Matrix factorizations and low rank approximation

Friday, March 11, 2016:

1. EIGENFACES

In this section, we learn intuition and math behind *Eigenfaces*, a technique to apply matrix **SVD** factorization to identify people's faces from the database.

1.1. Image manipulation.

Image manipulation. Let's assume we have n images of people's faces that are:

- Grey scale
- Same size ($m_1 \times m_2$)
- Same position/orientation of the face

For the method to work, let's reshape the matrix into one column vector that contains all of the image's information:

$$\underset{m_1 \times m_2}{\textsf{Face\_Image}} \rightarrow \underset{m_1 * m_2 \times 1}{\mathbf{t_1}}$$

After transforming each image into a vector, we can align them together to get the matrix:

$$\mathbf{T} = [\mathbf{t_1}, \mathbf{t_2}, \cdots, \mathbf{t_n}].$$

Usually, $m >> n$.

1.2. EVD - eigenvalue decomposition.

EVD - eigenvalue decomposition. In this method, we seek to compress T such that:

$$\mathbf{S} = \mathbf{T}\mathbf{T}^*$$

$$\mathbf{S} = \mathbf{U}\mathbf{D}\mathbf{U}^*$$

Then, let's pick a tolerance measure to pick k eigenvalues that reconstruct matrix $\mathbf{S}$ sufficiently accurately:

$$\frac{\lambda_1 + \lambda_2 + \cdots + \lambda_k}{\lambda_1 + \lambda_2 + \cdots + \lambda_n} < 1 - tolerance$$

Therefore, we have a matrix:

$$\underset{m \times m}{\mathbf{S}} \approx \underset{m \times k}{\mathbf{U}} \; \underset{k \times k}{\mathbf{D}} \; \underset{k \times m}{\mathbf{U}^*}.$$

Where, vectors u_i for $i \in [1 \ldots k]$, are the "eigenfaces". They form an approximate basis for the columns of T. Therefore, our reconstruction of original matrix $\mathbf{T}$ is:

$$\mathbf{T} \approx \mathbf{U}\mathbf{U}^*\mathbf{T} = \mathbf{U}\hat{\mathbf{T}}$$

where,

$$\hat{\mathbf{T}} = \mathbf{U}^*\mathbf{T}$$

Useful applications.

(1) **Storage efficiency**: store only matrices $\mathbf{U}$ and $\hat{\mathbf{T}}$ instead of $\mathbf{T}$.
(2) **Face recognition**: given a new face image encoded in a vector - $\mathbf{S}$, we can attempt to find an image $\mathbf{t_j}$ in our database that's closest to $\mathbf{S}$.
Our job is to find $i = argmin||\mathbf{t_p} - \mathbf{S}||, i \le p \le n$ Let:

$$\hat{\mathbf{S}} = \mathbf{U}^*\mathbf{S}$$

Check that $||\mathbf{S} - \mathbf{U}\mathbf{U}^*\mathbf{S}|| = ||\mathbf{S} - \mathbf{U}\hat{\mathbf{S}}||$ is small. If it's not, then the given image doesn't have a match in the database, so we can add it by updating $\mathbf{U}$ and $\hat{\mathbf{T}}$.
Caveat: L_2 distance is not a good measure of closeness between images. A lot of that difference could just be noise, difference in light, shades etc.

1.3. **Problem with Eigenfaces.** $\mathbf{S} = \mathbf{T}\mathbf{T}^*$ is very large.

Typically $n << m$.

Solution 1: For $\mathbf{S} = \mathbf{T}^*\mathbf{T}$, let's computer it's EVD.

(1) Suppose $\mathbf{T}^*\mathbf{T}\mathbf{v} = \lambda\mathbf{v} \to \mathbf{T}\mathbf{T}^*\mathbf{T}\mathbf{v} = \lambda\mathbf{T}\mathbf{v}$

(2) If we set $\mathbf{u} = \mathbf{T}\mathbf{v}$, we have a familiar system $\mathbf{S}\mathbf{u} = \lambda\mathbf{u}$

(3) Let $\mathbf{v_j}$ for $j \in [1 \cdots n]$ be eigenvectors of $\mathbf{T}^*\mathbf{t}$, normalized so that $||\mathbf{v_j}|| = 1$.

(4) Set $\mathbf{u_j} = \mathbf{T}\mathbf{v_j}$, then $\mathbf{u_i} \cdot \mathbf{u_j} = \mathbf{u_i}^*\mathbf{u_j} = \mathbf{v_i}^*\mathbf{T}^*\mathbf{T}\mathbf{v_j} = \lambda_j \mathbf{v_i}^*\mathbf{v_j} = \begin{cases} \lambda_j, i = j \\ 0, i \neq j \end{cases}$

Solution 2: Compute SVD of **T**

(1) Suppose rank of $\mathbf{T}$ is k. We know that $k \le min(m, n)$.
(2) $\mathbf{T} = \mathbf{U}\sum\mathbf{V}^*$
$\mathbf{T}\mathbf{T}^* = \mathbf{U}\sum^2\mathbf{U}^*$
$\mathbf{T}^*\mathbf{T} = \mathbf{V}\sum^2\mathbf{V}^*$
(3) The left singular vectors of $\mathbf{T}$ are the eigenfaces.
Observe that $\mathbf{u_j} = \mathbf{T}\mathbf{v_j} = (\sum_{i=1}^k \sigma_i \mathbf{u_i}\mathbf{v_i}^*)\mathbf{v_j} = \sigma_j\mathbf{u_j}$
(4) Rescale to get the left svecs of $\mathbf{T}$. In practice, we don't explicitly form $\mathbf{T}^*\mathbf{T}$. Instead, $\mathbf{T}^*\mathbf{T} = \sum_{i=1}^n \sigma_i^2 \mathbf{u_i}\mathbf{v_i}^*$