Machine Learning & Data Mining
Caltech CS/CNS/EE 155
Homework 5

Changhao Xu
UID: 2103530
February 17th, 2020

---

1 SVD and PCA

Question A:

$$XX^T = U\Sigma V^T (U\Sigma V^T)^T = U\Sigma V^T V \Sigma^T U^T = U\Sigma^2 U^T$$

Since $XX^T = U\Lambda U^T$, let $\Lambda = \Sigma^2$, then columns of U are the principal components of X. Eigenvalues of XX^T are squared singular values of X.

Question B: Intuitive explanation: eigenvalues of the PCA of X are the variance of X along the direction of eigenvector, and the variance are always non-negative.
Mathematical justification: from Question A, eigenvalues of the PCA of X have the property of $\Lambda = \Sigma^2$, so they must be non-negative.

Question C:

$$Tr(AB) = \sum_{i=1}^{N} (AB)_{ii} = \sum_{i=1}^{N}\sum_{j=1}^{N} A_{ij}B_{ji} = \sum_{i=1}^{N}\sum_{j=1}^{N} B_{ji}A_{ij} = \sum_{j=1}^{N} (BA)_{jj} = Tr(BA).$$

Let B = BC, then Tr(ABC) = Tr(BCA). Let A = CA, then Tr(CAB) = Tr(BCA).
In general, for any number of square matrices $A_1 \cdots A_N$, we have

$$Tr(A_1 \cdots A_N) = Tr(A_2 \cdots A_N A_1) = \cdots = Tr(A_N A_1 \cdots A_{N-1}).$$

Question D: To store a truncated SVD with $U\Sigma V^T$, for U we need N*k values, for Σ we need k values since it's a diagonal matrix and all other coefficients are 0, for V we need N*k values. Therefore, in total we need (2N+1)k values. When $(2N+1)k < N^2$, that is $k < \frac{N^2}{2N+1}$, storing the truncated SVD is more efficient than storing the whole matrix.

Question E: Since Σ only has non-zero values on entries Σ_{ii}, where $i \in \{1, \cdots N\}$, when multiply

$$(U\Sigma)_{ij} = \sum_{k=1}^{D} U_{ik}\Sigma_{kj} = \sum_{k=1}^{N} U_{ik}\Sigma_{kj} + \sum_{k=N+1}^{D} U_{ik}\Sigma_{kj} = \sum_{k=1}^{N} U_{ik}\Sigma_{kj} + \sum_{k=N+1}^{D} U_{ik}0 = \sum_{k=1}^{N} U_{ik}\Sigma_{kj}$$

where $i \in \{1, \cdots D\}, j \in \{1, \cdots N\}$. Therefore, $U\Sigma = U'\Sigma'$, where U' is the D $\times$ N matrix consisting of the first N columns of U, and Σ' is the N $\times$ N matrix consisting of the first N rows of Σ.

Question F: U' is a D $\times$ N matrix, and U'^T is a N $\times$ D matrix. Therefore, $U'U'^T$ is a D $\times$ D matrix and $U'^T U'$ is a N $\times$ N matrix. Since they are not equal, U' is not orthogonal.

Question G: Since columns of U' are still orthonormal,

$$(U'^T U')_{ij} = \sum_{k=1}^{D} U'^T_{ik} U'_{kj} = \sum_{k=1}^{D} U'_{ki} U'_{kj} = 1$$

---

if and only if i=j, where $i \in \{1, \cdots N\}, j \in \{1, \cdots N\}$. So $(U'^T U') = I_{N \times N}$.
On the other hand, $(U'U'^T) \neq I_{D \times D}$. This is because rows of U′ cannot be orthonormal since rank(U′) $\leq$ N while D >N.

Question H: On lecture 10 slide 53, $X^+ = V\Sigma^+ U^T$, where Σ^+ is a diagonal matrix.

$$\Sigma^+ = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_D \end{bmatrix} \qquad \sigma^+ = \begin{cases} 1/\sigma & \text{if } \sigma > 0 \\ 0 & \text{otherwise} \end{cases}$$

When Σ^+ is also invertible, that is $\sigma_i \neq 0$, $\Sigma^{-1} = \Sigma^+$ where all diagonal units are $\frac{1}{\sigma_i}$. So $X^+ = V\Sigma^{-1}U^T$.

Question I: $X^{+\prime} = (X^T X)^{-1} X^T \Leftrightarrow X^T X X^{+\prime} = X^T$.
On the other hand, since $XX^+ = I$, $X^T X X^+ = X^T$.
Therefore, $X^T X X^{+\prime} = X^T X X^+$. That is $X^{+\prime} = X^+$.

Question J: The least squares solution of pseudoinverse $X^{+\prime} = (X^T X)^{-1} X^T$ is prone to numerical errors. From Question A, $X^T X = V\Sigma^2 V^T$, eigenvalues of $X^T X$ are squared singular values of X. Compared with $X^+ = V\Sigma^+ U^T$, condition number $\kappa(X^T X)$ is higher than that of $\kappa(\Sigma)$.

2 Matrix Factorization

Question A:

$$\partial_{u_i} = \lambda u_i - \sum_{j=1}^{N} v_j (y_{ij} - u_i^T v_j)$$

$$\partial_{v_j} = \lambda v_j - \sum_{i=1}^{N} u_i (y_{ij} - u_i^T v_j)$$

Question B: Let $\partial_{u_i} = 0$, and $\partial_{v_j} = 0$.
That is,

$$\begin{cases} \lambda u_i - \sum_{j=1}^{N} v_j (y_{ij} - u_i^T v_j) = 0 \\ \lambda v_j - \sum_{i=1}^{N} u_i (y_{ij} - u_i^T v_j) = 0 \end{cases}$$

From 1st equation,

$$\lambda u_i = \sum_{j=1}^{N} v_j y_{ij} - \sum_{j=1}^{N} v_j u_i^T v_j = \sum_{j=1}^{N} v_j y_{ij} - \sum_{j=1}^{N} v_j v_j^T u_i$$

$$u_i = (\lambda I + \sum_{j=1}^{N} v_j v_j^T)^{-1} \sum_{j=1}^{N} v_j y_{ij}$$

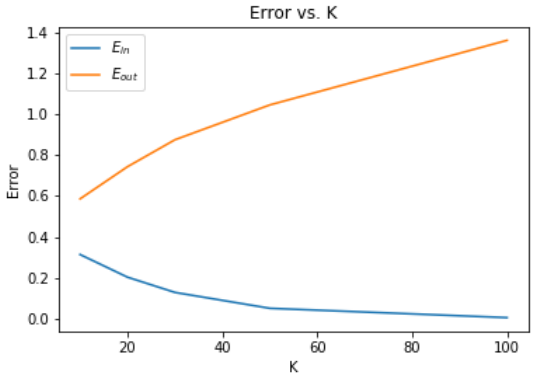
Changhao Xu
UID: 2103530

Similarly, from 2nd equation,

$$v_j = (\lambda I + \sum_{i=1}^{N} u_i u_i^T)^{-1} \sum_{i=1}^{N} u_i y_{ij}$$

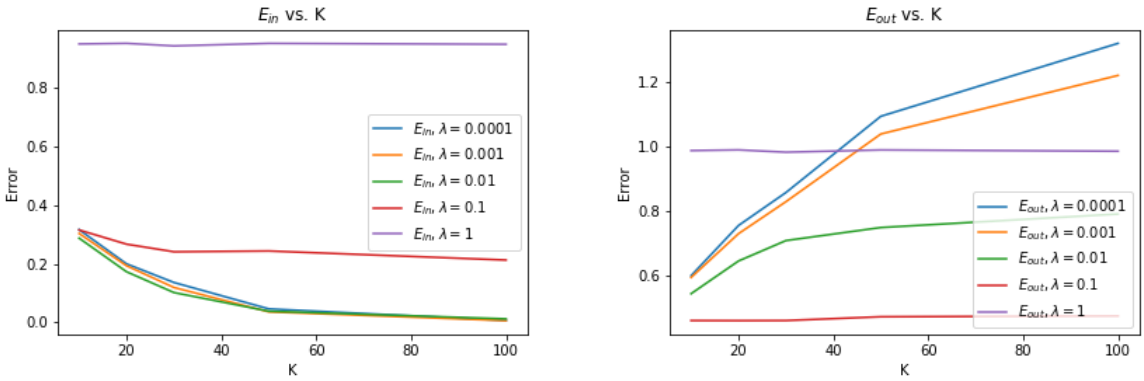
Question C: See jupyter notebook.

Question D:



As k increases, E_{in} decreases while E_{out} increases. The increase of k means more latent factors, and as more parameters our model have, overfitting occurs which has a low training error but a high out-of-sample error.

Question E:



As k increases for small regularization λ, E_{in} decreases while E_{out} increases. The increase of k means more latent factors, and as more parameters our model have while regularization term λ is small, overfitting occurs which has a low training error but a high out-of-sample error.
As λ increases, E_{in} increases while E_{out} first decrease then increase due to the penalty on overfitting. When

λ = 0.1, testing error is the lowest while training error is low as well, indicating good performance with different k. When λ is too large and reachs 1, underfitting occurs which has a high training and testing error.

3 Word2Vec Principles

Question A: $\log p(w_O|w_I) = {v'}_{w_O}^T v_{w_I} - \log \sum_{w=1}^{W} \exp({v'_w}^T v_{w_I})$.
Therefore,

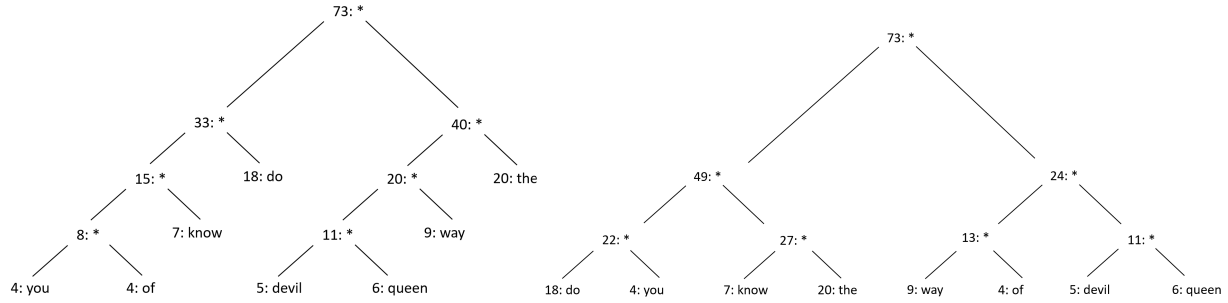
$$\nabla_{{v'}_{w_O}} \log p(w_O|w_I) = v_{w_I} - v_{w_I} \times \frac{\exp({v'}_{w_O}^T v_{w_I})}{\sum_{w=1}^{W} \exp({v'_w}^T v_{w_I})}$$

, and

$$\nabla_{v_{w_I}} \log p(w_O|w_I) = {v'}_{w_O} - \frac{\sum_{w=1}^{W} v'_w \exp\left({v'_w}^T v_{w_I}\right)}{\sum_{w=1}^{W} \exp\left({v'_w}^T v_{w_I}\right)}$$

.
That is, computing these gradients scale with O(W). Similarly, these gradients scale with O(D). So time complexity is O(WD).

Question B:



Expected representation length of Huffman tree = $\frac{(4+4+5+6)\times 4+(7+9)\times 3+(18+20)\times 2}{73} = 2.73973$, while expected representation length of balanced binary tree is 3.

Question C: The training objective will increase as D increases. A larger D means more features in the embedding space, which will lead to overfitting for very large D.

Question D: See Jupyter notebook.

Question E: (308, 10)

Question F: (10, 308)

Question G:
Pair(them, would), Similarity: 0.9628089

Pair(would, them), Similarity: 0.9628089
Pair(car, them), Similarity: 0.95937026
Pair(like, or), Similarity: 0.957788
Pair(or, like), Similarity: 0.957788
Pair(not, them), Similarity: 0.95743936
Pair(eat, would), Similarity: 0.9547398
Pair(a, eat), Similarity: 0.95205164
Pair(in, not), Similarity: 0.9470372
Pair(i, or), Similarity: 0.9458327
Pair(ned, dear), Similarity: 0.9441935
Pair(dear, ned), Similarity: 0.9441935
Pair(do, a), Similarity: 0.9412332
Pair(eleven, boat), Similarity: 0.93666583
Pair(boat, eleven), Similarity: 0.93666583
Pair(red, oh), Similarity: 0.93665606
Pair(oh, red), Similarity: 0.93665606
Pair(could, in), Similarity: 0.9363972
Pair(things, sing), Similarity: 0.9356929
Pair(sing, things), Similarity: 0.9356929
Pair(open, cans), Similarity: 0.9347308
Pair(cans, open), Similarity: 0.9347308
Pair(and, i), Similarity: 0.9308523
Pair(samiam, car), Similarity: 0.92888826
Pair(with, box), Similarity: 0.9280398
Pair(box, with), Similarity: 0.9280398
Pair(from, red), Similarity: 0.9266325
Pair(low, goodbye), Similarity: 0.92613554
Pair(goodbye, low), Similarity: 0.92613554
Pair(here, samiam), Similarity: 0.92585975

Question H: Many pairs appear at the same time, such as (them, would), (would, them). This is because they has the same similarity. Also words are more similar when they often appear closely in sentences, such as (dear, ned), (open, cans), (and, i).