FAST TENSOR-BASED MULTI-VIEW CLUSTERING WITH ANCHOR PROBABILITY TRANSITION MATRIX

Anonymous authors
Paper under double-blind review

1 APPENDIX

1.1 COMPARISON OF COMPUTATIONAL COMPLEXITY AND RUNNING TIME ACROSS METHODS

We compare the computational complexity and running time of different multi-view clustering methods to demonstrate the efficiency of our proposed Fast Tensor-Based Multi-View Clustering with Anchor Probability Transition Matrix (FTMVC-APTM). The table below presents the time complexity and running times for several popular methods on datasets such as MSRC, BBCSport, MNIST, and Yale.

As shown in the table, traditional methods such as MvLRSSC and GMC have significantly higher time complexity due to their reliance on complex graph construction and post-processing steps. For instance, on the MNIST dataset, MvLRSSC requires 1528.76 seconds, and GMC takes 1601.73 seconds, which highlights their inefficiency in handling large datasets. Even methods like FPMVS-CAG, which reduce some computational load, still require substantial time (e.g., 139.26 seconds on MSRC) due to post-processing.

In contrast, our proposed FTMVC-APTM avoids the need for post-processing and leverages anchor probability transition matrices to simplify the clustering process. As a result, FTMVC-APTM achieves much lower computational complexity and faster running times across all datasets. For example, on the MSRC dataset, FTMVC-APTM runs in only 0.39 seconds, compared to 0.75 seconds for FastMICE and 18.33 seconds for MvLRSSC. Similarly, FTMVC-APTM outperforms other methods on the BBCSport and Yale datasets, with running times of just 0.47 seconds and 0.06 seconds, respectively, further demonstrating its superior efficiency.

This comparison clearly shows that FTMVC-APTM offers a significant improvement in both computational complexity and running time, making it a highly efficient solution for multi-view clustering tasks.

Table 1: **Comparison of computational complexity and running time across methods**

Method	Time Complexity	Post-process	MSRC	BBCSport	MNIST	Yale
FastMICE	$O(nm^{1/2}V^{1/2} + nc^2t + m^3)$	N	0.75	10.49	6.18	1.89
MvLRSSC	$O(tn^3V) + O(n^3)$	Y	18.33	2.44	1528.76	3.02
RMSL	$O(t(n^3 + (L_1 + L_2)D_1Vn^2 + L_2Kn^2))$	Y	22.91	9.67	947.54	7.25
GMC	$O(t(n^3 + n^2m))$	N	19.47	27.38	1601.73	3.79
FPMVS-CAG	$O(t(nm^2 + nm^3))$	Y	139.26	10.72	17.64	5.76
MVFCAG	$O(nm^2c + nm)$	N	0.57	0.05	1.10	0.40
MVC-DMF-PA	$O(Vmt(dl^2 + nd^2)) + O(n^3)$	Y	4.59	5.24	7.63	0.12
Orth-NTF	$O(Vnkm + Vm^2k)$	N	0.89	1.09	2.91	0.80
MVC-DNTF	$O(V(nmd + d^2n + m^2k + nk))$	N	7.13	4.20	8.06	3.00
Ours	$O(nm^2cV + nmcV + V^2cn)$	N	0.39	0.47	3.81	0.06