Scan the QR to record your attendance

Before mid-term

- Students will NOT have internet access during the exam.
- Students MUST install Examplify on their laptops before the exam. They can refer to the slides in the file admin_week06_examplify_briefing.pdf for guidance (available in Canvas).
- Students MUST install and use the designated Python environment from the course on their laptops to prevent any library conflicts (Python environment is available at https://github.com/xbresson/CS5284_2025).
- It is the student's responsibility to ensure both Examplify and the Python environment are properly installed and functioning before the exam. Failure to do so will result in a ZERO grade.

Quick review – Graph SVM

- Semi-supervised classification → model relations between labeled & unlabeled data → graph
- Graph regularization: minimize Dirichlet energy → enforce the label consistency
  - $f^\top L f \approx \sum_{ij} A_{ij} |f(x_i) - f(x_j)|^2 \rightarrow f(x_i) = f(x_j)$

Graph SVM

Representer theorem : $f(x) = \mathrm{sign}\left(\sum_{i=1}^{n} \boxed{\xi_i^\star} K(x, x_i) + b\right) \in \pm 1$

Optimization problem : $\alpha^\star = \arg \min_{0 \leq \alpha \leq \lambda} \frac{1}{2} \alpha^T Q \alpha - \alpha^T 1_n \;\; \text{s.t.} \;\; \alpha^T \ell = 0$

with $\boxed{Q = LHK(\mathrm{I} + \gamma \mathcal{L} K)^{-1} HL} \in \mathbb{R}^{n \times n}$

Solution : $\boxed{\xi^\star = (\mathrm{I} + \gamma \mathcal{L} K)^{-1} HL \alpha^\star}$

SVM

$$f_{\mathrm{SVM}}(x) = \mathrm{sign}(\boxed{\alpha^T} LK(x) + b) \in \pm 1$$

$$\min_{0 \leq \alpha \leq \lambda} \frac{1}{2} \alpha^T Q \alpha - \alpha^T 1_n \;\; \text{s.t.} \;\; \alpha^T \ell = 0$$

with $\boxed{Q = LKL} \in \mathbb{R}^{n \times n}$

Google PageRank

- PageRank: incoming edges = popularity
  - Stochastic matrix: row-normalized as probability → $\sum_{j \in V} A_{ij} = 1, A \leftarrow D^{-1}A$
  - Irreducible matrix: fully connected graph
  - Stochastic + Irreducible: $A \leftarrow \alpha D^{-1}A + (1-\alpha)\frac{I_n}{n}$, I_n is full identity matrix
  - PageRank function: solve $u^\top A = u^\top$
    - EVD, pick the eigenvector of the largest eigenvalue ($\lambda = 1$ here), O(n^2)
    - Power method, parallelized, O(EK)
      - $u^{k=0} = \frac{I_n}{n}$
      - $u^{k+1} = \alpha D^{-1}A^\top u^k + (1-\alpha)\frac{I_n}{n}$

Do you remember we ever picked the smallest eigenvalues before?

Collaborative recommendation

- Recommendation = Matrix (user-item ratings) completion
- Low-rank assumption → "commonalities" behind rating actions

$$\min_{X \in R^{n \times m}} \boxed{rank(X)} + \frac{\lambda}{2} ||Ind_{obs} \odot (X - M)||_F^2$$

Low-rank

Accurate prediction on given ratings

min rank(X) → min ||X||$_*$

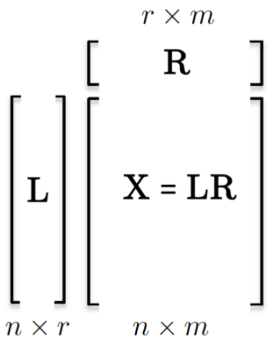
||X||$_*$=$\sum_{k=1}^{p} |\sigma_k(x)|$

Nuclear norm: sum of singular values

Convex relaxation

Primal-dual optimization

$$\min_{L \in \mathbb{R}^{n \times r}, R \in \mathbb{R}^{r \times m}} \frac{1}{2}||L||^2 + \frac{1}{2}||R||^2 + \frac{\lambda}{2}||Ind_{obs} \odot (LR - M)||_F^2$$



$r \ll n, m$

Non-convex relaxation

Non-negative matrix factorization

Content recommendation

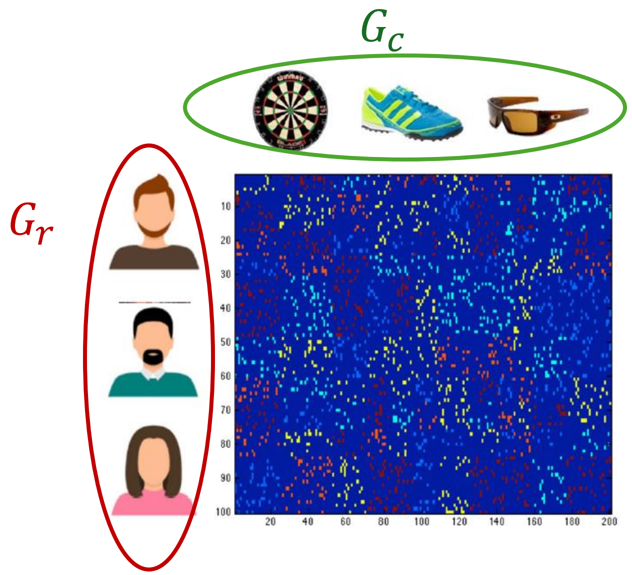
- Utilize content, user / item features, similarity
  - $\min_{X \in R^{n \times m}} ||X||_{G_r}^{diffusion} + ||X||_{G_c}^{diffusion} + \frac{\lambda}{2} ||Ind_{obs} \odot (X - M)||_F^2$
  - Dirichlet norm: $||X||_{G_r}^{diffusion} = ||X||_{G_c}^{diffusion} = tr(X^\top L X) = \sum A_{ij} |x_i - x_j|^2$
  - Solve linear system

$$(\mathrm{I}_m \otimes L_r + L_c \otimes \mathrm{I}_n + \lambda \mathrm{O}_{mn})x = \lambda m$$

Hint: $\frac{\partial\ Loss}{\partial\ x} = 0$

Kronecker product

- Hybrid system
  - collaborative + content → low-rank + similarity
  - combine their losses



Core API

- Numpy
  - np.linalg.svd(), np.linalg.inv(), np.where(D<0.5)
- Scipy
  - Scipy.sparse.kron()

Standard pre-processing

- Center data (along feature dimension) : zero-mean property

$$x_i \leftarrow x_i - \text{mean}(\{x_i\}) \in \mathbb{R}^d$$

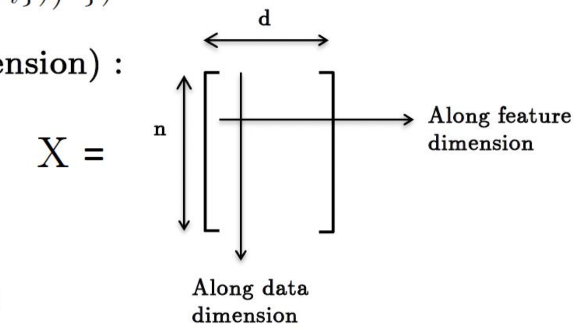
- Normalize data variance (along feature dimension) : z-scoring property

$$x_i \leftarrow x_i \;/\; \text{std}(\{x_i\}) \in \mathbb{R}^d$$

$$\text{with } \text{std}(\{x_i\})^2 = \text{mean}(\{(x_j - \text{mean}(\{x_i\}))^2\})$$

- Project data on L2-sphere (along feature dimension or data dimension) :

$$x_i \leftarrow x_i \;/\; \|x_i\|_2 \in \mathbb{R}^d$$



- Normalize max and min of feature value :

$$x_i \leftarrow \frac{x_i - \min(\{x_i\})}{\max(\{x_i\}) - \min(\{x_i\})} \in [0,1]^d$$

02_Graph_Science/code03_solution.ipynb/Question 2

Xnvar = Xzc/ np.sqrt(np.sum(Xzc**2,axis=0)+1e-10)

➔ Xnvar = Xzc / (np.sqrt(np.mean(Xzc**2, axis=0)) + 1e-10)