A. Appendix

We use the same notations as in the main paper. We also use the notation $(x)_+ = \max\{0, x\}$, and $\lambda_i^{\downarrow}(A)$ (resp. $\lambda_j^{\uparrow}(A)$) for the i-th largest (resp. the j-th smallest) eigenvalue of A.

A.1. "Worst-case empirical risk" definition

We give here the definition of "worst-case empirical risk" since the term is not standard. By using the formulation of [17, Section 3.1], if we consider that the prediction rule $f_{M,\mathcal{L}_k^{n_i}}(X_i) = \operatorname{argmax}_{\hat{C}\in\mathcal{L}_k^{n_i}} \langle \hat{C}, X_i M X_i^\top \rangle$ is always a singleton, then the empirical risk of our relaxed problem is defined as (we omit the usual scale factor $\frac{1}{m}$ which is a constant):

$$R^{\Delta}(f) = \sum_{i=1}^{m} \Delta(\hat{C}_i, C_i) \text{ where } \hat{C}_i \in f_{M,\mathcal{L}_k^{n_i}}(X_i) \tag{22}$$

If there are multiple solutions (i.e. if $f_{M,\mathcal{L}_k^{n_i}}(X_i)$ is not always a singleton), the empirical risk is not well-defined. An open question is how to select $\hat{C}_i$ from multiple possible solutions. Since we are more interested in learning M than predicting $\hat{C}_i$ in our paper, we leave that question aside. We then take into account the fact that the prediction rule may not be a singleton and consider what we call the "worst-case empirical risk":

$$\sum_{i=1}^{m} \max_{\hat{C}_i \in f_{M,\mathcal{L}_k^{n_i}}(X_i)} \Delta(\hat{C}_i, C_i) = \sum_{i=1}^{m} \max_{\hat{C}_i \in \mathcal{L}_k^{n_i}} (\Delta(\hat{C}_i, C_i) + \iota_i(M; \hat{C}_i)) \tag{23}$$

where ι_i is defined in Eq. (14). Among all the possible predictions in $f_{M,\mathcal{L}_k^{n_i}}(X_i)$, we then consider the prediction $\hat{C}_i$ that returns the largest (i.e. worst) possible Δ value.

We note that Eq. (22) and Eq. (23) are equivalent if $f_{M,\mathcal{L}_k^{n_i}}$ is always a singleton.

Eq. (13) then optimizes $M \succeq 0$ to minimize this "worst-case empirical risk".

A.2. Convex upper bound of our problem

We show here that the convex surrogate in Eq. (10) is an upper bound of Eq. (13).

Since $R(M) = \|M\|^2$, we have $\forall M \succeq 0, \forall \lambda \geq 0, \lambda R(M) \geq 0$. Eq. (10) is then an upper bound of

$$\min_{M\succeq 0} \left[\sum_{i=1}^{m} \max_{\hat{C}_i \in \mathcal{L}_k^{n_i}} (\Delta(\hat{C}_i, C_i) + \ell_i(M; \hat{C}_i)) \right]. \tag{24}$$

Since $\operatorname{argmax}_{\hat{C}\in\mathcal{L}_k^{n_i}} \langle \hat{C}, X_i M X_i^\top \rangle = f_{M,\mathcal{L}_k^{n_i}}(X_i) \subseteq \mathcal{L}_k^{n_i}$ and $C_i \in \mathcal{L}_k^{n_i}$, we have by the definition of $f_{M,\mathcal{L}_k^{n_i}}(X_i)$:

$$\forall B \in f_{M,\mathcal{L}_k^{n_i}}(X_i), \langle B, X_i M X_i^\top \rangle \geq \langle C_i, X_i M X_i^\top \rangle \Leftrightarrow \langle B - C_i, X_i M X_i^\top \rangle \geq 0. \tag{25}$$

We then have for all i and for all $M \succeq 0$:

$$\begin{aligned}
\max_{\hat{C}_i \in \mathcal{L}_k^{n_i}} (\Delta(\hat{C}_i, C_i) + \iota_i(M; \hat{C}_i)) &= \max_{\hat{C}_i \in f_{M,\mathcal{L}_k^{n_i}}(X_i)} \Delta(\hat{C}_i, C_i) \\
&\leq \max_{\hat{C}_i \in f_{M,\mathcal{L}_k^{n_i}}(X_i)} (\Delta(\hat{C}_i, C_i) + \overbrace{\langle \hat{C}_i - C_i, X_i M X_i^\top \rangle}^{\ell_i(M;\hat{C}_i)}) \text{ see Eq. (25)} \\
&\leq \max_{\hat{C}_i \in \mathcal{L}_k^{n_i}} (\Delta(\hat{C}_i, C_i) + \ell_i(M; \hat{C}_i)) \text{ since } f_{M,\mathcal{L}_k^{n_i}}(X_i) \subseteq \mathcal{L}_k^{n_i}.
\end{aligned}$$

This completes the argument that Eq. (10) is an upper bound of Eq. (13).

A.3. Proof of Theorem 3.1

Proof. First recall that in problem (13) we use

$$\Delta(\hat{C}, C) = \|\hat{C} - C\|^2 = \|\hat{C}\|^2 - 2\langle \hat{C}, C\rangle + \|C\|^2.$$

Recall the partition matrix set $\mathcal{L}_k^n$ in (7), we can rewrite (13) equivalently as:

$$\max_{M \succeq 0, \mathrm{tr}(M)=1} \min_{\hat{C} \in \operatorname*{argmax}_{A \in \mathcal{L}_k^n} \langle A, XMX^\top \rangle} \langle \hat{C}, C\rangle, \tag{26}$$

where we have used the fact that for all $\hat{C} \in \mathcal{L}_k^n$ we have $\|\hat{C}\|^2 = \mathrm{tr}(\hat{C}^2) = \mathrm{tr}(\hat{C}) = k$, which is a constant that can be dropped (i.e. we have $\Delta(\hat{C}, C) = \|C\|^2 + k - 2\langle \hat{C}, C\rangle$).

Let $U \in \mathbb{R}^{n \times s}$ be a matrix with orthonormal columns such that $\mathsf{P}_X = XX^\dagger = UU^\top$ (we note that $s = \mathrm{rank}(X)$), and $V \in \mathbb{R}^{n \times (n-s)}$ a matrix with orthonormal columns that contains the orthogonal complement of the column space of U (to simplify the discussion, we say that V is the orthogonal complement of U).

We recall that we note $r = \min\{k, \mathrm{rank}(\mathsf{P}_X C \mathsf{P}_X)\}$.

• Suppose first $\mathrm{rank}(\mathsf{P}_X C \mathsf{P}_X) = \mathrm{rank}(U^\top C U) \geq k$ (i.e. $r = k$), then obviously $\mathrm{rank}(U) = s \geq k$. Since

$$\hat{C} \in \operatorname*{argmax}_{A \in \mathcal{L}_k^n} \langle A, XMX^\top \rangle, \tag{27}$$

the column space of $\hat{C}$ must be included in the column space of X. Indeed, we have:

$$\langle \hat{C}, XMX^\top \rangle = \langle XX^\dagger \hat{C} XX^\dagger, XMX^\top \rangle = \langle UU^\top \hat{C} UU^\top, XMX^\top \rangle \tag{28}$$

We can then write the rank-k orthogonal projection matrix $\hat{C} = HH^\top = H(H^\top H)^{-1}H^\top \in \mathcal{L}_k^n$, where $H = UQ$ and $\mathrm{rank}(H) = k$ for some matrix with orthonormal columns $Q \in \mathbb{R}^{s \times k}$ (i.e. $QQ^\top \in \mathcal{L}_k^s$). Thus, the objective value in (26) is upper bounded by:

$$\langle \hat{C}, C\rangle = \langle QQ^\top, U^\top C U\rangle \leq \max_{A \in \mathcal{L}_k^s} \langle A, U^\top C U\rangle = \sum_{i=1}^{k} \lambda_i^{\downarrow}(U^\top C U). \tag{29}$$

Now if $M \propto X^\dagger (\mathsf{P}_X C \mathsf{P}_X)_{(r)} (X^\dagger)^\top$, we have

$$XMX^\top \propto XX^\dagger (\mathsf{P}_X C \mathsf{P}_X)_{(k)} (X^\dagger)^\top X^\top = (UU^\top C UU^\top)_{(k)} = U(U^\top C U)_{(k)} U^\top, \tag{30}$$

Since $\mathrm{rank}(XMX^\top) = k$, $f_{M,\mathcal{L}}(X)$ is a singleton. By decomposing $XMX^\top \propto U(U^\top CU)_{(k)}^{1/2} (U^\top CU)_{(k)}^{1/2} U^\top$, we can write:

$$f_{M,\mathcal{L}}(X) = \{\hat{C}\},\ \hat{C} = U(U^\top CU)_{(k)}^{1/2} ((U^\top CU)_{(k)}^{1/2} U^\top U (U^\top CU)_{(k)}^{1/2})^\dagger (U^\top CU)_{(k)}^{1/2} U^\top \tag{31}$$

$$= U(U^\top CU)_{(k)}^{1/2} ((U^\top CU)_{(k)})^\dagger (U^\top CU)_{(k)}^{1/2} U^\top \tag{32}$$

We then find $\langle \hat{C}, C\rangle = \sum_{i=1}^{k} \lambda_i^{\downarrow}(U^\top CU)$, i.e. the upper bound is achieved, proving the optimality of M for this case. Even if the approximation $(\mathsf{P}_X C \mathsf{P}_X)_{(k)}$ is not unique in some cases, all the approximations written in this form return the same optimal objective value (i.e. Eq. (29)).

• If, on the other hand, $\mathrm{rank}(\mathsf{P}_X C \mathsf{P}_X) = \mathrm{rank}(U^\top CU) \leq k$ (i.e. $r = \mathrm{rank}(\mathsf{P}_X C \mathsf{P}_X)$), then we can choose $\hat{C} = HH^\top$, where $H = [UQ, VZ]$, and $Q \in \mathbb{R}^{s \times s}$ and $Z \in \mathbb{R}^{(n-s) \times (k-s)_+}$ are matrices with orthonormal columns. As already mentioned, $V \in \mathbb{R}^{n \times (n-s)}$ is the orthogonal complement of U, the choice of Z then does not depend on M (because the column space of $XMX^\top$ is included in the column space of U which is orthogonal to the column space of V, and $\hat{C}$ depends

on M only through the matrix $XMX^\top$). With this choice we see that the objective value in (26) is upper bounded by:

$$\langle \hat{C}, C \rangle = \langle QQ^\top, U^\top C U \rangle + \langle ZZ^\top, V^\top C V \rangle \tag{33}$$

$$\leq \operatorname{tr}(U^\top C U) + \sum_{j=1}^{(k-s)_+} \lambda_j^\uparrow (V^\top C V). \tag{34}$$

$\langle ZZ^\top, V^\top CV \rangle = \min_{A \in \mathcal{L}_{(k-s)_+}^{(n-s)}} \langle A, V^\top CV \rangle = \sum_{j=1}^{(k-s)_+} \lambda_j^\uparrow (V^\top CV)$ comes from the fact that we try to minimize $\langle \hat{C}, C \rangle$ in (26) and Z does not depend on M. Actually, since $\sum_{j=1}^{(k-s)_+} \lambda_j^\uparrow (V^\top CV)$ is a constant that does not depend on the learned variable M, it can be dropped from the problem along with the submatrix VZ in H (i.e. we can equivalently consider that $H = UQ$ since it is the only part that depends on the variable M that we optimize, the submatrix VZ is necessary only to satisfy the constraint $\operatorname{rank}(\hat{C}) = \operatorname{rank}(H) = k$).

By noting that $(\mathsf{P}_X C \mathsf{P}_X)_{(\operatorname{rank}(\mathsf{P}_X C \mathsf{P}_X))} = \mathsf{P}_X C \mathsf{P}_X$, a similar argument as in the previous case shows again the choice $M \propto X^\dagger (\mathsf{P}_X C \mathsf{P}_X)_{(r)} (X^\dagger)^\top = X^\dagger C (X^\dagger)^\top$ achieves the upper bound in Eq. (34) and is thus optimal. □

A.4. Proof of Theorem 3.2

Proof. If $\operatorname{rank}(C) \leq k$, then obviously $\operatorname{rank}(\mathsf{P}_X C \mathsf{P}_X) \leq \operatorname{rank}(C) \leq k$, hence $r = \operatorname{rank}(\mathsf{P}_X C \mathsf{P}_X)$. Therefore, $X^\dagger (\mathsf{P}_X C \mathsf{P}_X)_{(r)} (X^\dagger)^\top = X^\dagger \mathsf{P}_X C \mathsf{P}_X (X^\dagger)^\top = X^\dagger C (X^\dagger)^\top$. □

A.5. Similarity with linear regression

It is worth noting that Theorem 3.1 also covers cases where the solution is not equivalent to an intuitive linear regression problem, hence is more general. Moreover, we derived our solution from the large-margin structured output SVM framework by choosing a special loss ι_i. In particular, the prediction rule (see Eq. (12)) equipped with the learned metric can be used on test sets (see Eq. (18)), while it is not clear from the linear regression formulation (see Eq. (16)) how one can use the learned metric on test sets to perform clustering. Therefore, we regard the similarity to linear regression as a delightful coincidence, which sheds new light on this classical approach from the perspective of large-margin prediction.

A.6. Technical details and complexity

A.6.1 Training the metric

We explain why the complexity to compute the matrix $L = W = X^\dagger J \in \mathbb{R}^{d \times k}$ where $M = LL^\top$, $X \in \mathbb{R}^{n \times d}$ and $J \in \mathbb{R}^{n \times k}$ is $O(nd \min\{n, d\})$.

- The complexity of computing the pseudoinverse $X^\dagger \in \mathbb{R}^{d \times n}$ is $O(nd \min\{n, d\})$ (this complexity can be improved if X is low-rank or sparse).
- The calculation of $J \in \mathbb{R}^{n \times k}$ such that $JJ^\top = YY^\dagger$ can be done efficiently from a labeled assignment matrix $Y \in \{0, 1\}^{n \times k}$ with $Y\mathbf{1} = \mathbf{1}$. By noting $\mathbf{y}_c \in \{0, 1\}^n$ the c-th column of Y, the c-th column of J can be written $\mathbf{j}_c = \mathbf{y}_c / \max\{1, \|\mathbf{y}_c\|\} = \mathbf{y}_c / \max\{1, \sqrt{\mathbf{y}_c^\top \mathbf{1}}\}$. The complexity of computing J is then in $O(n)$ due to the sparsity of Y.
- $J \in \mathbb{R}^{n \times k}$ has the same number of nonzero elements as $Y \in \mathbb{R}^{n \times k}$, i.e. (at most) one per row, and thus (at most) n in total. J is then sparse. Once the matrix $X^\dagger \in \mathbb{R}^{d \times n}$ has been computed, the matrix multiplication $X^\dagger J$ would require a complexity of $O(ndk)$ in a naive implementation. Let us note $q_c = \sum_i Y_{ic}$ the number of observations in the c-th cluster, the complexity to compute the c-th column of $X^\dagger J$ is $O(dq_c)$ due to the sparsity of J. The complexity to compute all the columns of $X^\dagger J$ is then $O(\sum_{c=1}^k dq_c) = O(d \sum_{c=1}^k q_c) = O(nd)$.

The computation of $M = LL^\top = WW^\top$ is not necessary (see Section 3.4). The training complexity is then $O(nd \min\{n, d\})$ and does not depend on k.

A.6.2 Complexity of combining all the data together

The complexity of our training algorithm is $O(nd \min\{n, d\})$ where d is the space dimensionality and $n = \sum_{i=1}^m n_i$ where n_i is the number of observations in the i-th training dataset. The complexity of combining all the data together is linear in n (and thus in each n_i) if $n > d$ (and quadratic if $n < d$), i.e. linear in the size of the problem. Moreover, since d is fixed, combining more and more data increases the chances to have $n > d$.