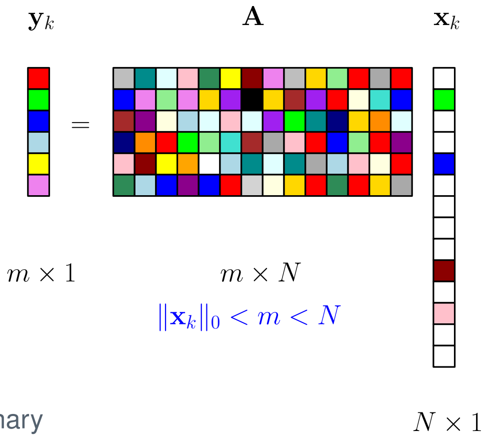
# A Bayesian Algorithm for Joint Dictionary Learning and Sparse Signal Recovery

Geethu Joseph

February 4, 2017

# Sparse Representation



$\|\mathbf{x}_k\|_0 < m < N$

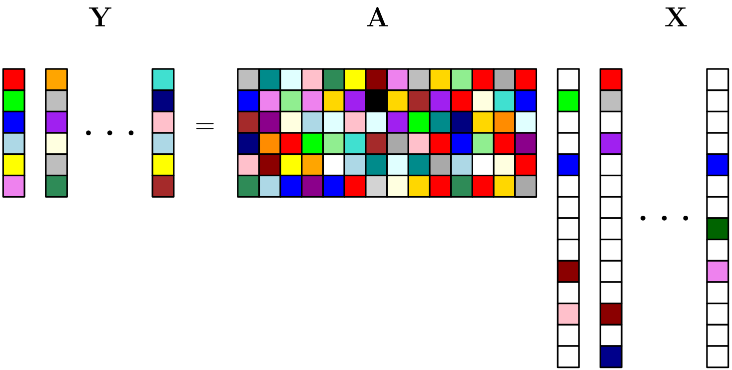
- ***A***: Dictionary
- $\boldsymbol{x}_k$: Sparse representation

# Choice of Dictionary

1. Predefined dictionary - non-adaptive
   - Fourier, Discrete Cosine Transform, Wavelet
2. **Learned dictionary** - better-adapted to signal
   - often leads to more compact representation[†]

[†] M. Elad, “Sparse and Redundant Representations”, Springer, 2010
J. Mairal, et.al., ”Task-driven dictionary learning,", IEEE Trans. Patt. Anal. Mach. Intell., 2012

# Dictionary Learning




- Matrix factorization problem: Learn both ***A*** and sparse ***X***

# System Model

- A set of $K$ training signals

$$\boldsymbol{y}_k = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{w}_k, \qquad k = 1, 2, \ldots, K$$

- Measurement noise $\boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$

- Ambiguity in amplitude: **all columns of *A* has unit norm**
- Assumption: Knowledge of $N$

# Sparse Bayesian Learning Framework*

**Fictitious prior on $\boldsymbol{x}_k$**

$$\begin{aligned} \boldsymbol{x}_k &\sim \mathcal{N}(0, \boldsymbol{\Gamma}_k) \\ \boldsymbol{y}_k | \boldsymbol{x}_k &\sim \mathcal{N}(\boldsymbol{A}\boldsymbol{x}_k, \sigma^2 \boldsymbol{I}) \end{aligned}$$

$\boldsymbol{\Gamma}_k = \text{Diag}\{\gamma_k\} \in \mathbb{R}_+^{N \times N}$

## Estimation method: Type II ML estimation

1. Learn parameters $\gamma_k$ and $\boldsymbol{A}$ that maximizes $-\log p(\boldsymbol{y}^K; \boldsymbol{\Lambda})$
2. Estimate $\boldsymbol{X}$ using the estimates of parameters

* D. P. Wipf and B. D. Rao, "An empirical Bayesian strategy for solving the simultaneous sparse approximation problem," TSP 2007

# Parameter Learning

- **Expectation-maximization algorithm** with $\boldsymbol{x}_k$ as hidden data

**Expectation-Maximization Algorithm**

$$\textbf{E-step: } Q\left(\boldsymbol{\Lambda}, \boldsymbol{\Lambda}^{(r-1)}\right) = \mathbb{E}_{\boldsymbol{x}^K | \boldsymbol{y}^K; \boldsymbol{\Lambda}^{(r-1)}} \left\{\log p\left(\boldsymbol{y}^K, \boldsymbol{x}^K; \boldsymbol{\Lambda}\right)\right\}$$

$$\textbf{M-step: } \boldsymbol{\Lambda}^{(r)} = \arg\max_{\boldsymbol{\Lambda} \in \mathbb{\Lambda}} Q\left(\boldsymbol{\Lambda}, \boldsymbol{\Lambda}^{(r-1)}\right).$$

- Tuple of unknown parameters: $\boldsymbol{\Lambda} = \{\boldsymbol{A}, \gamma_k k = 1, 2, \ldots K\}$
- Feasible set: $\mathbb{\Lambda} = \left\{\boldsymbol{A} \in \mathbb{R}^{m \times N} : \boldsymbol{A}_i^\mathsf{T} \boldsymbol{A}_i = 1, \forall i\right\} \times \mathbb{R}^{KN}$

# EM Algorithm

**E-step:Update the statistics of $\boldsymbol{x}_k$**

- Statistics: mean and covariance
- Closed form expressions in terms of parameters

$\downarrow\uparrow$

**M-step: Update the parameters**

- Separable in variables: $\boldsymbol{A}$ and $\gamma_k$
- Closed form expression for $\gamma_k$ update
- Non-convex optimization problem corresponding to $\boldsymbol{A}$ update

# Dictionary Update

- Non-convex optimization problem

$$\underset{\boldsymbol{A}:\boldsymbol{A}_i^\mathsf{T}\boldsymbol{A}_i}{\arg\min} \; -\mathrm{Tr}\left\{\boldsymbol{M}\boldsymbol{Y}^\mathsf{T}\boldsymbol{A}\right\} + \frac{1}{2}\mathrm{Tr}\left\{\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^\mathsf{T}\right\},$$

  - $\boldsymbol{M}$ and $\boldsymbol{\Sigma}$: functions of statistics of $\boldsymbol{x}_k$

- Closed form solution if $\boldsymbol{\Sigma}$ is a diagonal matrix

- Solved using **alternating minimization** procedure
  - Update one column of $\boldsymbol{A}$ at a time
  - Closed form updates

# Overall Algorithm

## E-step: Update $\boldsymbol{\Sigma}^{(k)}, \boldsymbol{\mu}_k$

**for** $k = 1, \ldots, K$

$$\boldsymbol{\Phi} = \left(\sigma^2 \boldsymbol{I} + \boldsymbol{A}^{(r)} \boldsymbol{\Gamma}_k^{(r)} \boldsymbol{A}^{(r)}\right)^{-1}$$

$$\boldsymbol{\Sigma}^{(k)} = \boldsymbol{\Gamma}_k^{(r)} \left(\boldsymbol{I} - \boldsymbol{A}^{(r)\mathsf{T}} \boldsymbol{\Phi} \boldsymbol{A}^{(r)\mathsf{T}} \boldsymbol{\Gamma}_k^{(r)}\right)$$

$$\boldsymbol{\mu}_k = \sigma^{-2} \boldsymbol{\Sigma}^{(k)} \boldsymbol{A}^{(r)\mathsf{T}} \boldsymbol{y}_k$$

## M-step: Update $\boldsymbol{A}$ and $\gamma_k$

**for** $k = 1, \ldots, K$

$$\gamma_k^{(r)} = \text{Diag}\left\{\boldsymbol{\mu}_k \boldsymbol{\mu}_k^{\mathsf{T}} + \boldsymbol{\Sigma}^{(k)}\right\}$$

$$\boldsymbol{\Sigma} = \sum_{k=1}^{K} \boldsymbol{\mu}_k \boldsymbol{\mu}_k^{\mathsf{T}} + \boldsymbol{\Sigma}^{(k)}$$

### AM: Update $\boldsymbol{A}$

**for** $i = 1, 2, \ldots, N$

$$\boldsymbol{v} = \left(\boldsymbol{Y}\boldsymbol{M}^{\mathsf{T}}\right)_i - \sum_{j=1}^{i-1} \boldsymbol{\Sigma}[i,j] \hat{\boldsymbol{A}}_j^{(r,u)} - \sum_{j=i+1}^{N} \boldsymbol{\Sigma}[i,j] \hat{\boldsymbol{A}}_j^{(r,u-1)}$$

$$\hat{\boldsymbol{A}}_i^{(r,u)} = \begin{cases} \frac{1}{\|\boldsymbol{v}\|} \boldsymbol{v} & \text{if } \boldsymbol{v} \neq \boldsymbol{0} \\ \hat{\boldsymbol{A}}_i^{(r,u-1)} & \text{otherwise.} \end{cases}$$

# AM procedure Converges to Nash equilibrium

## Proposition

*The sequence of function values* $\left\{ g\left(\hat{\boldsymbol{A}}^{(u)}\right)\right\}_{u\in\mathbb{N}}$ *generated by the AM procedure converges, and every subsequential limit* $\hat{\boldsymbol{A}}$ *of the sequence* $\left\{\hat{\boldsymbol{A}}^{(u)}\right\}_{u\in\mathbb{N}}$ *is a Nash equilibrium point, namely,*

$$g\left(\hat{\boldsymbol{A}}_1,\ldots,\hat{\boldsymbol{A}}_{i-1},\hat{\boldsymbol{A}}_i,\hat{\boldsymbol{A}}_{i+1},\ldots,\hat{\boldsymbol{A}}_N\right) \leq g\left(\hat{\boldsymbol{A}}_1,\ldots,\hat{\boldsymbol{A}}_{i-1},\boldsymbol{a},\hat{\boldsymbol{A}}_{i+1},\ldots,\hat{\boldsymbol{A}}_N\right),$$

*for any vector* $\boldsymbol{a}$ *with unit norm and for* $i = 1, 2, \ldots, N$.

# AM procedure Converges to Stationary Point

**Theorem**

*For any initialization of the AM procedure $\hat{\boldsymbol{A}}^{(0)}$ such that $g\left(\hat{\boldsymbol{A}}^{(0)}\right) < \infty$, the sequence $\left\{g\left(\hat{\boldsymbol{A}}^{(u)}\right)\right\}_{u\in\mathbb{N}}$ generated by the AM procedure converges to a stationary point of the optimization problem. Moreover, the stationary point is not a local maxima.*

**Proof.**

Using Łojasiewicz gradient inequality □

- Initialization need not be a feasible point

# AM procedure: Sublinear Rate of Convergence

**Theorem**

*For any initialization of the AM procedure $\hat{\boldsymbol{A}}^{(0)}$ such that $g\left(\hat{\boldsymbol{A}}^{(0)}\right) < \infty$, there exists $C > 0$ such that the sequence $\left\{g\left(\hat{\boldsymbol{A}}^{(u)}\right)\right\}_{u\in\mathbb{N}}$ generated by the AM procedure satisfies*

$$\left\|\hat{\boldsymbol{A}}^{(u)} - \hat{\boldsymbol{A}}\right\| \leq C/u.$$

**Proof.**

Using Łojasiewicz exponent □

- Independent of system dimensions

# Summary

- Proposed a joint dictionary learning and sparse signal recovery algorithm
- Formulated using SBL framework
- Implemented using EM algorithm with AM procedure
- Convergence properties of AM procedure is studied