPeer()
5: send $cModel$ to p
6: **function** onReceiveModel(m)
7: $cModel \leftarrow$ createModel($m, pModel$)
8: $pModel \leftarrow m$
9: **function** createModelNaive(m_1, m_2)
10: **return** update(m_1)
11: **function** createModelEns(m_1, m_2)
12: **return** update(merge(m_1, m_2))



Algorithm Properties:

- Asynchronous communication
- Nodes iteratively send their models
- Nodes update and store the received models
  $\Rightarrow$ Models are taking random walks in the network
- Centroid updates are based on the moving average technique
- Various merging techniques for improving convergence speed
- Low communication cost (every node sends only one message in each Δ time)
- NewsCast protocol for peer sampling service (provides the comm. overlay)
- Privacy issue: data never leaves the node (just the models)

Experiments

Simulation settings:

- Algorithms were implemented in the PeerSim simulation environment
- The models in the network were evaluated in every Δ time periods
- Message delay $[\Delta, 4\Delta]$, drop with probability 0.5
- The churn of nodes was also modeled

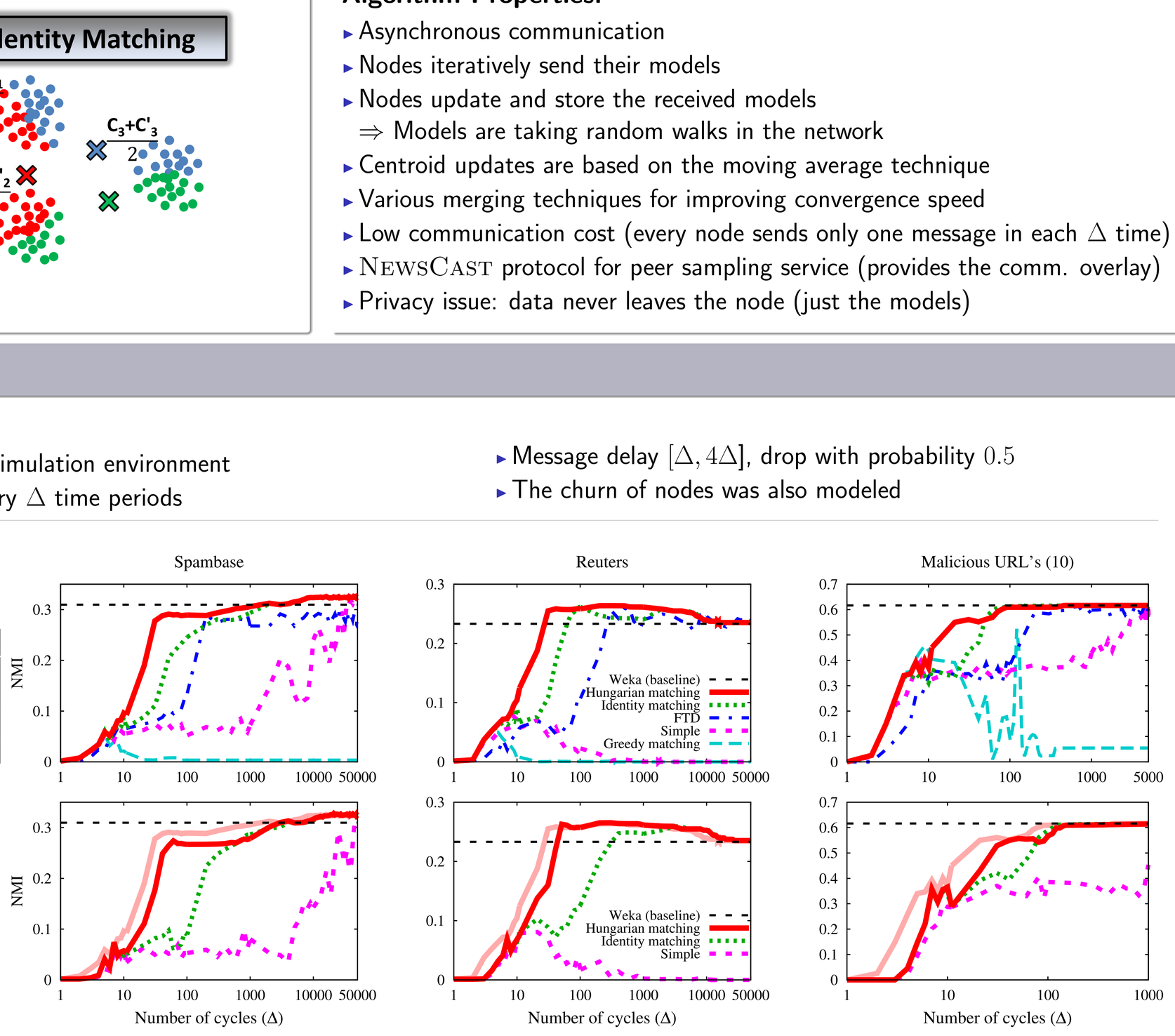
Properties of the data sets

	Reuters	SpamBase	Malicious10
Training set size	2000	4140	2155622
Test set size	600	461	240508
Number of features	9947	57	10
Classlabel ratio	1300/1300	1813/2788	792145/1603985

Evaluation Metic

$$NMI(\Omega, \mathcal{L}) = \frac{\mathcal{I}(\Omega, \mathcal{L})}{(\mathcal{E}(\Omega) + \mathcal{E}(\mathcal{L}))/2}$$

- Ω: the set of cluster assignments
- $\mathcal{L}$: the set of labels corresponding to the evaluation set
- $\mathcal{I}$, $\mathcal{E}$: mutual information and entropy



References

[1] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In Proc. of 5th Berkeley Symposium on Math. Stat. and Prob., pages 281–297, 1967.

[2] G. Di Fatta, F. Blasa, S. Cafiero, and G. Fortino. Fault tolerant decentralised k-means clustering for asynchronous large-scale networks. J. Par. Distr. Comp., 73(3):317–329, 2013.

[3] I. Eyal, I. Keidar, and R. Rom. Distributed data clustering in sensor networks. Distributed Computing, 24(5):207–222, 2011.