spectral

kl583

August 2016

1 Introduction

1.1 What is Spectral?

lmao if only I knew

If you type in Spectral Theorem in Google, you'll get a Berkeley EECS website that says something like the following theorem :

Theorem 1 (Spectral Theorem) *Let $A \in S^n$ be a symmetric matrix then we can decompose A as:*

$$A = \sum_{i=1}^{n} \lambda_i u_i u_i^t = U \Lambda U^t$$

Where $\Lambda = diag(\lambda_1, ..\lambda_n)$ and U is an orthogonal matrix containing the eigenvectors of A. This is also called an eigendecomposition.

This theorem kind of gives the general flavor of spectral methods even if I can't find a definite definition. More generally, spectral methods (from what little I've read) seek to solve problems by using some form of matrix decomposition - typically by formulating their problem as a diagonalizable matrix. [1]

There are more mathy things / stuff you can prove here, but I'll refer to Karl Stratos's first chapter of his thesis.

1.2 Why Spectral?

- Generally fast since diagonalizable matrices are easy to deal with
- Gives strong algorithmic and theoretical guarantees
- Fairly easy to implement - my favorite

[1] From what I can tell, traditionally, along with a lot of the stuff I do here, spectral methods typically do some eigenvalue stuff/SVD. Exceptions are like in Arora et al 2012a and I think Parikh et al 2014.

2 Quick Review over SVD

The form $U\Lambda U^t$ should be familiar to you. The fact that Λ is a diagonalized matrix of the the eigenvalues λ should too. And the fact that U contains the eigenvectors. And if the title doesn't remind you, Kelvin, you're supposed to talk about SVD.

Theorem 2 (Singular Value Decomposition) *Let $A \in R^{mxn}$ be any matrix , $r = min(m, n)$, and $q = max(m, n)$ then we can decompose A as:*

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^t = USV^t$$

Where $S = diag(\sigma_1, ..\sigma_r, [0]^{q-r})$. and U and V are orthogonal matrices. $u_1, .., u_r$ and $v_1, .., v_r$ are the left and right singular values of A.

Note that SVD isn't the same as the eigendecomposition, but rather you can *derive* it. This actually helps prove what left and right singular values actually mean (meaning .

Theorem 3 *Let $\lambda_{[1,n]}$ be the eigenvalues of AA^T and $\lambda'_{[1,m]}$ be the eigenvalues of A^TA. The first r eigenvalues are the same, and correspond to σ^2.*

easier proof

$$AA^T = (USV^T)(USV^T)^T$$
$$= USV^BTS^TU^T = US^2U^T$$

3 CCA

Most of us know this. We want to maximize the *Pearson Correlation Coefficient*:

$$Cor(L, R) = \frac{Cov(L, R)}{var(L)var(R)} = \frac{E[LR] - E[L]E[R]}{\sqrt{E[L^2] - E^2[L]}\sqrt{E[R^2] - E^2[R]}}$$

More precisely, we want to create L and R by projecting our two views X and Y onto two projection vectors w and v. Our jobs are to find w and v. Assume wlog E[X], E[Y] = 0 [2]

[2] if that really bothers you, set $X_i = X_i - E[X]$ and $Y_i = Y_i - E[Y]$ and use the new data points

$$(w*, v*) = argmax_{w,b} Cor(w^T X, v^T Y)$$
$$= \frac{E[(w^T X)(v^T Y)^T] - E[w^T X]E[(v^T Y)^T]}{\sqrt{E[(w^T X)^2] - E^2[w^T X]}\sqrt{E[(v^T Y)^2] - E^2[v^T Y]}}$$
$$= \frac{E[(w^T X)(v^T Y)^T]}{\sqrt{E[(w^T X)^2]}\sqrt{E[(v^T Y)^2]}}$$
$$= \frac{w^T E[(X)(Y^T)]v}{\sqrt{E[(w^T X)^2]}\sqrt{E[(v^T Y)^2]}}$$
$$= \frac{w^T E[(X)(Y^T)]v}{\sqrt{w^T E[X^2]w}\sqrt{Ev^T[Y^2]v}}$$

Note that the scaling on w and v don't matter at all! That means we can arbitrarily set the norm to whatever we please. It's easiest if we set them =1.

The problem now is: how do we form this into a matrix? Use the covariance matrix C=[Cxx, Cxy; Cyx, Cyy]. This gives us:

$$(w*, v*) = argmax_{w,b} w^T C_{XY} w^T C_{XY} v$$

Let $\omega = C_{XX}^{-1/2} C_{XY} C_{YY}^{-1/2}$, $k = C_{XX}^{-1/2} w$, and $j = C_{YY}^{-1/2} v$. Now our optimization is:

$$(k*, j*) = argmax k^T \omega j$$

We can use SVD to solve for this. (trust me)

Sorry me in 8 hours but you're going to regurgitate this awfully texed mess somehow. Or skip it.

4 CCA for Word Embeddings

Naturally, two "views" of a word could be the word itself and some of the words in a window around said word. In other words, we can use CCA to find an low-D space where the word is maximally correlated with its context.

5 NMF

Let $A \in R^{n \times d}$ that satisfies:

1. have rank $r \leq min(n, d)$
2. $A_{i,j} \geq 0 \forall i, j$

We then want to find $B \in R^{n \times r}$ and $C \in R^{r \times d}$ such that $A = BC$. Yup. That's it.

Note that, when its unique, we have:

$$A_i = \sum_{j=1}^{r} B_{i,j} C_j$$

That is, A_i is a linear combination of $C_1, ..C_r$ where the scalars are the $B_{i,1}, ...B_{i,r}$. We'll look at an example.

5.1 NMF for Topic Models

Big picture: we're going to have three matrices with n words, d documents, and r topics:

1. $A : n \times d$: the corpus; words per document
2. $B : n \times r$: the words-topics matrix
3. $C : r \times d$: the topic-document matrix

Definition 4 (P-separability) *The matrix B is p-separable if for each column i in B, there exists some row where the only non-zero entry is in column i and $\geq p$.*

In other words, if row j in B has only one one non-zero entry p' in i, then any $B_j C_i = p' C_{ij}$ - A can be expressed as a convex combination of the *anchor rows* in C.

We can solve this by expanding the convex hull. Heuristically, note that the anchor rows span the other rows - that is anything within the convex hull is not an anchor row. Conversely, the extreme points are. If we iteratively remove the extreme points, we are constantly picking new anchor words.

References