Math 2.4 Info

As with the other three parts, you can get from 0 to 10 points. There will be homework and two opportunities to take the oral exam (last score counts).

Scoring guidelines:

- For up to 6 points, submit a complete homework (you can omit problems marked with *) and understand everything that you've submitted. This includes the ML and math that is essential to understanding the covered topics and that you should already know at this point (for example, but not limited to, learning, unsupervised learning, vectors, spaces, eigenvectors, eigendecompositions, optimization, constrained optimization, etc.).
- For 7-8 points also understand everything else that we've covered in lectures, except the most advanced concepts and proofs.
- For 9-10 points also submit problems marked with * and expect questions related to advanced concepts and proofs.

Submit on Ucilnica (pdf for the Theory and a notebook for the Implementation part, showing the code). Use LaTeX for the Theory part. Use Python and/or R for the Implementation part.

Homework is due when you take the exam for the 1st time. First submission is the final submission, that submission will be used even if you later take the exam for the 2nd time.

Essential reading

- What we covered in the lectures.
- `www.ee.cuhk.edu.hk/~wkma/engg5781/notes/lecture%204-%20PSD-%20note.pdf`
- Sections 1-4 and 7 of Ghojogh, B., Crowley, M., Karray, F., & Ghodsi, A. (2023). Multidimensional Scaling, Sammon Mapping, and Isomap. In Elements of Dimensionality Reduction and Manifold Learning (pp. 185-205). Cham: Springer International Publishing.
- Sections 1-3 of Ghojogh, B., Ghodsi, A., Karray, F., & Crowley, M. (2020). Locally linear embedding and its variants: Tutorial and survey. arXiv preprint arXiv:2011.10925.

Recommended reading

- Meilă, M., & Zhang, H. (2023). Manifold learning: what, how, and why. Annual Review of Statistics and Its Application, 11.
- `www.cs.cmu.edu/~bapoczos/Classes/ML10715_2015Fall/slides/ManifoldLearning.pdf`
- Anowar, F., Sadaoui, S., & Selim, B. (2021). Conceptual and empirical comparison of dimensionality reduction algorithms (PCA, KPCA, LDA, MDS, SVD, LLE, ISOMAP, LE, ICA, t-SNE). Computer Science Review, 40, 100378.

Homework

1 Theory

1.1 General matrix statements

Prove the following statements:

- If A is a $n \times n$ orthogonal matrix, then $A^{-1} = A^T$.
- $(AB)^T = B^T A^T$.
- $Tr(ABC) = Tr(BCA) = TR(CAB)$.

1.2 Eigendecomposition of symmetric squared real matrices

Let A be a $n \times n$ real matrix. Prove that then $A = V \Lambda V^T$, where V is an orthonormal matrix (and an orthonormal basis of eigenvectors of A) and Λ is a diagonal matrix of the corresponding eigenvalues of A.

1.3 Gram matrix

Let X be a $n \times m$ real matrix. Prove the following statements about the Gram matrix $X^T X$:

- It is symmetric.
- It is positive semi-definite.
- It has the same non-zero eigenvalues as the matrix XX^T.

1.4 Filling a gap in our Courant-Fisher theorem proof*

In our proof of this theorem we assumed that the subspace $\mathcal{V} \cap \mathcal{W}$ always contains at least one vector. Note that $\mathcal{V} = \{y = V^T x | x \in \mathcal{S}_{n-k+1}\}$ and $\mathcal{W} = \{y \in \mathbb{R}^n | y_{k+1} = ... = y_n = 0\}$

This is a direct consequence of this general proposition: Let $\mathcal{S}_1$ and $\mathcal{S}_2$ be subspaces of $\mathbb{R}^n$. If dim $\mathcal{S}_1$ + dim $\mathcal{S}_2 > n$, then dim $(\mathcal{S}_1 \cap \mathcal{S}_2) \geq 1$.

Prove this proposition.

1.5 Vector and matrix derivatives

Prove the following:

- $\frac{\partial}{\partial x} x^T a = \frac{\partial}{\partial x} a^T x = a$.
- $\frac{\partial}{\partial X} a^T X b = ab^T$.
- $\frac{\partial}{\partial X} Tr(X^2) = 2X^T$.
- (*) $\frac{\partial}{\partial X} Tr(AXBX^T C) = A^T C^T X B^T + CAXB$.

2 Implementation

Implement the methods listed below. Each method should be implemented as a function that takes the data (points or distances) and essential parameters as input, and outputs the most basic output for that method (for details, see below).

You may use libraries for basic building blocks, such as matrix inversion, optimization, eigendecomposition, nearest neighbors, and graph path finding, but it counts as advanced knowledge if you understand what goes on in the black boxes that you use.

For each method, demonstrate that it works. For this, use a 2D embedding of two toy datasets: As the starting point use a 3D swiss roll dataset and a 3D multivariate normal dataset (let each dimension have a noticeably different variance and have at least some covariance for each pair of dimensions). Then add 7 dimensions to each dataset, for a total of 10 dimensions. These extra dimensions can be just Gaussian noise that is small enough not to interfere with the *important* dimensions.

2.1 PCA

The essential parameter is the dimensionality of the embedding. Output the embedding, rotation matrix, and the proporiton of variance explained by each dimension.

2.2 ISOMAP

Essential parameters are the number of neighbors and the dimensionality of the embedding. Output the embedding.

2.3 Local Linear Embedding

Essential parameters are the number of neighbors and the dimensionality of the embedding. Output the embedding.

(*) Implement automatic selection of the number of neighbors.

The End.