(Unsupervised) Feature Learning +
(Supervised) Machine Learning
(Self Taught) Deep Learning

Lecture 06

Razvan C. Bunescu
School of Electrical Engineering and Computer Science
bunescu@ohio.edu
Sparse Coding

• A **sparse coder** learns:
  1. An **over-complete** set of **basis vectors** $A = [A_1, ..., A_K]$;
  2. **Sparse features** $s = [s_1, ..., s_K]$;
  
  such that:

\[
x = As = \sum_{k=1}^{K} s_k A_k
\]

• Over-complete basis vectors => bases $A_k$ cannot be linearly independent.

• Can we learn sparse features such that the basis vectors are linearly independent?
Independent Component Analysis (ICA)

• Would like a **sparse model** that learns:
  1. An **under-complete** set of basis vectors $A = [A_1, ..., A_K]$;
  2. Sparse features $s = [s_1, ..., s_K]$;

such that:

  • Bases $A_k$ are **linearly independent**.
  • Data is **decoded** (reconstructed) as:

$$x = As = \sum_{k=1}^{K} s_k A_k$$
Independent Component Analysis (ICA)

- Consider the complete ICA case, $K = N$:
  \[ x = [x_1, \ldots, x_N] \]
  \[ s = [s_1, \ldots, s_N] \]
- Then $x = As$, where $A$ is an $N \times N$ square matrix.
- But bases $A_k$ are linearly independent $\Rightarrow$ $A$ is non-singular.
- Then original decoding is equivalent with the encoding:
  \[ x = As \iff s = A^{-1}x = Wx \]
- $A^{-1} = W = [W_1, \ldots, W_K]$ is non-singular $\Rightarrow$ basis vectors $W_k$ are linearly independent.
Independent Component Analysis (ICA)

- **Encoding** formulation of ICA learns:
  1. An under-complete set of basis vectors $W = [W_1, ..., W_K]$;
  2. Sparse features $s = [s_1, ..., s_K]$;
  
  such that:
  - Bases $W_k$ are linearly independent.
  - Data is encoded as $s = Wx$.

- What if we want a **sparse coder** to learn **orthonormal** bases?
- Similar derivation applies $\Rightarrow$ **orthonormal ICA**.
Orthonormal ICA

- **Orthonormal ICA** learns:
  1. An under-complete set of basis vectors \( W = [W_1, ..., W_K] \);
  2. Sparse features \( s = [s_1, ..., s_K] \);

  such that:
  - Bases \( W_k \) are **orthonormal**.
  - Data is **encoded** as \( s = Wx \).

- **Orthonormal bases** \( \Rightarrow WW^T = I \).
- **Sparse features** \( s = Wx \) \( \Rightarrow \) minimize \( \|Wx\|_1 \).
Orthonormal ICA

• Optimization formulation of **orthonormal ICA**: 

\[
\begin{align*}
\text{minimize:} & \quad \|Wx\|_1 \\
\text{subject to:} & \quad WW^T = I
\end{align*}
\]

• “Independent” Component Analysis because:

1. Encoding basis are linearly independent (orthonormal)?
   i.e. [Independent Component Analysis].

2. Components (sources s) are independent (orthonormal)?
   i.e. [Independent Component Analysis].
Orthonormal ICA

- Would like sparse features $s = [s_1, \ldots, s_K]$ that are statistically independent:
  
  1. **Quasi-Independence:**
     - Compute $W$ such that features $s_k$ have properties of independent random variables.
  
  2. **True-Independence:**
     - Compute $W$ such that features $s_k$ are random variables.

Independent variables are *uncorrelated*
Orthonormal ICA

• Quasi-Independent components through sample covariance:
  - Let \( X = [x^{(1)}, ..., x^{(m)}] \) and \( S = WX = [s^{(1)}, ..., s^{(m)}] \)

\[
\Sigma_S = \frac{1}{m} SS^T = \frac{1}{m} (WX)(WX)^T = \frac{1}{m} WXX^TW^T
\]

\[
= W \left( \frac{1}{m} XX^T \right) W^T = W \Sigma_x W^T
\]

\[
\Sigma_S = I \Rightarrow W \Sigma_x W^T = I
\]

\[
\Sigma_x = I
\]

\[
WW^T = I
\]

X must be whitened!
Orthonormal ICA

• Sparse uncorrelated feature learning by **orthonormal ICA**:
  1. ZCA whitening of $X$.
  2. Solve optimization problem:

$$\text{minimize: } \|WX\|_1$$
$$\text{subject to: } WW^T = I$$

---
Orthonormal ICA

1. ZCA whitening of $X$.

2. Solve optimization problem:
   
   \[
   \text{minimize: } \|WX\|_1 \\
   \text{subject to: } WW^T = I
   \]

• Repeat until convergence (gradient descent):
  
  I. $W \leftarrow W - \alpha \nabla_w \|W\|_1$
  
  II. $W \leftarrow f(W)$, such that $WW^T = I$
      
      • in practice $f(W) = (WW^T)^{-1/2}W$

Lecture 6
Reconstruction ICA (RICA)

- Replace ICA’s orthonormality constraint with a soft reconstruction penalty term:

\[ W = \arg \min_W J(W) = \lambda \|W\|_1 + \frac{1}{2} \|W^T WX - X\|_2^2 \]

- \( W \) strives for non-redundant features

- When under-complete features and \( WW^T = I \) \( \Rightarrow \) ICA.
- When under-complete features and \( \lambda = \infty \) \( \Rightarrow \) ICA.
RICA vs. Sparse AE

- **Sparse AutoEncoder:**
  - with L1 sparsity and tied weights.

  \[ W = \arg \min_w \lambda \| \sigma(WX) \|_1 + \frac{1}{2} \| \sigma(W^T \sigma(WX)) - X \|_2^2 \]

- **RICA = Sparse **Linear** AutoEncoder:**

  \[ W = \arg \min_w \lambda \| WX \|_1 + \frac{1}{2} \| W^T WX - X \|_2^2 \]

- RICA (like sAE) can learn with **over-complete** features.
- RICA (like sAE) is more robust to non-whitened data.
Independent Component Analysis

- Would like sparse features \( s = [s_1, ..., s_K] \) that are statistically independent:

**Derive ICA by starting from assumption of independence.**

- Compute \( W \) such that features \( s_k \) have properties of independent random variables.

2. **True-Independene:**

- Compute \( W \) such that features \( s_k \) are random variables.
ICA: The Cocktail Party Problem

Read Andrew Ng’s derivation of ICA.