DETAILED PROOFS

Proof of Lemma 4.1.

Proof. First note that $L(A^g) = D(A^g) - A^g = (D(A) - D(A^c)) - (A - A^c) = L(A) - L(A^c)$. Thus, Eq. (2) can equivalently be written as $\mathrm{Tr}(H^T \cdot L(A^g) \cdot H) = \mathrm{Tr}(H^T \cdot (L(A) - L(A^c)) \cdot H) = \mathrm{Tr}(H^T \cdot L(A) \cdot H) - \mathrm{Tr}(H^T \cdot L(A^c) \cdot H)$.

Given H, the term $\mathrm{Tr}(H^T \cdot L(A) \cdot H)$ is constant. Thus, minimizing the previous term is equivalent to maximizing $\mathrm{Tr}(H^T \cdot L(A^c) \cdot H)$.

Let y_k be a column vector of H. Noticing that (see [19]) $y_k^T L(A^c) y_k = \sum_{i,j} \frac{1}{2} \cdot a^c_{i,j} \cdot (y_{k,i} - y_{k,j})^2$, and exploiting orthogonality it follows: $\mathrm{Tr}(H^T \cdot L(A^c) \cdot H) = \sum_k \sum_{i,j} \frac{1}{2} \cdot a^c_{i,j} \cdot (y_{k,i} - y_{k,j})^2 = \sum_{i,j} \frac{1}{2} \cdot a^c_{i,j} \cdot \left\|h_i - h_j\right\|_2^2$, where the last step used $y_{k,i} = h_{i,k}$.

To ensure that A^c as well as A^g are non-negative, it holds $0 \leq a^c_{i,j} \leq a_{i,j}$. Thus, if $a_{i,j} = 0$ then $a^c_{i,j} = 0$. Exploiting this fact and the symmetry of the graph leads to $\sum_{i,j} \frac{1}{2} \cdot a^c_{i,j} \cdot \left\|h_i - h_j\right\|_2^2 = \sum_{(i,j)\in\mathcal{E}} a^c_{i,j} \cdot \left\|h_i - h_j\right\|_2^2$.

Next, we show that there exists a solution where each $a^c_{i,j} \in \{0, a_{i,j}\}$. As known, $0 \leq a^c_{i,j} \leq a_{i,j}$. Let $M = [a^c_e]_{e\in\mathcal{E}}$ be a maximum of Eq. (3) where some $a^c_{i,j} > 0$ but $< a_{i,j}$. Let M' be the solution where this entry is replaced by $a^c_{i,j} = a_{i,j}$. Since only $\|.\|_0$ constraints are used, M and M' fulfill the same constraints. Since $\left\|h_i - h_j\right\|_2^2$ is non-negative, $f_1(M') \geq f_1(M)$. It follows, that a solution minimizing Eq. (2) can be found by investigating $a^c_{i,j} = 0$ or $a^c_{i,j} = a_{i,j}$ only. □

Proof of Lemma 5.1.

Proof. The goal is to find a matrix A^g whose sum of the first k eigenvalues is minimal (and fulfills the given constraints). Since, however, A^g is not known, we refer to the principle of eigenvalue perturbation.

Let A^t be the matrix obtained in the previous iteration of the alternating optimization and let y_i be the ith generalized eigenvector of $L(A^t)$ (these are the *columns* of the matrix H from above, i.e. $y_{i,j} = h_{j,i}$). Furthermore, denote the corresponding eigenvalues with λ_i. We define $L(A^g) - L(A^t) =: \Delta L$ and $D(A^g) - D(A^t) = \Delta D$.

Based on the theory of eigenvalue perturbation [18], the eigenvalue λ_i^g of $L(A^g)$ can be approximated by

$$\lambda_i^g \approx \lambda_i + y_i^T \cdot (\Delta L - \lambda_i \cdot \Delta D) \cdot y_i$$

$$= \lambda_i + y_i^T \cdot ((L(A^g) - L(A^t)) - \lambda_i \cdot (D(A^g) - D(A^t))) \cdot y_i$$

Using the fact that $L(A^g) = L(A) - L(A^c)$ and $D(A^g) = D(A) - D(A^c)$, and after rearranging the terms, we obtain

$$\lambda_i^g \approx \overbrace{\lambda_i + y_i^T \cdot ((L(A) - L(A^t)) - \lambda_i \cdot (D(A) - D(A^t))) \cdot y_i}^{=:c_i} - \underbrace{y_i^T \cdot ((L(A^c)) - \lambda_i \cdot (D(A^c)) \cdot y_i}_{=:g_i}$$

Since c_i is constant, minimizing λ_i^g is equivalent to maximizing g_i. Simplifying yields:

$$g_i = y_i^T \cdot L(A^c) \cdot y_i - \lambda_i \cdot y_i^T \cdot D(A^c) \cdot y_i$$

$$= \sum_{j,j'} \frac{1}{2} a^c_{j,j'} (y_{i,j} - y_{i,j'})^2 - \lambda_i \sum_j y_{i,j}^2 \cdot d_j^c$$

where $d_j^c = [D(A^c)]_{j,j} = \sum_{j'} a^c_{j,j'}$. Thus

$$g_i = \sum_{j,j'} \frac{1}{2} a^c_{j,j'} (y_{i,j} - y_{i,j'})^2 - \lambda_i y_{i,j}^2 a^c_{j,j'}$$

$$= \sum_{j,j'} a^c_{j,j'} \left(\frac{1}{2} (y_{i,j} - y_{i,j'})^2 - \lambda_i y_{i,j}^2 \right)$$

and exploiting the symmetry of the graph, we obtain

$$g_i = \sum_{(j,j')\in\mathcal{E}} a^c_{j,j'} \left((y_{i,j} - y_{i,j'})^2 - \lambda_i y_{i,j}^2 - \lambda_i y_{i,j'}^2 \right)$$

Since the overall goal is to minimize $\sum_{i=1}^k \lambda_i^g$, we aim at maximizing

$$\sum_{i=1}^k g_i = \sum_{i=1}^k \sum_{(j,j')\in\mathcal{E}} a^c_{j,j'} \left((y_{i,j} - y_{i,j'})^2 - \lambda_i y_{i,j}^2 - \lambda_i y_{i,j'}^2 \right)$$

$$= \sum_{(j,j')\in\mathcal{E}} a^c_{j,j'} \left(\sum_{i=1}^k (y_{i,j} - y_{i,j'})^2 - \sum_{i=1}^k \lambda_i y_{i,j}^2 - \sum_{i=1}^k \lambda_i y_{i,j'}^2 \right)$$

By noticing that $y_{i,j} = h_{j,i}$ we obtain

$$= \sum_{(j,j')\in\mathcal{E}} a^c_{j,j'} \left(\underbrace{\left\|h_j - h_{j'}\right\|_2^2 - \left\|\sqrt{\lambda} \circ h_j\right\|_2^2 - \left\|\sqrt{\lambda} \circ h_{j'}\right\|_2^2}_{x} \right)$$

Note that some of the terms x might be negative. Clearly, since we aim to maximize the equation – and since $a^c_{i,j} \geq 0$ – for these terms we have to choose $a^c_{i,j} = 0$. For the remaining (non-negative) terms, the same arguments apply as in the proof of Lemma 4.1: i.e. they are either 0 or $a_{i,j}$. Thus, overall, for each term we have $a^c_e \in \{0, a_e\}$. □

Proof of Lemma 5.2

Proof. Note that $a^g_{i,j} = a_{i,j} - a^c_{i,j}$ and $d_i^g = d_i - d_i^c$. Let y_k be a column vector of H. It holds $y_k^T \cdot L_{sym}(A^g) y_k \overset{[19]}{=} \sum_{i,j} \frac{1}{2} a^g_{i,j} (\frac{y_{k,i}}{\sqrt{d_i^g}} - \frac{y_{k,j}}{\sqrt{d_j^g}})^2 = \sum_{i,j} \frac{1}{2} a^g_{i,j} (\frac{y_{k,i}^2}{d_i^g} + \frac{y_{k,j}^2}{d_j^g} - \frac{2 \cdot y_{k,i} y_{k,j}}{\sqrt{d_i^g}\sqrt{d_j^g}}) = \sum_i \frac{1}{2} y_{k,i}^2 + \sum_j \frac{1}{2} y_{k,j}^2 - \sum_{i,j} \frac{a^g_{i,j} y_{k,i} y_{k,j}}{\sqrt{d_i^g}\sqrt{d_j^g}}$. Since y_k is given, the first two terms are constant. Furthermore, due to orthogonality it holds $Tr(H^T L_{sym} H) = \sum_k y_k^T \cdot L_{sym} y_k$. Thus, minimizing the trace is equivalent to maximizing $\sum_k \sum_{i,j} \frac{a^g_{i,j} y_{k,i} y_{k,j}}{\sqrt{d_i^g}\sqrt{d_j^g}} = \sum_{i,j} \frac{a^g_{i,j}}{\sqrt{d_i^g}\sqrt{d_j^g}} h_i \cdot h_j^T$, noticing that $y_{k,i} = h_{i,j}$. Exploiting the graph's symmetry concludes the proof. □

Proof of Corollary 5.4

Proof. Adding $e = (i, j)$ to $\mathcal{X}$ has the following effects: the term a^c_e changes from 0 to a_e; the degree of the two incident nodes becomes $d_i^{\mathcal{X}\cup\{e\}} = d_i^{\mathcal{X}} - a_e$. Therefore,

$$f_3(v^{\mathcal{X}\cup\{e\}}) = f_3(v^{\mathcal{X}}) - \frac{p_e}{\sqrt{d_i^{\mathcal{X}}} \cdot \sqrt{d_j^{\mathcal{X}}}} - \sum_{\substack{(x,y)\in(\mathcal{E}_i\cup\mathcal{E}_j)\setminus\mathcal{X} \\ (x,y)\neq(i,j)}} \frac{p_{x,y}}{\sqrt{d_x^{\mathcal{X}}} \cdot \sqrt{d_x^{\mathcal{X}}}}$$

$$+ \sum_{\substack{x\neq j \\ (i,x)\in\mathcal{E}_i\setminus\mathcal{X} \\ \vee(x,i)\in\mathcal{E}_i\setminus\mathcal{X}}} \frac{p_{i,x}}{\sqrt{d_i^{\mathcal{X}} - a_e}\sqrt{d_x^{\mathcal{X}}}} + \sum_{\substack{x\neq i \\ (j,x)\in\mathcal{E}_j\setminus\mathcal{X} \\ \vee(x,j)\in\mathcal{E}_j\setminus\mathcal{X}}} \frac{p_{x,j}}{\sqrt{d_j^{\mathcal{X}} - a_e}\sqrt{d_x^{\mathcal{X}}}}$$

$$= f_3(v^{\mathcal{X}}) + s(i, a_e, \mathcal{X}) + s(j, a_e, \mathcal{X}) + \delta(e, \mathcal{X}) = f_3(v^{\mathcal{X}}) + \Delta(e, \mathcal{X})$$

Since $\mathcal{X}$ is given, $f_3(v^{\mathcal{X}})$ is constant. Thus, the edge $e \in \mathcal{E}'$ maximizing $f_3(v^{\mathcal{X}\cup\{e\}})$ is found by maximizing $\Delta(e, \mathcal{X})$. □

PSEUDOCODE AND COMPLEXITY ANALYSIS

For convenience, we provide in Algorithm 2 the excerpt of the pseudocode to compute the edge set $\mathcal{X}$ for the case of L_{sym}. This code excerpt replaces the lines 5 - 15 of Algorithm 1.

Complexity analysis: Let γ denote the number of unique edge weights per node and x the number of nearest neighbors (x-nearest neighbor graph). Using a heap, we note the following complexities:

/* Update of A^c/A^g */

1 $\mathcal{X} = \emptyset$;
2 for each node i set $count_i \leftarrow |\mathcal{E}_i| - m$;
3 priority queue PQ on tuples $(gain, edge)$;
4 for each node i and unique edge weight $a_{i,j}$ compute $s(i, a_{i,j}, \mathcal{X})$;
5 for each edge e add tuple $(\Delta(e, \mathcal{X}), e)$ to PQ ;
6 **for** $x = 1, \ldots, \theta$ *and PQ not empty* **do**
7   get first element from PQ $\rightarrow (gain, e_{best} = (i, j))$;
8   **if** $gain \leq 0$ **then** break;
9   $\mathcal{X} \leftarrow \mathcal{X} \cup \{e_{best}\}$;
10   $count_i - -$; $count_j - -$;
11   recompute $s(i, ., \mathcal{X})$ and $s(j, ., \mathcal{X})$;
12   **for** *edges* $e = (i', j')$ *incident to i and j* **do**
13   remove e from PQ ;
14   **if** $count_{i'} > 0 \wedge count_{j'} > 0$ **then**
15   recompute $\delta(e, \mathcal{X})$;
16   add tuple $(\Delta(e, \mathcal{X}), e)$ to PQ ;
17 construct A^c according to $v^{\mathcal{X}}$; $A^g = A - A^c$;

Algorithm 2: RSC for L_{sym}. Replace lines 5 - 15 of Algorithm 1 with the above code.

- line 4: $O(\gamma \cdot n \cdot x) \subseteq O(\gamma \cdot |\mathcal{E}|)$
- line 5: $O(|\mathcal{E}|)$
- line 7: $O(log(|\mathcal{E}|))$
- line 11: $O(2 \cdot \gamma \cdot x)$
- line 13: $O(log(|\mathcal{E}|))$
- line 15: $O(1)$
- line 16: $O(log(|\mathcal{E}|))$

Noticing that line 12 iterates at most $2 \cdot x$ times, and line 6 at most θ times, leads to: $O(\gamma \cdot |\mathcal{E}| + \theta \cdot (log(|\mathcal{E}|) + \gamma \cdot x + x \cdot log(|\mathcal{E}|))) = O(\gamma \cdot |\mathcal{E}| + \theta \cdot (x log(|\mathcal{E}|) + \gamma \cdot x))$. Thus, in our case $O(|\mathcal{E}| + \theta \cdot x \cdot log(|\mathcal{E}|))$.

QUALITY OF EMBEDDINGS

Measures

Let h_i denote the embedding of instance i and $c_i \in C$ its class according to the ground truth. Let $NN_x(i)$ denote the set of x nearest neighbors of i in the embedding space. Denote with $occ_x(c, i)$ the number of times the class c occurs in the neighborhood of node i (including the node itself), i.e.

$$occ_x(c, i) = |\{j \in NN_x(i) \cup \{i\} \mid c_j = c\}|$$

Then the purity is given by

$$pur_x(i) = \frac{1}{x+1} \max_{c \in C} occ_x(c, i)$$

$$PUR(x) = \frac{1}{N} \sum_{i=1}^{N} pur_x(i)$$

Denote with $C_c = \{i \mid c_i = c\}$ the set of all instances from class c. Let $P_{c,c'}$ denote the list of all pairwise distances between instances from class c to c', i.e.

$$P_{c,c'} = [dist(h_i, h_j)]_{i \in C_c, j \in C_{c'}}$$

Denote with $avg_x(L)$ the average over the $(x \cdot 100)\%$ smallest elements in a list. Then

$$P_{c,c'}(x) = avg_x(P_{c,c'})$$

and

$$GS_c(x) = \frac{P_{c,c'}(x) - P_{c,c}(x)}{\max\{P_{c,c'}(x), P_{c,c}(x)\}}$$

where $c' = \arg\min_{c' \neq c} P_{c,c'}(x)$.

Results for Global Separation

Due to space limitations, we could present in the paper the plots for two exemplary classes only. Here, we report the plots for *all* classes. As shown, for many classes, RSC performs very well. As already mentioned in the paper, naturally not every class is perfect as reflected by NMI scores of around 0.61 and 0.85.

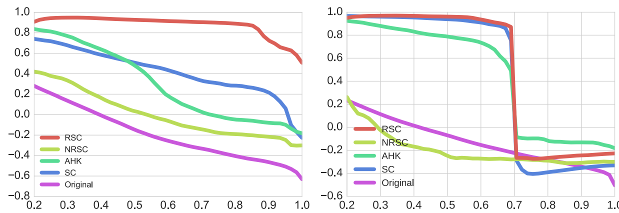

Figure 12: Banknote data (2 classes)

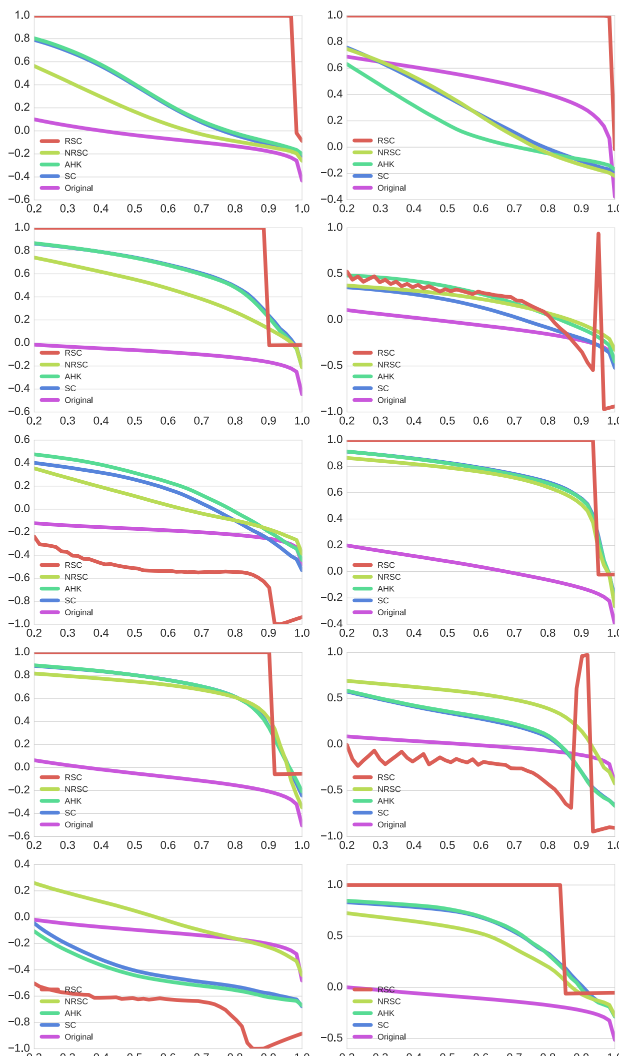

Figure 13: USPS data (10 classes)