Supplementary: Batch Active Learning via Coordinated Matching

Javad Azimi AZIMI@EECS.OREGONSTATE.EDU

Alan Fern AFERN@EECS.OREGONSTATE.EDU

Xiaoli Z. Fern XFERN@EECS.OREGONSTATE.EDU

Glencora Borradaile GLENCORA@EECS.OREGONSTATE.EDU

Department of EECS, Oregon State University, Corvallis, OR, 97330 USA

Brent Heeringa HEERINGA@CS.WILLIAMS.EDU

Department of Computer Science, Williams College, Williamstown, MA 01267 USA

1. Fast Updated Hungarian

We show how, given a min-cost matching M in the complete bipartite graph G on vertex sets μ and S (with $|S| < |\mu|$) how to update the min-cost matching when vertex x is deleted from μ. Of course, if x is not used in M, the matching does not change.

The two most common and asymptotically most efficient algorithms for computing a minimum-cost matching are the successive-shortest-path and Hungarian algorithms. Both algorithms maintain:

- an orientation of the graph $\bar{G}$ such that edges directed from S to μ are not in the matching and edges directed from μ to S are in the matching,
- an assignment of potentials to the vertices $\pi : V \to \Re$ such that the *reduced cost* $c^\pi(uv) = c(uv) + \pi(u) - \pi(v)$ is nonnegative for each arc uv in the oriented graph.

Both algorithms start with an empty matching and iteratively increase the size of the matching by reversing a b-to-u path P in $\bar{G}$ where b is an unmatched node of S and $u \in \mu$ is the node that is closest to b according to distances given by the reduced costs. These algorithms require $|S|$ iterations, each taking the time required to find a shortest path tree (Ahuja et al., 1993). Since the reduced costs, with respect to which the shortest path tree is found, are nonnegative, Dijkstra's algorithm may be used.

Let $\bar{G}$ and π be the oriented graph and potentials that are computed along with the min-cost matching M. Let $\bar{G}_x$ be the graph with x deleted. Suppose that bx was an edge in M. We wish to increase the size of the matching by one to include b in the matching. We can do so in the same way as described above, by finding a shortest path with respect to the reduced costs. Since $\bar{G}_x$ is a subgraph of $\bar{G}$, the reduced cost of an arc in $\bar{G}_x$ is the same as the reduced cost in $\bar{G}$ and so is nonnegative. A single call to Dijkstra's algorithm is sufficient to update the matching.

References

Ahuja, R., Magnanti, R., and Orlin, J. *Network flows: theory, algorithms, and applications*. Prentice Hall, 1993.