Dictionary Learning Applications in Control Theory

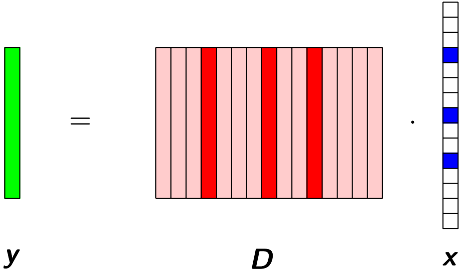
Paul Irofti, Florin Stoican

Politehnica University of Bucharest
Faculty of Automatic Control and Computers
Department of Automatic Control and Systems Engineering
Email: paul@irofti.net, florin.stoican@acse.pub.ro

Recent Advances in Artificial Intelligence, June 20th, 2017

Acknowledgment: This work was supported by the Romanian National Authority for Scientific Research, CNCS - UEFISCDI, project number PN-II-RU-TE-2014-4-2713.

Sparse Representation (SR)



Orthogonal Matching Pursuit (OMP)

Algorithm 1: OMP[a]

```
Arguments: D, y, s
Initialize: r = y, I = ∅
for k = 1 : s do
    Compute correlations with residual: z = D^T r
    Select new column: i = arg max_j |z_j|
    Increase support: I ← I ∪ {i}
    Compute new solution: x = LS(D, y, I)
    Update residual: r = y − D_I x_I
```

Arguments: $\boldsymbol{D}$, $\boldsymbol{y}$, s

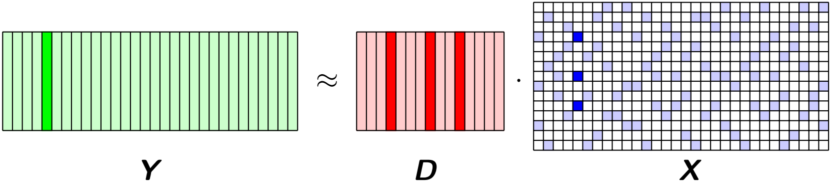
Initialize: $\boldsymbol{r} = \boldsymbol{y}$, $\mathcal{I} = \emptyset$

for $k = 1 : s$ **do**

- Compute correlations with residual: $\boldsymbol{z} = \boldsymbol{D}^T \boldsymbol{r}$
- Select new column: $i = \arg\max_j |z_j|$
- Increase support: $\mathcal{I} \leftarrow \mathcal{I} \cup \{i\}$
- Compute new solution: $\boldsymbol{x} = \mathrm{LS}(\boldsymbol{D}, \boldsymbol{y}, \mathcal{I})$
- Update residual: $\boldsymbol{r} = \boldsymbol{y} - \boldsymbol{D}_{\mathcal{I}} \boldsymbol{x}_{\mathcal{I}}$

[a] Pati, Rezaiifar, and Krishnaprasad 1993.

Dictionary Learning (DL)



The Dictionary Learning (DL) Problem

Given a data set $Y \in \mathbb{R}^{p \times m}$ and a sparsity level s, minimize the bivariate function

$$\begin{aligned} \underset{\boldsymbol{D},\boldsymbol{X}}{\text{minimize}} \quad & \|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}\|_F^2 \\ \text{subject to} \quad & \|\boldsymbol{d}_j\|_2 = 1,\ 1 \leq j \leq n \\ & \|\boldsymbol{x}_i\|_0 \leq s,\ 1 \leq i \leq m, \end{aligned} \tag{1}$$

where $D \in \mathbb{R}^{p \times n}$ is the dictionary (whose columns are called atoms) and $X \in \mathbb{R}^{n \times m}$ the sparse representations matrix.

Approach

Algorithm 2: Dictionary learning – general structure

Arguments: signal matrix $\boldsymbol{Y}$, target sparsity s
Initialize: dictionary $\boldsymbol{D}$ (with normalized atoms)
for $k = 1, 2, \ldots$ **do**
    With fixed $\boldsymbol{D}$, compute sparse representations $\boldsymbol{X}$
    With fixed $\boldsymbol{X}$, update atoms $\boldsymbol{d}_j$, $j = 1 : n$

DL Algorithms

K-SVD[1] solves the optimization problem in sequence

$$\min_{\boldsymbol{d}_j, \boldsymbol{X}_{j,\mathcal{I}_j}} \left\| \left(\boldsymbol{Y}_{\mathcal{I}_j} - \sum_{\ell \neq j} \boldsymbol{d}_\ell \boldsymbol{X}_{\ell,\mathcal{I}_\ell} \right) - \boldsymbol{d}_j \boldsymbol{X}_{j,\mathcal{I}_j} \right\|_F^2 \tag{2}$$

where all atoms excepting d_j are fixed.

This is seen as a rank-1 approximation and the solution is given by the singular vectors corresponding to the largest singular value.

$$\boldsymbol{d}_j = \boldsymbol{u}_1, \quad \boldsymbol{X}_{j,\mathcal{I}_j} = \sigma_1 \boldsymbol{v}_1. \tag{3}$$

[1] Aharon, Elad, and Bruckstein 2006.

LC-KSVD

$$\begin{aligned} \underset{\boldsymbol{D},\boldsymbol{X},\boldsymbol{A},\boldsymbol{W}}{\text{minimize}} \quad & \|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}\|_F^2 + \alpha\|\boldsymbol{Q} - \boldsymbol{A}\boldsymbol{X}\|_F^2 + \beta\|\boldsymbol{H} - \boldsymbol{W}\boldsymbol{X}\|_F^2 \\ \text{subject to} \quad & \|\boldsymbol{d}_j\|_2 = 1,\ 1 \le j \le n \\ & \|\boldsymbol{x}_i\|_0 \le s,\ 1 \le i \le m, \end{aligned} \tag{4}$$

- dictionary atoms evenly split among classes
- $\boldsymbol{q}_i$ has non-zero entries where $\boldsymbol{y}_i$ and $\boldsymbol{d}_i$ share the same label.
- linear transformation $\boldsymbol{A}$ encourages discrimination in $\boldsymbol{X}$
- $\boldsymbol{h}_i = \boldsymbol{e}_j$ where j is the class label of $\boldsymbol{y}_i$
- $\boldsymbol{W}$ represents the learned classifier parameters

Fault Detection and Isolation in Water Networks

FDI via DL

Water networks pose some interesting issues:

- large scale, distributed network with few sensors
- user demand unknown or imprecise
- pressure dynamics nonlinear (analytic solutions impractical)

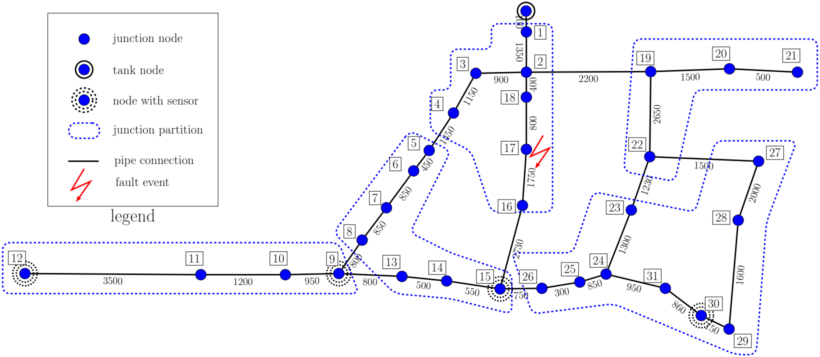
The DL approach for FDI:

- a residual signal compares expected and measured pressures

$$r_i(t) = p_i(f_i(t), f_j(t), t) - \bar{p}_i, \ \forall i, j \tag{5}$$

- to each fault is assigned a class and DL provides the atoms which discriminate between them
- each residual is sparsely described by atoms and thus, FDI is achieved iff the classification is unambiguous

Hanoi



junction node
tank node
node with sensor
junction partition
pipe connection
fault event
legend
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
1350
900
2200
1500
500
400
800
1750
1150
1450
450
850
850
2650
1500
1230
2000
1300
1600
3500
1200
950
800
500
550
2730
300
850
950
860

Sensor Placement

Let $\boldsymbol{R} \in \mathbb{R}^{n \times mn}$ be measured pressure residuals in all n network nodes. For each node we simulate m different faults.

Given $s < n$ available sensors, apply OMP on each column $\boldsymbol{r}$

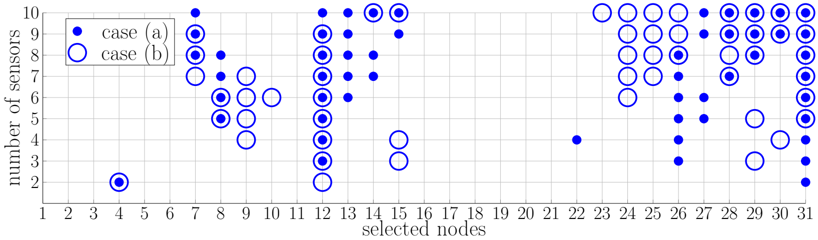
$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \|\boldsymbol{r} - \boldsymbol{I}_n \boldsymbol{x}\|_2^2 \\ \text{subject to} \quad & \|\boldsymbol{x}\|_0 \leq s, \end{aligned} \tag{6}$$

resulting in matrix $\boldsymbol{X}$ with s-sparse columns approximating $\boldsymbol{R}$.

Placement Strategies

(a) select the s most common used atoms

(b) from each m-block select most frequent s atoms; of the $n \cdot s$ atoms, select again the first s.



Learning

Algorithm 3: Placement and FDI learning[a]

Inputs: training residuals $\boldsymbol{R} \in \mathbb{R}^{n \times nm}$
parameters s, α, β
Result: dictionary $\boldsymbol{D}$, classifier $\boldsymbol{W}$, sensor nodes $\mathcal{I}_s$

Select s sensor nodes $\mathcal{I}_s$ based on matrix $\boldsymbol{R}$ using (a) or (b)
Let $\boldsymbol{R}_{\mathcal{I}_s}$ be the restriction of $\boldsymbol{R}$ to the rows in $\mathcal{I}_s$
Use $\boldsymbol{R}_{\mathcal{I}_s}$, α and β to learn $\boldsymbol{D}$ and $\boldsymbol{W}$ from (4)

[a] **Irofti** and Stoican 2017.

Fault Detection

Algorithm 4: Fault detection and isolation

```
Inputs: testing residuals R ∈ ℝ^{s×mn}
        dictionary D, classifier W
Result: prediction P ∈ ℕ^{mn}
for k = 1 to mn do
    Use OMP to obtain x_k using r_k and D
    Label: L_k = W x_k
    Classify: p_k = arg max_c L_k
```

Inputs: testing residuals $\boldsymbol{R} \in \mathbb{R}^{s \times mn}$

dictionary $\boldsymbol{D}$, classifier $\boldsymbol{W}$

Result: prediction $\boldsymbol{P} \in \mathbb{N}^{mn}$

for $k = 1$ **to** mn **do**

- Use OMP to obtain $\boldsymbol{x}_k$ using $\boldsymbol{r}_k$ and D
- Label: $\boldsymbol{L}_k = \boldsymbol{W}\boldsymbol{x}_k$
- Classify: $p_k = \arg\max_c \boldsymbol{L}_k$

Position c of the largest entry from $\boldsymbol{L}_k$ is the predicted class.

Today

Improved sensor placement. Iteratively choose s rows from $\boldsymbol{R}$ solving at each step

$$i = \arg\min_{k} \quad \|\text{proj}_{\boldsymbol{R}_{\mathcal{I}}} \boldsymbol{r}_k\|_2^2 + \lambda \frac{1}{\|\boldsymbol{\delta}_{k,\mathcal{I}}\|_1}, \boldsymbol{r}_k \in \boldsymbol{R}_{\mathcal{I}^c}, \tag{7}$$

where $\mathcal{I}$ is the set of currently selected rows and $\boldsymbol{\delta}_{k,\mathcal{I}}$ vector elements are the distances from node k to the nodes in $\mathcal{I}$.

Graph aware DL. Adding graph regularization[2]

$$\begin{aligned} &\|\boldsymbol{Y} - \boldsymbol{DX}\|_F^2 + \alpha \|\boldsymbol{Q} - \boldsymbol{AX}\|_F^2 + \beta \|\boldsymbol{H} - \boldsymbol{WX}\|_F^2 + \\ &\qquad + \gamma \text{Tr}(\boldsymbol{D}^T \boldsymbol{LD}) + \lambda \text{Tr}(\boldsymbol{XL}_c \boldsymbol{X}^T) + \mu \|\boldsymbol{L}\|_F^2, \end{aligned} \tag{8}$$

where $\boldsymbol{L}$ is the graph Laplacian.

[2] Yankelevsky and Elad 2016.

Zonotopic Area Coverage

Zonotopic sets

Area packing, mRPI (over)approximation and other related notions may be described via unions of zonotopic sets:

$$\min_{Z_k} \mathrm{vol}(S) - \mathrm{vol}\left(\bigcup_k Z_k\right), \tag{9}$$
$$\text{subject to } Z_k \subseteq S.$$

Zonotopes, given in generator representation[3]

$$Z_k = \mathcal{Z}(c_k, G_k) = \{c_k + G_k\xi : \|\xi\|_\infty \leq 1\} \tag{10}$$

are easy to handle for:

- Minkowski sum:
  $\mathcal{Z}(G_1, c_1) \oplus \mathcal{Z}(G_2, c_2) = \mathcal{Z}(\begin{bmatrix} G_1 & G_2 \end{bmatrix}, c_1 + c_2)$
- linear mappings: $R\mathcal{Z}(G_1, c_1) = \mathcal{Z}(RG_1, Rc_1)$

[3] Fukuda 2004.

Formulation

Each zonotope is parameterized after its center and a scaling vector (c_k, λ_k). These variables help formulate the:

- inclusion constraint $\mathcal{Z}\left(c_k, G \cdot \mathrm{diag}(\lambda_k)\right) \subseteq U$:

$$s_i^\top c_k + \sum_j |s_i^\top G_j| \lambda_{jk} \leq r_i, \quad \forall i, \tag{11}$$

  where $U = \{u : \; s_i^\top u \leq r_i\}$.

- explicitly describe the volume[4] $\mathrm{vol}\left(\mathcal{Z}(c_k, G\lambda_k)\right)$:

$$\mathrm{vol}(\mathcal{Z}(c_k, G\Lambda_k)) = \sum_{1 \leq j_1 < \ldots j_n \leq N} \left|\det(G^{j_1 \ldots j_n})\right| \cdot \prod_{j \in \{j_1 \ldots j_n\}} \lambda_{jk}. \tag{12}$$

The formulation becomes simpler if the scaling is homogeneous $(\lambda_k^* = \lambda_{jk}, \forall j)$.

[4] Gover and Krikorian 2010.

Implementation

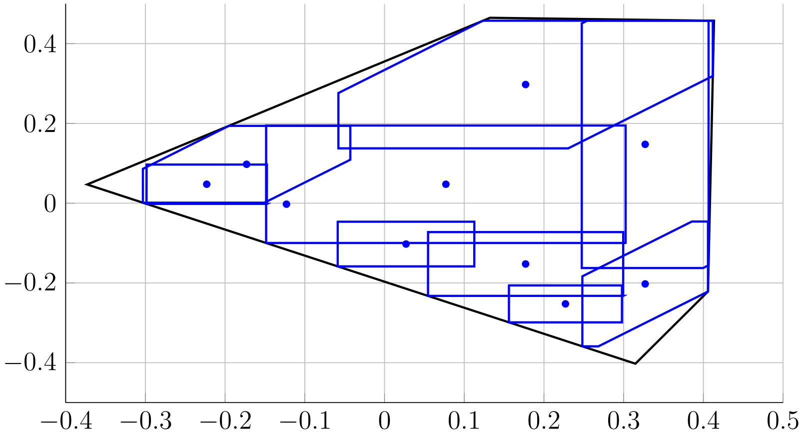
We track the OMP formalism, without its theoretical convergence guarantees:

Algorithm 5: Area Coverage with zonotopic sets

Inputs: area to be covered U, sparsity constraint s
Result: pairs of centers and scaling factors (c_k, λ_k)

for $k = 1$ **to** s **do**
    Enlarge the zonotopes until they saturate the constraints
    Select Z_k where $k = \arg\min_k \mathrm{vol}(S_k \setminus Z_k)$
    Update the uncovered area $\mathrm{vol}(S_{k+1}) = \mathrm{vol}(S_k \cup Z_k)$

Result



Thank You!

Questions?