EMERGENCE OF SIMPLE-CELL RECEPTIVE PROPERTIES BY LEARNING A SPARSE CODE FOR NATURAL IMAGES

Bruno A. Olshausen & David J. Field

Presenter: Ozgur Yigit Balkan

Outline

- Linear data representations
- Sparse Vector Recovery
  - Existing Methods
- Dictionary Learning (Sparse Coding)
- Why is sparseness important?
- How to learn the dictionary?
- Experiments
- Conclusion

Linear Data Representation

- PCA – ICA – Sparse Coding
- Underlying model is linear.
- Each data point is assumed to be created with the following linear model

$$y = A\,x$$

- $y \in R^M$, is the data vector (observed)
- $A \in R^{M \times N}$, is the set of basis vectors (columns of A are basis vectors a_i).
- $x \in R^N$ is the weights vector. (unknown)

What is Sparse Coding?

$$Y = AX$$

- PCA: Given $Y = \{y_1, y_2, .. y_L\}$. Creates an orthogonal basis set A, such that the underlying sources (weights) are uncorrelated.
- ICA: Given $Y = \{y_1, y_2, .. y_L\}$. Creates a basis set of vectors as columns of A, such that the underlying sources are independent.
- Sparse Coding: Given $Y = \{y_1, y_2, .. y_L\}$. Find set of basis vectors A such that the associated vectors x_i are sparse.
  - Usually called “dictionary learning”.
  - $[y_1\ y_2\ \ldots y_L] = A\,[x_1\ x_2\ \ldots x_L]$

Sparse Vector Recovery

- Measure of sparseness?
- $\|x\|_p^p = \left(\sum_{j=1}^{N}(x^p)\right)$.
- Diversity: Number of nonzero elements in a vector.
- $D(x) = \|x\|_p^p$ as $p \to 0$.
- If $x \in R^N$, the measure of sparseness is: $N - \|x\|_p^p$ as $p \to 0$.
- Sparse Inverse Problem:

$$\min \|x\|_0 \quad \text{s.t} \quad y = Ax$$

- Used columns in A are called the support set. Finding support set is equivalent to finding x. Global solution is NP-hard.

Some of the existing methods

- Convex relaxation of objective function $\|x\|_0$

$$\min \|x\|_1 \quad \text{s.t} \quad y = Ax$$

- In noisy cases,

$$\min \|x\|_1 \quad \text{s.t} \quad \|y - Ax\|_2 \leq \epsilon$$

- Other possible formulations

$$\min \|y - Ax\|_2 \quad \text{s.t} \quad \|x\|_1 \leq k$$

- or,

$$\min \|y - Ax\|_2 + \lambda\|x\|_1$$

Greedy Pursuit Methods

- These methods, in general, starts with an empty support set and adds one of the basis vectors from A at each iteration.
- $A_S \triangleq$ current support set (initially empty)
- Orthogonal Matching Pursuit

$$\underset{i}{\operatorname{argmax}} \ |\langle r^k, a_i \rangle|$$

$$A_S = \mathrm{A_S} \cup \{a_i\}$$

- Order Recursive Matching Pursuit

$$\underset{i}{\operatorname{argmin}} \ \|r_{S\cup\{i\}}\|_2$$

$r_S = y - P_S y$
$= P_S{}^{\perp} y.$

How well do they work?

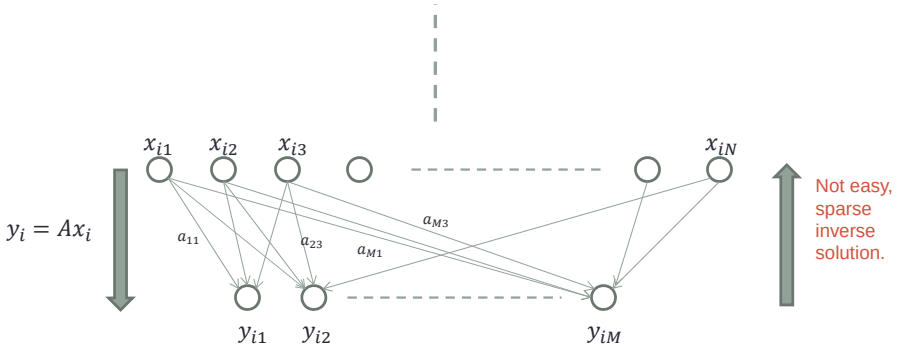
- Theoretical: These algorithms guarantee to find the global solution if A satisfy certain conditions, which are very conservative conditions and doesn't apply to real world applications.

$$\|x\|_0 < \frac{1}{2}(1 + \frac{1}{\mu})$$

- Practical: They work really well, especially convex relaxation methods.
- Note: In practice, one doesn't necessarily need the sparsest solution.

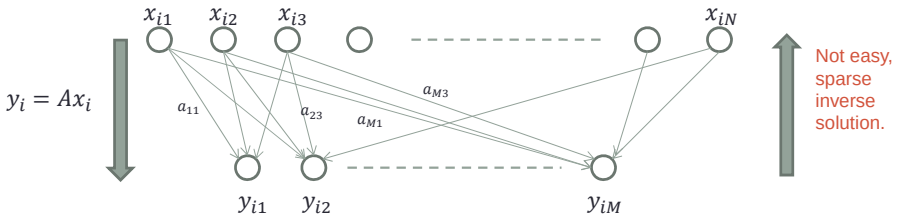
Sparse Coding (Dictionary Learning)

- Problem definition: Given data $\{y_1, y_2, \dots, y_n\}$, find dictionary A, such that associated x_i could be sparse. (diversity $< M$)



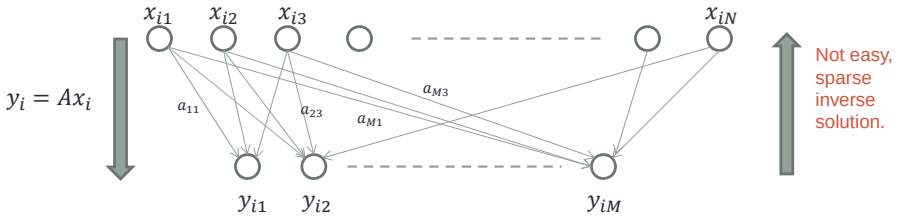
Why sparseness?

- Each cell has a low probability of activation given a single image (subimage block).
- This simplifies feature detection, e.g. edge detection
- Storage and retrieval with associative memory. (For higher levels of the network)



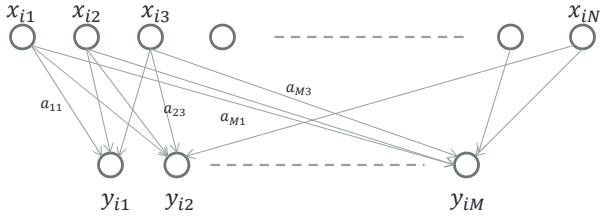
Why sparseness?

- Sparseness also results in independence.
- $H(x_{.1}, x_{.2}, \dots, x_{.N}) = \sum_j H(x_{ij}) - \text{mutual info.}$
- Under the assumption that joint entropy (information in the image) is preserved, if we minimize individual entropies, mutual info gets smaller. Thus, we would get more independent cells.
- $H(x_{ij}) \downarrow \rightarrow$ sparsity providing prior distributions.
- Minimum entropy code



How to learn A or a_{ij}?

- Assume we are given x_i's. So we know X, Y.
- We know the model Y = AX.
- $A_{sol} = \underset{A}{\operatorname{argmin}} \|Y - AX\|_F^2 = YX^T(XX^T)^{-1} = YX^+.$
- However we don't know x_i's upfront.



How to learn A?

Task: Train a dictionary $\mathbf{A}$ to sparsely represent the data $\{\mathbf{y}_i\}_{i=1}^M$, by approximating the solution to the problem posed in Equation (12.2).
Initialization: Initialize $k = 0$, and

- **Initialize Dictionary:** Build $\mathbf{A}_{(0)} \in \mathbb{R}^{n \times m}$, either by using random entries, or using m randomly chosen examples.
- **Normalization:** Normalize the columns of $\mathbf{A}_{(0)}$.

Main Iteration: Increment k by 1, and apply

- **Sparse Coding Stage:** Use a pursuit algorithm to approximate the solution of

$$\hat{\mathbf{x}}_i = arg\min_{\mathbf{x}} \|\mathbf{y}_i - \mathbf{A}_{(k-1)}\mathbf{x}\|_2^2 \quad \text{subject to } \|\mathbf{x}\|_0 \le k_0.$$

obtaining sparse representations $\hat{\mathbf{x}}_i$ for $1 \le i \le M$. These form the matrix $\mathbf{X}_{(k)}$.

- **MOD Dictionary Update Stage:** Update the dictionary by the formula

$$\mathbf{A}_{(k)} = arg\min_{\mathbf{A}} \|\mathbf{Y} - \mathbf{A}\mathbf{X}_{(k)}\|_F^2 = \mathbf{Y}\mathbf{X}_{(k)}^T\left(\mathbf{X}_{(k)}\mathbf{X}_{(k)}^T\right)^{-1}.$$

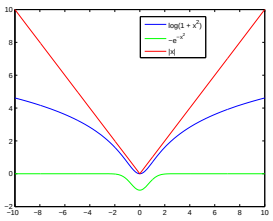
- **Stopping Rule:** If the change in $\|\mathbf{Y} - \mathbf{A}_{(k)}\mathbf{X}_{(k)}\|_F^2$ is small enough, stop. Otherwise, apply another iteration.

Output: The desired result is $\mathbf{A}_{(k)}$.

MOD method
[Engan et al.]

How to learn A or a_{ij}? (Olshausen & Field paper)

- E = -(preserve information) - λ(sparseness of x_i)
- (preserve information) $= -\|Y - AX\|_F^2$.
- (sparseness of x_i) = $-\sum_j S\left(\frac{x_{ij}}{\sigma}\right)$.
- Possible S(x) could be $-e^{-x^2}$, $\log(1 + x^2)$, $|x|$.
- They are all unimodal and peaked around 0.
- Minimize E.
- $min_{A,X} \quad \|Y - AX\|_F^2 + \lambda \sum_{ij} \log(1 + x_{ij}^2)$

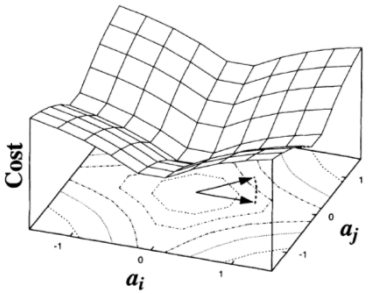


How to learn A or a_{ij}? (Olshausen & Field paper)

- E = -(preserve information) - λ(sparseness of x_i)

(preserve information) $= -\|Y - AX\|_F^2$.

(sparseness of x_i) = $-\sum_j S\left(\frac{x_{ij}}{\sigma}\right)$.



$\log(1 + x^2)$

Algorithm

$$Y = AX \longrightarrow I = \phi A$$

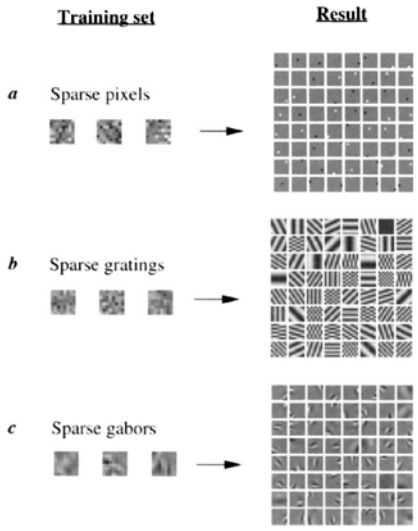
$$\dot{a}_i = b_i - \sum_j C_{ij} a_j - \frac{\lambda}{\sigma} S'\left(\frac{a_i}{\sigma}\right)$$

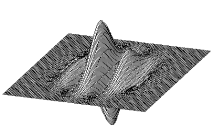
where $b_i = \Sigma_{x,y} \phi_i(x,y) I(x,y)$ and $C_{ij} = \Sigma_{x,y} \phi_i(x,y) \phi_j(x,y)$.

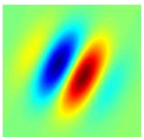
$$\Delta\phi_i(x_m, y_n) = \eta \left\langle a_i \left[I(x_m, y_n) - \hat{I}(x_m, y_n) \right] \right\rangle \qquad (6)$$

where $\hat{I}$ is the reconstructed image, $\hat{I}\ (x_m, y_n) = \Sigma_i a_i \phi_i(x_m, y_n)$, and η is the learning rate.

Experiments (Artificial data)

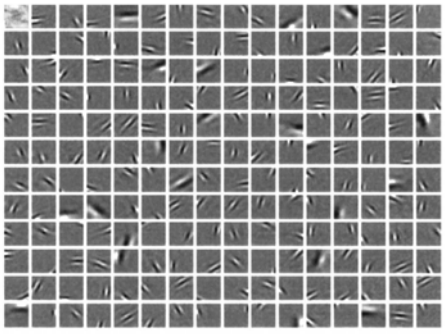






Dictionary learning algorithm worked for artificial data.

Experiments (Natural Images)



Ten 512x512 natural images in the American northwest

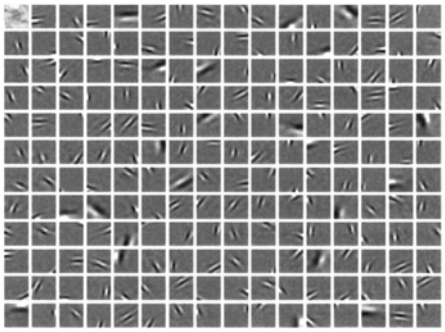
16x16 image patches

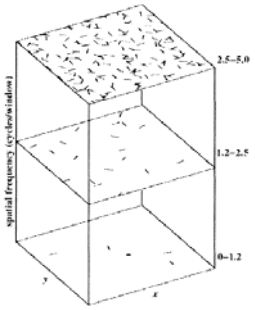
$S(x) = \log(1 + x^2)$.

192 basis functions

Spatially localized, oriented, band-pass (selective to structure at different spatial scales)

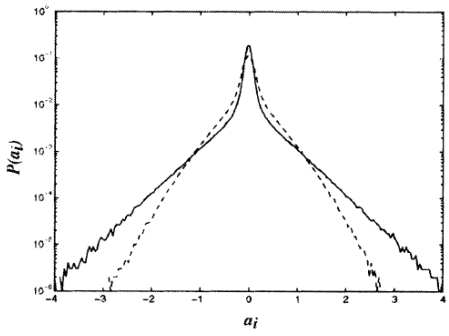
Experiments (Natural Images)





Spatially localized, oriented, band-pass (selective to structure at different spatial scales)

Experiments (Natural Images)



Solid line – After learning dictionary
Broken line – Random initial conditions

Ten 512x512 natural images in the American northwest

16x16 image patches

$S(x) = \log(1 + x^2)$.

192 basis functions

Decreased entropy, increased independence.

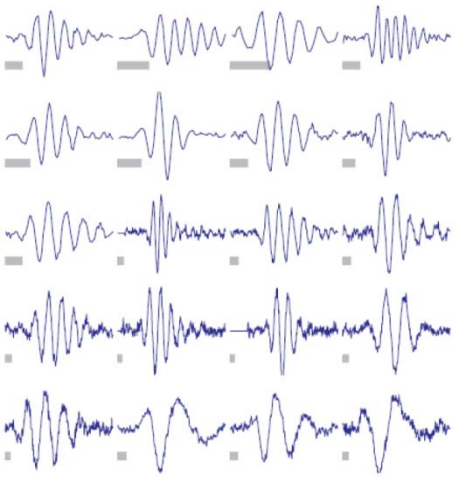
Conclusion

- Dictionary learning problem although it is hard, can be solved iteratively by first learning the sparse activations and updating the dictionary and so on.
- Sparsification as a preprocessing step is desirable since it has benefits of feature detection, memory storage, and independence etc.
- Sparsification for natural images results in spatially localized, oriented, and bandpass filters (basis functions), just like V1.

Note: Discovery of sensory filters by dictionary learning are not limited to the visual inputs but also auditory.

Sparse Coding for audio?

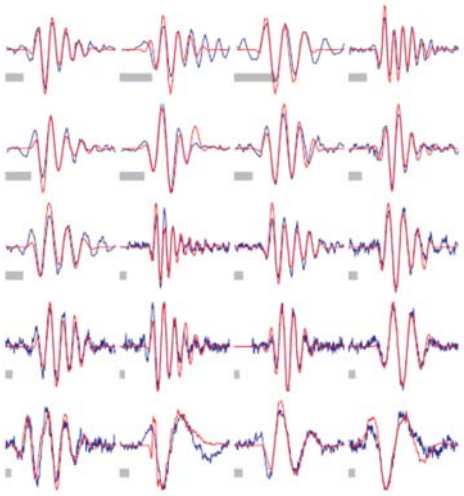
Cat auditory nerve fibres



Evan Smith & Lewicki, Nature 2006

Sparse Coding for audio?

Learned basis functions



Evan Smith & Lewicki, Nature 2006

Questions?