고급소프트웨어실습 8 주차 과제
PCA 에서 압축을 위한 최적 eigenvector 개수 구하기

20140938 임다은

문제 1

Eigenvector 의 개수는 원래 데이터의 차원(dimension)과 같으므로 모든 eigenvector 를 이용 하여 원래 데이터를 투영하면 데이터 압축의 효과를 볼 수 없다. 또한 데이터의 분산 특성 을 고려하지 않고 특정 몇 개의 eigenvector 를 이용하여 투영시키면 데이터의 유용한 정보 가 손실될 수 있다. 유용한 정보를 손실하지 않으면서 최대로 압축 효과를 얻을 수 있도록 eigenvector 의 개수를 설정할 수 있는 방법이 있는 지 설명해 보자.

Covariance Matrix 에서 Singular Value Deecomposition 을 하게 되면, Eigenvalues 와 그에 해당하는 Eigen vectors 를 얻어낼 수 있다. PCA 방법을 활용하면 n X n 행렬이 있다고 할 때 n 개의 Eigen vector 가 나오게 된다. SVD 이후에는 eigenvalue 들을 내림차순으로 정렬하고, eigenvector 를 이에 따라 정렬하면 분산의 크기에 따라 정렬된 eigenvector 가 나온다.

이 때 PCA 를 통해서 특정한 패턴을 찾을 수 있다면, 정보의 많은 손실없이 데이터를 압축할 수 있게된다. 특정 몇개의 eigenvector 를 구할 때 가장 정보의 손실을 줄이는 방법은 가장 도드라지는 성분, 즉 principal components 들을 이용하여 투영하는 것이다.

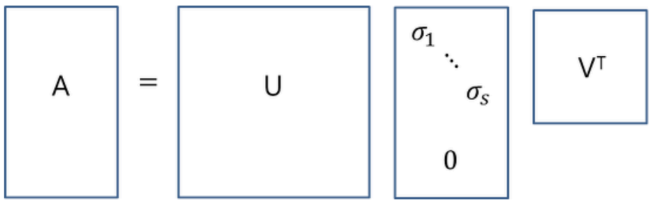
먼저 샘플들을 모아서 PCA 분석을 진행하면 샘플들은 각각 특정 차원(N)의 벡터공간에서 eigenvector 가 군집을 형성하게 될 것이다. 이 군집의 내에서 평균을 구한다면 이를 특징벡터로 활용할 수 있게 된다. 또는 eigenvalue 가 충분히 큰 n<N 개의 basis 를 이용하여 원본의 데이터를 표현하면 그보다 작은 것들을 노이즈로 처리하고, 원본과 비슷한 이미지를 리턴하여 줄 것이다. 이 때 eigenvalue 가 충분히 큰 n 개의 개수는 실험을 통하여 적절한 값을 도출하여야 한다. 이 때 실험은 가장 분산이 큰 eigenvalue 를 가진 vector 들부터 차례로 대입하면서 원본 이미지와의 오차를 구하며, 오차가 줄어드는 값이 현저히 줄어드는 구간으로 정해야 할 것이다.

또한 Normalize 된 eigenvector 를 구할 때 X 개의 샘플이 있다면 X*X 차원정도의 대각화를 이루도록 covariance matrix 를 구하면 된다. 모든 계산을 하는 것은 불필요하다.

투영하기 전에 SVD 단계에서 계산을 줄이는 것도 가능하다. 모든 Eigenvector 를 저장하면 압축효과를 전혀 볼 수가 없기에 적당한 개수를 정해서 저장해야하는데, 여기서 Eigen vector 의 개수를 설정하는 것이 또 다른 문제가 될 수 있다. 이는 Eigen vector 의 개수를

변화시켜 보면서 그 때 Eigen vectors 로부터 재구성한 이미지들이 얼마나 원본과 유사한지를 나타내는 측도를 분석하면서 (예를 들어, Mean Square Error 분석) 적당한 Eigenvectors 의 개수를 구하면, 데이터의 압축은 최대로 하면서 정보의 압축은 최대로 하는 효과를 얻을 수 있다. 예를 들면 가장 분산이 큰 Eigen vector 부터 시작해서 얼굴 인식정도를 그래프에 표현하면 Eigen vector 의 개수가 늘어날수록 더욱 더 정확하게 얼굴을 인식한다는 것을 볼 수 있다. 하지만 어느 순간 그 증가하는 정도가 급격히 줄어들고 더 많은 Eigen vector 를 추가해도 별 도움이 안 되는 순간을 포착할 수 있다. 이때를 찾아 그 뒤에 있는 Eigen vector 는 무시하고 그 앞에 있는 것들로만 추려내면 압출 효과를 내면서도 동시에 얼굴인식의 기능을 충분히 유지할 수 있다.

문제에서 설명한 데이터차원의 개수만큼 모든 eigenvector 를 이용하여 데이터를 투영하는 것을 Singular value decomposition (SVD)에서 full SVD 라고 부른다. 그러나 위의 설명과 같이 계산의 양이 너무 많아 통상적으로는 계산을 줄이기 위해 reduced SVD 로 구현한다.



<그림 1 full SVD>

이 때 reduced SVD 를 구현하는 방법은 총 세가지가 있다. 첫째로, M*N 직사각형 대각행렬인 Covariance matrix (Σ)에서 대각파트가 아닌 0 들을 없애고, U 에서는 0 들과 대응되는 열벡터들을 제거한 형태를 thin SVD 라고 한다. 둘째로, thin SVD 에서 0 인 singular value 들을 추가적으로 제거하여 SVD 를 하는 것을 compact SVD 라고 한다. 그리고 마지막으로 0 이 아닌 singular value 까지 제거한 것을 truncated SVD 라고 한다. thin SVD 와 compact SVD 는 같은 A 값을 가지는 반면 truncated SVD 의 결과는 A 에 근사한 행렬값을 가지게 된다.

출처 [1]
https://m.blog.naver.com/PostView.nhn?blogId=helloktk&logNo=80036492405&proxyReferer=https:%2F%2Fwww.google.co.kr%2F
출처 [2] http://darkpgmr.tistory.com/106 [다크 프로그래머]