

Dictionary Learning

Refs:

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5714407>

<https://youtu.be/Ri0ComuqS7Y>

Sample Code Based Tutorial:

<https://dictlearn.readthedocs.io/en/latest/tutorial.html>

Few Other Related Code Bases

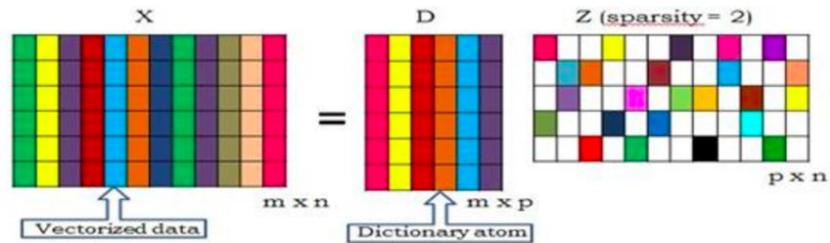
<https://github.com/topics/netflix-problem>

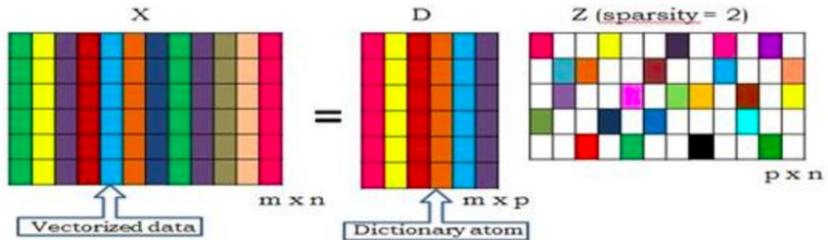
Dictionary Learning

→ Sparse dictionary learning / Sparse coding

Goal:

- Find sparse representation of input data
- Data is represented as linear combination of basic elements (atoms)
- Dictionary consists of atoms





Sparsity:

The sparsity term controls the number of nonzero entries of each column, can be more than 1, but less than a number .

Atoms:

- Do not need to be orthogonal
- May be over-complete spanning set.

What is overcomplete?

Complete Basis: set of vectors such that any vector in the vector space can be represented by a linear combination of vectors from the basis.

Overcomplete basis: complete when one or more vectors are removed.

Dictionary learning:

Input data: $X = [x_1, x_2, \dots, x_k]$, $x_i \in \mathbb{R}^d$
↑
feature vectors

Dictionary: $D \in \mathbb{R}^{d \times n}$: $[d_1, d_2, \dots, d_n]$

Learn a representation $R = [r_1, r_2, \dots, r_k]$,
 $r_i \in \mathbb{R}^n$

Minimize Reconstruction error]: $\min \|X - DR\|_F^2$
 r_i sparse.

→ We want to keep individual representations
sparse.

Example of sparse coding:

$$\begin{matrix} \boxed{7} \\ \text{input} \end{matrix} = 1. \begin{matrix} \boxed{1} \\ \text{from dictionary} \end{matrix} + 1. \begin{matrix} \boxed{-1} \\ \text{decomposition of signals} \\ (\text{may be pixels}) \end{matrix} + 1. \begin{matrix} \boxed{4} \\ \dots \end{matrix}$$

Dictionary

→ if undercomplete : $n < d$

↳ problem of dimensionality reduction
+ dictionary learning.

→ if overcomplete : $n > d$.

→ sparsity + dictionary learning

2
4

• Let's consider $D = \begin{bmatrix} 1 & 0 & 1 \\ -2 & -3 & 1 \\ 3 & 3 & 0 \end{bmatrix}$

Consider the vector $[1 \ 1 \ 0]^T$.

What will be the sparse solution
to y .

Ans : $[1 \ -1 \ 0]^T$.

Optimization problem

$$\underset{\substack{D \in C, \\ x_i \in R^n}}{\operatorname{argmin}} \sum_{i=1}^k \|x_i - Dx_i\|_2^2 + \gamma \|x_i\|_0$$

$\uparrow l_0\text{-norm}$

$$C = \{D \in R^{d \times n} : \|D_i\|_2 \leq 1 \forall i=1..n\}$$

\hookrightarrow Constraints on D

$$\gamma > 0.$$

$\rightarrow C$ prevents D from becoming arbitrarily large.

lo: total no of nonzero elements in a column

λ : Controls sparsity vs. error min

If λ = small, we care more
on minimization.

If λ = large, focus more on
sparsity.

Make the problem convex:

$$\underset{\substack{D \in C, r_i \in \mathbb{R}^n}}{\operatorname{argmin}} \sum_{i=1}^K \|x_i - Dr_i\|_2^2 + \gamma \|r_i\|_0$$

\uparrow
nonconvex



$$\underset{\substack{D \in C, r_i \in \mathbb{R}^n}}{\operatorname{argmin}} \sum \|x_i - Dr_i\|_2^2 + \gamma \|r_i\|_1$$

\uparrow
convex

Not Jointly Convex.

What is jointly convex?

Joint convexity means the restriction of the function to any segment is a convex function of a single variable.

How to solve the optimization?

↳ Method of optimal directions.

→ K-SVD

Method of Optimal Directions:

$$\min_{D, R} \left\{ \|X - DR\|_F^2 \right\} \text{ s.t. } \forall \|r_i\|_0 \leq T$$

↓
Probenius norm

Method of Optimal Directions:

$$\min_{D, R} \left\{ \|X - DR\|_F^2 \right\} \quad \text{s.t. } \|r_i\|_0 < T$$

↓
Frobenius norm

$$\|A\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{i,j}|^2}$$

$\|r_i\|_0 < T \Rightarrow$ each representation is
having fewer than
+ non-zero elements

\Rightarrow sparses. are removed from
minimization and added in constraint

Alternate version: (using Matching Pursuit)

1. Find R using matching pursuit with D fixed.
2. Update dictionary by solving

$$D = X R^+$$

R^+ ← pseudoinverse

3. Renormalize D to fit constraints (C)
4. Repeat upto convergence (small residual)

What is R^+ ?

• R^+ is the pseudoinverse

→ Inverse exists for square matrix only.

→ Real data is not always consistent.

For any $m \times n$ matrix, the pseudoinverse
of A is defined as:

1. Full column matrix (Rank = Column no.):

$$A^+ = (A^T \cdot A)^{-1} \cdot A^T$$

2. Full row matrix (Rank = Row no.)

$$A^+ = A^T \cdot (A \cdot A^T)^{-1}$$

Pseudo inverse:

3. If rank deficient: ($R < m$ and $R < n$)

$$\begin{aligned} A^+ &= (U\Sigma V^T)^{-1} \\ &= (V^T)^{-1} \Sigma^{-1} U^{-1} \\ &= V \Sigma^{-1} U^T \quad [\text{since } U \& V \text{ orthogonal}] \end{aligned}$$

If $\Sigma = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \\ 0 & 0 \end{bmatrix} \Rightarrow \Sigma^{-1} = \begin{bmatrix} 1/\sigma_1 & 0 & 0 \\ 0 & 1/\sigma_2 & 0 \end{bmatrix} U^{-1} = U^T$

Pseudoinverse (Example)

Find the inverse of :

$$1. \ A = \begin{bmatrix} 1 & 2 & 1 & 3 \\ 4 & 3 & 2 & 1 \end{bmatrix}$$

$$2. \ A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \end{bmatrix}$$

[Clue °] If we can find a minor/minors
whose (determinant of nonzero submatrix)
 \Rightarrow rank of matrix = order of a
highest order non-vanishing
minor]

What is matching pursuit?

→ Sparse approximation algorithm

→ approximate a function $f(t)$ using weighted sum of atoms:

$$f(t) \approx \hat{f}_N(t) = \sum_{n=1}^N \overset{\text{NP}}{a_n} \cdot \boxed{g_m(t)}$$

↑
 approximation
 .
 ↑
 atom

[χ_n = index of atom]

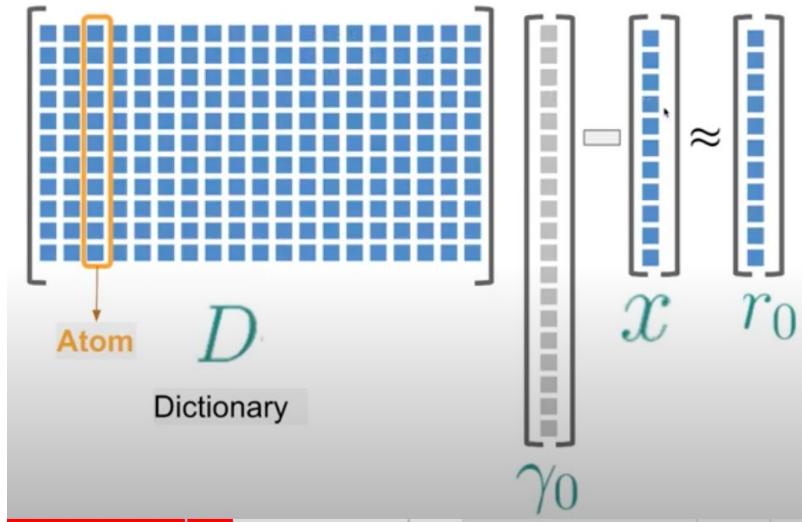
→ Try to get smaller number of atoms from dictionary.

→ an indicates elements of R_i .

Basic idea of matching pursuit.

- Choose atom one at a time to reduce the approximation error.
- find atom with the highest inner product with the input
- Subtract the inner product with atom from original input
- Repeat until norm of residual is small

A Visual Representation and Code Based Example



Gamma (γ) value is chosen so that difference can be minimized.

*Ref:

<https://www.youtube.com/watch?v=sK6jTVo0Bhk&t=643s>

$$\left[\begin{array}{c|ccccc} D & \gamma_1 & x & r_1 < r_0 \\ \hline & & & & \end{array} \right]$$

$$\left[\begin{array}{c|cc|c} D & \text{blue blocks} & \text{magenta block} & \text{blue blocks} \end{array} \right] \xrightarrow{\gamma_2} \left[\begin{array}{c|c} x & r_2 \end{array} \right] \approx \left[\begin{array}{c} r_2 \end{array} \right]$$

```

#Input:  $f(t)$ ,  $D$  (with normalized columns  $g_i$ )
#Output: List of coefficients  $a$  and indices for the atoms  $\gamma$ 

#Initialize
 $R_1 \leftarrow f(t)$ 
 $n \leftarrow 1$ 

While  $\|R_n\| > threshold$ :
    Find  $g_{\gamma_n} \in D$  with maximum inner product  $|\langle R_n, g_{\gamma_n} \rangle|$ 
     $a_n \leftarrow \langle R_n, g_{\gamma_n} \rangle$ 
     $R_{n+1} \leftarrow R_n - a_n g_{\gamma_n}$ 
     $n \leftarrow n + 1$ 

Return  $a, \gamma$ 

```

MoD algorithm :

→ Efficient for low dimensional data

→ Pseudo inverse (R^+) computation
is complex

Another option : K-SVD.

K-SVD

Extension of K-Means.

K-Means:

$$\min_{D, R} \|x - DR\|_F^2 \quad \text{s.t. } \forall i \|z_i\|_0 = 1$$

D, R



Cluster center

Each sample:
→ One hot vector

→ '1' component is in the nearest
cluster and '0' in every other
part.

K-SVD

$$\min \left\{ \|X - DR\|_F^2 \right\} \text{ s.t. } \forall i, \|r_i\|_0 \leq T$$

→ Relaxation over K-means

→ Where T - non-zero elements
are allowed

K-SVD :

- Fix the dictionary
- Find the best R using orthogonal Matching Pursuit
- Iteratively update atoms of D

What is orthogonal Matching Pursuit?

What is orthogonal Matching Pursuit?

- start with matching pursuit steps
- find atom with the highest inner product with the signal

→ update all coefficients (a_n) so far:
• computing orthogonal projection of
the signal onto the subspace
spanned by the set of atoms
selected so far.

- start with matching pursuit steps
- find atom with the highest inner product with the signal

→ update all coefficients (a_m) so far:

• Computing orthogonal projection of the signal onto the subspace spanned by the set of atoms selected so far.

→ subtract $R_n - a_m g_{x_n}$.

→ Repeat until norm of residual is small

Update D :

$$\begin{aligned} \|x - DR\|_F^2 &= \|x - d_j r_T^j\|_F^2 \rightarrow [\text{update one column of } D \text{ at a time}] \\ &= \left\| \left(x - \sum_{j \neq k} d_j r_T^j \right) - d_k r_T^k \right\|_F^2 \\ &= \|x_K - d_k r_T^k\|_F^2 \end{aligned}$$

[\rightarrow Consider individual vector products]

[separate k -th column of d]

SVD to update D

$$\|X - DR\|_F^2 = \|E_K - d_K r_T^K\|_F^2$$

① approximate
with rank-1
matrices using SVD

② update d_K to
minimize reconstruction error

Limitation : like MOD suffers for high dimensional data