How to use PCA when the feature dimension is so huge then svd or eig will be so slow and out of memory?

Rasool Fakoor *

Problem Definition

Assume:
m: #data points(training examples).
n: #features.
A: training examples set which is A_{n*m}.
In PCA algorithm, to calculate the $\sum$:

$$\sum = AA^T \tag{1}$$

Problem: The $\sum$ will be $n * n$. If $n >> 64k$ then $\sum$ is **HUGE**.

Clever solution when ($m << 64K$)

Instead of calculating $\sum = AA^T$, lets calculate $L = A^T A$. The size of of L would be $m * m$. As a result, now svd or eig deals with much smaller matrix to calculate the eigenvalues and eigenvectors. However, the question here is what is the relationship between L's eigenvalues and eigenvectors and $\sum$'s eigenvalues and eigenvectors?

Proof

If v is eigenvetor of L then Av is eigenvector of $\sum$:

$$Lv = \gamma v$$
$$A^T A = \gamma v$$
$$A(A^T A) = A(\gamma v) = \gamma Av$$
$$(AA^T)Av = \gamma(Av)$$
$$\sum(Av) = \gamma(Av)$$

Summary

When face with above problem, follow the following steps:

1. $L = A^T A$
2. $[v, u, s] = svd(L)$
3. $v' = Av$

Reference

- `http://ranger.uta.edu/~heng/CSE6363.html`

*`https://sites.google.com/site/rfakoor/`