Incremental Multi-Dimensional Scaling

Arvind Agarwal **Jeff M. Phillips** **Hal Daumé III** **Suresh Venkatasubramanian**

School of Computing University of Utah

{arvind, jeffp, hal, suresh}@cs.utah.edu

Multi-Dimensional Scaling (MDS) is a widely used method for embedding a given distance matrix into a low dimensional space, used both as a preprocessing step for many machine learning problems, as well as a visualization tool in its own right. In this paper, we present an incremental version of MDS (iMDS). In iMDS, d-dimensional data points are presented in a stream, and the task is to embed the current d-dimensional data point into a k-dimensional space for $k < d$ such that distances from the current point to the previous points are preserved. Let $\{x_1, \ldots, x_{t-1}\} \in \mathbb{R}^k$ be the data points at time step t that have already been embedded into a k-dimensional space, $\{r_1, \ldots, r_{t-1}\}$ be the given distances computed in $\mathbb{R}^d$, then objective of iMDS is to find a point x_t:

$$x_t = \arg\min_{p \in \mathbb{R}^k} \sum_{i=1}^{t-1} \left(\|p - x_i\|_2 - r_i \right)^2. \quad (1)$$

A batch version of MDS (bMDS) finds all points at the same time, in other words, given the pairwise distances r_{ij} $\forall i, j = 1, \ldots, t$, bMDS solves: $\arg\min_{x_1,\ldots,x_t \in \mathbb{R}^k} \sum_{i=1}^{t} \sum_{j=1}^{t} (\|x_i - x_j\|_2 - r_{ij})^2$.

Motivation. Primary motivation for doing iMDS is to be able to extend MDS for out-of-sample (OOS) points [Bengio et al., 2003, Trosset and Priebe, 2008]. This OOS extension of MDS is necessary for the testing of a model trained on data projected onto a k-dimensional subspace using MDS. Such a model can be tested only by projecting the test data into the training data's space, and in the absence of the OOS extension, one is either forced to do an orthogonal projection, or use other dimensionality reduction method that gives the projection matrix i.e. PCA; none of which are optimized to preserve distances. Other motivations for doing iMDS is to do dimensionality reduction for the streaming data, useful for data mining and data base applications. Since MDS is a major component of the non-linear dimensionality reduction method (ISOMAP), it can also be used for both, a streaming ISOMAP and an OOS extension of ISOMAP. In streaming ISOMAP [Law et al., 2004], a new point affects the neighborhood structure, and therefore the geodesic distances. After updating the neighborhood structure and finding the new geodesics, updating the coordinates of the affected points is one major challenge. iMDS provides a natural solution to that problem with a complexity of $O(ct)$ where c is the number of affected points, compared to $O(t^2)$ time taken by the algorithm proposed in [Law et al., 2004]. Note that c is usually much smaller than t.

To the best of our knowledge, there has not been any previous study of iMDS that exactly optimizes (1) with guaranteed convergence. However, there exist various heuristics to solve bMDS e.g. SVD-based method, majorization-based method, Newton's method; none of these are appropriate for the incremental settings for the large dataset. The first two methods are global methods and require updating the entire distance matrix, while Newton's method is not suitable for large datasets [Bronstein et al., 2008].

Algorithm overview. We now give a brief overview of the iMDS algorithm. This algorithm is based on our recently submitted work [Agarwal et al.,], a unified algorithm for solving MDS and its different variants. That algorithm has two levels of iteration, both with guaranteed convergence. For the cost function in (1), each step of the outer level chooses one point x_t and solves (1) using all n points, not just the first $t-1$. The inner level reduces to iteratively applying two steps. First, for a guess of x_t for all $i \in [0 : n]$ set a point $\hat{x}_i$ at a distance r_i from x_i in the direction of x_t. Second, recompute the estimate of x_t as the mean of all $\hat{x}_i$. It terminates when the change in x_t is below a threshold. Although the inner level solves (1), the use as a one-pass, incremental algorithm was not studied. The focus of this work, is the use of that framework when points are presented in a stream, and only a single pass over the data is permitted. The earlier work allowed multiple passes in the outer level and all of the points were considered in the inner layer. Here we only consider the first $t-1$ points in the tth call of the inner layer. Another challenge of the incremental setting is that once a point is placed at some position, it is fixed at that position, and it can not be moved; whereas, in the non-incremental setting, points were allowed to be moved again.

Presenter: Arvind Agarwal **Category: Data Mining** **Presentation Preference: Oral**

Experiments. We study the empirical behavior of our iMDS algorithm. We perform experiments on two data sets. The first is a trivial three dimensional data set lying *exactly* on a two dimensional space. The second is a 100-dimensional data set lying *close* to a 10-dimensional subspace. Level of closeness is varied by perturbing the data in $\mathbb{R}^{100}$ by a fixed fraction ϵ of the data width.

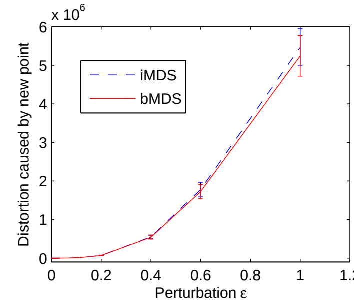
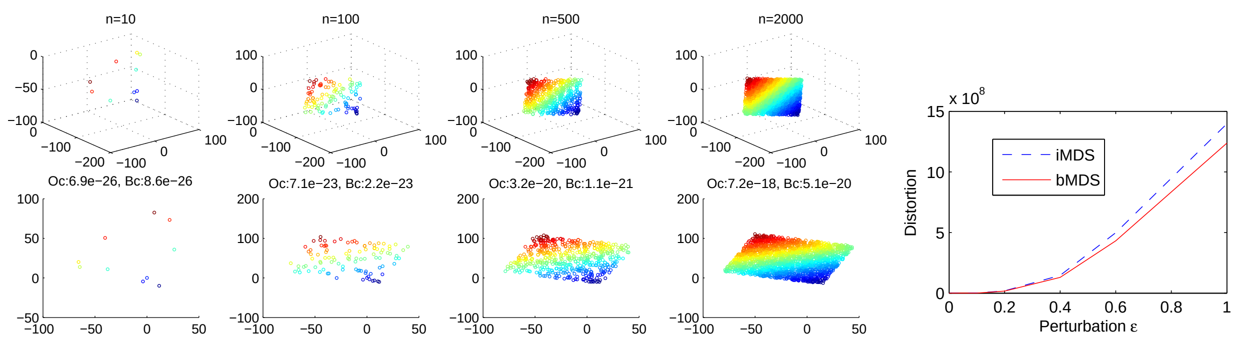


Figure 1: Distortion caused by the new point in OOS extension for iMDS and bMDS.

We first perform experiments for an OOS extension of a given embedding. In OOS extension, an initial embedding of 500 points is learned for the 100-dimensional data using MDS, and then a new point is added to this embedding by iMDS. bMDS is run again over all the data points including the new point to see the relative behavior. Figure 1 shows the distortion caused by the new point for iMDS and bMDS for different $\epsilon = \{0, 0.02, 0.1, 0.2, 0.4, 0.6, 1\}$. Both algorithms use the same stopping criterion and numbers are recorded at the same convergence level. We observe that when there is small distortion (small ϵ) during the embedding iMDS performs as well as the batched version, but as the distortion increases the batched algorithm has a small, yet noticeable, advantage. This is an expected behavior because bMDS optimizes the global error while iMDS only optimizes the local error.

Next, we perform experiments for the streaming setting. For the first data set, Figure 2(a) shows that our incremental algorithm is able to recover the two dimensional manifold nearly exactly. Figure 2(b) shows the relative behavior of iMDS and bMDS algorithms for the second, 100-dimensional data set for different ϵ for 1000 data points. We observe that the streaming setting has the similar behavior as the OOS extension, only now the difference between iMDS and bMDS is larger because the the iMDS version does not have the benefit of the good initial embedding learned by bMDS in the OOS extension.



(a) (b)

Figure 2: (a) Demonstration of iMDS for $t = \{10, 100, 500, 2000\}$. Top row is the actual data in 3-dimensions and bottom is the corresponding embedding in 2-dimensions. In bottom row, each column is titled with iMDS cost (Oc) and bMDS cost (Bc). (b) Performance comparison of incremental and batch MDS for different levels of perturbation.

Reducing memory footprint. The current version of the algorithm maintains the entire history of all the previously embedded points. In k-dimensional Euclidean space, only $k+1$ points (that do not all lie in a subspace) are required to define that space and to find the exact position of each incremental point. For robustness to noise, we desire slightly more points. Our experiments have shown that maintaining only $2k$ points is usually sufficient to get performance nearly as good as maintaining all points, but runs much faster.

References

[Agarwal et al.,] Agarwal, A., Phillips, J. M., and Venkatasubramanian, S. A Unified Algorithmic Framework for Multi-Dimensional Scaling. submitted to KDD 2010, `http://www.cs.utah.edu/~jeffp/papers/sphericalMDS.pdf`.

[Bengio et al., 2003] Bengio, Y., Paiement, J.-F., and Vincent, P. (2003). Out-of-Sample Extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering. In *In Advances in Neural Information Processing Systems*, pages 177–184. MIT Press.

[Bronstein et al., 2008] Bronstein, A. M., Bronstein, M. M., and Kimmel, R. (2008). *Numerical Geometry of Non-Rigid Shapes*. Springer.

[Law et al., 2004] Law, M. H. C., Zhang, N., and Jain, A. K. (2004). Nonlinear Manifold Learning for Data Stream. In *In Proc. SIAM International Conference for Data Mining*, pages 33–44.

[Trosset and Priebe, 2008] Trosset, M. W. and Priebe, C. E. (2008). The out-of-sample problem for classical multidimensional scaling. *Comput. Stat. Data Anal.*, 52(10):4635–4642.