1 SVD and PCA [35 Points]

Problem A [3 points]:

Solution A: The principal components of X are the vectors $u_1, u_2, \ldots, u_n$ such that $XX^T = U\Lambda U^T$, where $U = [u_1\, u_2\, \cdots\, u_n]$ and Λ is a diagonal matrix with the eigenvalues of XX^T along the diagonal. Given the singular value decomposition (SVD) decomposition $X = U\Sigma V^T$, $XX^T = (U\Sigma V^T)(U\Sigma V^T)^T = U\Sigma V^T V \Sigma^T U^T$. Since V is orthogonal, $U\Sigma V^T V\Sigma^T U^T = U\Sigma\Sigma^T U^T = U\Sigma^2 U^T = U\Lambda U^T$. Thus, the columns of U are the principal components of X, where $\Lambda = \Sigma^2$, and thus the singular values of X are the square roots of the eigenvalues of XX^T.

Problem B [4 points]:

Solution B: Intuitive explanation: Since the feature covariance matrix Σ is expressed as $\Sigma = XX^T = U\Lambda U^T$, the diagonal terms of Σ, Σ_{dd}, correspond to the covariances of feature d with itself in the training data, which are non-negative for all features d. Therefore, all the diagonal terms of Λ, which are the eigenvalues of the PCA of X, are non-negative.

Mathematical explanation: Since the eigenvalues of the PCA of X are the squares of the singular values of the SVD of X, the eigenvalues of the PCA of X must be non-negative.

Problem C [5 points]:

Solution C: Using the definition of matrix multiplication, $C = AB$ where $c_{ij} = \sum_{k=1}^{m} a_{ik}b_{kj}$ where m is the number of columns of A and the number of rows of B, $\text{Tr}(AB) = \sum_{i=1}^{N}(AB)_{ii} = \sum_{i=1}^{N}(\sum_{j=1}^{N} a_{ij}b_{ji}) = \sum_{i=1}^{N}(\sum_{j=1}^{N} b_{ji}a_{ij}) = \sum_{j=1}^{N}(\sum_{i=1}^{N} b_{ji}a_{ij}) = \sum_{i=1}^{N}(BA)_{ii} = \text{Tr}(BA)$.

Generalizing to square matrices A, B, and C,

$$\text{Tr}(ABC) = \sum_{i=1}^{N}(ABC)_{ii} = \sum_{i=1}^{N}\sum_{k=1}^{N}(AB)_{ik}C_{ki} = \sum_{i=1}^{N}(\sum_{k=1}^{N}(\sum_{j=1}^{N} a_{ij}b_{jk})c_{ki}) = \sum_{i=1}^{N}\sum_{k=1}^{N}\sum_{j=1}^{N} a_{ij}b_{jk}c_{ki}$$

$$= \sum_{i=1}^{N}\sum_{k=1}^{N}\sum_{j=1}^{N} a_{ij}(b_{jk}c_{ki}) = \sum_{i=1}^{N}\sum_{k=1}^{N}\sum_{j=1}^{N}(b_{jk}c_{ki})a_{ij} = \sum_{j=1}^{N}(\sum_{i=1}^{N}(\sum_{k=1}^{N} b_{jk}c_{ki})a_{ij}) = \sum_{j=1}^{N}\sum_{i=1}^{N}(BC)_{ji}A_{ij}$$

$$= \sum_{j=1}^{N}(BCA)_{jj} = \text{Tr}(BCA).$$

Furthermore,

$$\text{Tr}(BCA) = \sum_{i=1}^{N}(BCA)_{ii} = \sum_{i=1}^{N}\sum_{k=1}^{N}(BC)_{ik}A_{ki} = \sum_{i=1}^{N}(\sum_{k=1}^{N}(\sum_{j=1}^{N} b_{ij}c_{jk})a_{ki}) = \sum_{i=1}^{N}\sum_{k=1}^{N}\sum_{j=1}^{N} b_{ij}c_{jk}a_{ki}$$

$$= \sum_{i=1}^{N}\sum_{k=1}^{N}\sum_{j=1}^{N} b_{ij}(c_{jk}a_{ki}) = \sum_{i=1}^{N}\sum_{k=1}^{N}\sum_{j=1}^{N}(c_{jk}a_{ki})b_{ij} = \sum_{j=1}^{N}(\sum_{i=1}^{N}(\sum_{k=1}^{N} c_{jk}a_{ki})b_{ij}) = \sum_{j=1}^{N}\sum_{i=1}^{N}(CA)_{ji}B_{ij}$$

$$= \sum_{j=1}^{N}(CAB)_{jj} = \text{Tr}(CAB).$$

Therefore, $\text{Tr}(ABC) = \text{Tr}(BCA) = \text{Tr}(CAB)$ holds for any square matrices A, B, and C.

Problem D [3 points]:

Solution D: To store a truncated SVD with k singular values, the first k columns of U, the first k singular values in Σ are needed, and the first k rows of V^T are needed. Therefore, To store a truncated SVD with k singular values of an $N \times N$ matrix X, $Nk + k + Nk = (2N + 1)k$ values are needed.

Problem E [3 points]: .

Solution E: Since X has rank $N < D$, the values along the diagonal in Σ are 0. Therefore, the only nonzero entries of Σ are Σ_{ii} where $i \leq N$. Using the definition of matrix multiplication (see problem 1C),

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

where U' is the $D \times N$ matrix consisting of the first N columns of U, and where Σ' is the $N \times N$ matrix consisting of the first N rows of Σ. Therefore, $U\Sigma = U'\Sigma'$.

Problem F [3 points]:

Solution F: Since U' is not square, $U'U'^T$ has different dimensions from U'^TU', so $U'U'^T \neq U'^TU'$. Since a matrix A is orthogonal if $AA^T = A^TA = I$, because U' is not square, U' is not orthogonal.

Problem G [4 points]:

Solution G: The ij-th entry of $U'^T U'$ is the dot product of vectors u_i and u_j, where u_i^T is the ith row of U'^T, and u_j is the jth column of U'. If $i = j$, since U' has orthonormal columns, then $u_i \cdot u_j = u_i \cdot u_i = 1$, so every ii-th entry of $U'^T U'$ is 1. If $i \neq j$, $u_i \cdot u_j = 0$, so so every ij-th entry of $U'^T U'$ where $i \neq j$ is 0. Since $U'^T U'$ has dimensions $N \times N$, $U'^T U' = I_{N \times N}$.

Assume $U' U'^T = I_{D \times D}$. Thus, the ij-th entry of $U' U'^T$ must be 1 if $i = j$ or 0 if $i \neq j$. This means that, similar to above, U' must have orthonormal rows. However, since U' is only guaranteed to have orthonormal columns, a contradiction is reached, so $U' U'^T = I_{D \times D}$ does not hold for any U' as given. Therefore, it is not true that $U' U'^T = I_{D \times D}$ for U' as given.

Problem H [4 points]:

Solution H:

Problem I [4 points]:

Solution I: Using the SVD of X, $X = U\Sigma V^T$ and assuming that Σ is invertible,

$$X^{+'} = (X^TX)^{-1}X^T = ((U\Sigma V^T)^T U\Sigma V^T)^{-1}(U\Sigma V^T)^T = (V\Sigma^T U^T U\Sigma V^T)^{-1}V\Sigma^T U^T$$

$$= (V\Sigma^2 V^T)^{-1}V\Sigma^T U^T = V^{T^{-1}}\Sigma^{2^{-1}}V^{-1}V\Sigma^T U^T = V^{T^{-1}}\Sigma^{2^{-1}}\Sigma^T U^T = V^{T^{-1}}\Sigma^{2^{-1}}\Sigma^T U^T.$$

Since V is orthogonal, $V^T = V^{-1}$, $V^{T^{-1}} = V^{-1^{-1}} = V$. Since Σ is diagonal and invertible, $\Sigma^T = \Sigma$ and $\Sigma^{2^{-1}} = \Sigma^{-1^2}$. Therefore,

$$V^{T^{-1}}\Sigma^{2^{-1}}\Sigma^T U^T = V\Sigma^{2^{-1}}\Sigma U^T = V\Sigma^{-1^2}\Sigma U^T = V\Sigma^{-1}\Sigma^{-1}\Sigma U^T = V\Sigma^{-1}U^T.$$

Thus, assuming that Σ is invertible, $X^{+'} = (X^TX)^{-1}X^T$ is equivalent to $V\Sigma^{-1}U^T$.

Problem J [2 points]:

Solution J: The least squares psuedoinverse is more prone to numerical error than the pseudoinverse of Problem H, since the condition number of $X^T X$ is greater than that of Σ, since Σ is diagonal.

Philip Carr
February 2019

2 Matrix Factorization [30 Points]

Problem A [5 points]:

Solution A:

2A. $L = \frac{\lambda}{2}\left(\|U\|_F^2 + \|V\|_F^2\right) + \frac{1}{2}\sum_{i,j}\left(y_{ij} - u_i^T v_j\right)^2$

$$\partial_{u_i} L = \frac{\lambda}{2}\partial_{u_i}\left(\|U\|_F^2 + \|V\|_F^2\right) + \frac{1}{2}\partial_{u_i}\left(\sum_{i,j}\left(y_{ij} - u_i^T v_j\right)^2\right)$$

$$= \frac{\lambda}{2}\partial_{u_i}\left(\|u_i\|_2^2\right) + \frac{1}{2}\partial_{u_i}\left(\sum_j\left(y_{ij} - u_i^T v_j\right)^2\right)$$

$$= \frac{\lambda}{2}\cdot 2u_i + \frac{1}{2}\cdot 2\sum_j\left((-v_j)\left(y_{ij} - u_i^T v_j\right)^T\right) = \lambda u_i - \sum_j\left(v_j\left(y_{ij} - u_i^T v_j\right)^T\right).$$

$$\partial_{v_j} L = \frac{\lambda}{2}\partial_{v_j}\left(\|U\|_F^2 + \|V\|_F^2\right) + \frac{1}{2}\partial_{v_j}\left(\sum_{i,j}\left(y_{ij} - u_i^T v_j\right)^2\right) = \frac{\lambda}{2}\partial_{v_j}\left(\|v_j\|_2^2\right)$$

$$+ \frac{1}{2}\partial_{v_j}\left(\sum_j\left(y_{ij} - u_i^T v_j\right)^2\right) = \lambda v_j - \sum_i\left(u_i\left(y_{ij} - u_i^T v_j\right)\right).$$

Problem B [5 points]:

Solution B:

For u_i,

2B. $\lambda \partial_{u_i} L = 0 \Rightarrow \lambda u_i - \sum_j (v_j (y_{ij} - u_i^T v_j)^T) = 0$

$\Rightarrow \lambda u_i - \sum_j (v_j (y_{ij} - v_j^T u_i)) = 0 \Rightarrow \lambda u_i + \sum_j v_j v_j^T u_i = \sum_j v_j y_{ij}$

$(I\lambda + \sum_j v_j v_j^T) u_i = \sum_j v_j y_{ij} \Rightarrow u_i = (I\lambda + \sum_j v_j v_j^T)^{-1} \sum_j v_j y_{ij}$.

For v_j,

$\partial_{v_j} L = 0 \Rightarrow \lambda v_j + \sum_i (u_i (y_{ij} - u_i^T v_j)) = 0$

$\Rightarrow \lambda v_j + \sum_i (u_i u_i^T v_j) = \sum_i u_i y_{ij} \Rightarrow (I\lambda + \sum_i (u_i u_i^T)) v_j = \sum_i u_i y_{ij}$

$\Rightarrow v_j = (I\lambda + \sum_i (u_i u_i^T))^{-1} \sum_i u_i y_{ij}$.

Problem C [10 points]:

Solution C: *See 2D.py and prob2utils.py for the solution code.*

Problem D [5 points]:

Solution D:

Problem E [5 points]:

Solution E:

3 Word2Vec Principles [35 Points]

Problem A [5 points]:

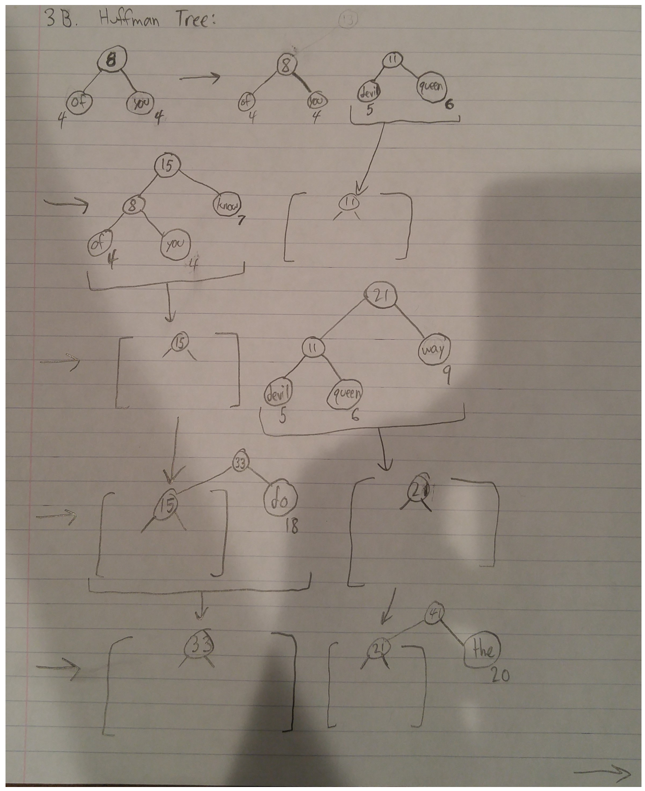
Solution A: Computing these gradients scales as $O(W^3)$, since gradients must be computed for every possible pair of w vectors (this is $O(W^2)$), and since gradients must be computed for every word in the word list (this is $O(W)$).

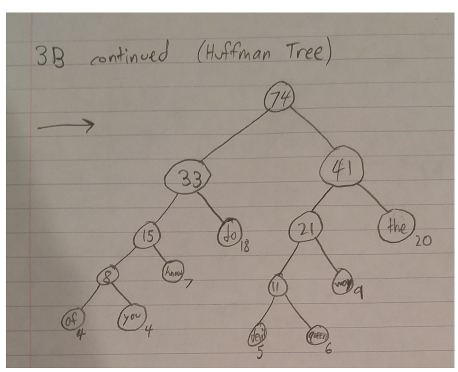
Since increasing D increases the length of each w vector, computing these gradients scales as $O(D)$, since if d more dimensions are added to each w vector, there are d more pairs terms to multiply together with each gradient.

In total, computing these gradients scales with $O(W^3D)$.

Problem B [10 points]:

Solution B:





The complete Huffman Tree is above.

Problem C [3 points]:

Solution C:

Problem D [10 points]:

Solution D: *See solution code in P3C.py*

Problem E [2 points]:

Solution E:

Problem F [2 points]:

Solution F:

Problem G [1 points]:

Solution G:

Problem H [2 points]:

Solution H: