EECS 275 Matrix Computation

Ming-Hsuan Yang

Electrical Engineering and Computer Science
University of California at Merced
Merced, CA 95344
http://faculty.ucmerced.edu/mhyang

Lecture 25

Overview

- Sparse coding
- Overcomplete dictionary
- Matching pursuit
- Basis pursuit
- K-SVD
- Applications

Main idea

- Sparse representation of signals
- Learning an overcomplete dictionary that contains prototypes or signal-atoms
- Signals are described by sparse linear combination of these atoms
- Given dictionary, how to find sparse representation?
- Given data, how to find dictionary?
- K-SVD: An iterative method that alternates between
    - sparse coding of the examples based on the current dictionary, and
    - a process of updating the dictionary atoms to better fit the data

Sparse representation of signals

- Using an overcomplete dictionary matrix $D \in \mathbb{R}^{n \times K}$ that contains K prototype signal-atoms for columns $\{\mathbf{d}_j\}_{j=1}^{K}$, a signal $\mathbf{y} \in \mathbb{R}^n$ can be represented as a sparse linear combination of these atoms

$$\mathbf{y} = D\mathbf{x}, \text{ or } \mathbf{y} \approx D\mathbf{x} \;\text{ subject to }\; \|\mathbf{y} - D\mathbf{x}\|_p \leq \varepsilon$$

  where the vector $\mathbf{x} \in \mathbb{R}^K$ contains the representation coefficients of the signal $\mathbf{y}$,and ℓ_p-norm for $p = 1, 2$, and ∞ are often used
- If $n < K$ and D is a full-rank matrix, an infinite number of solutions are available for the representation problems, hence constraints on the solution must be set
- The sparsest representation is the solution of either

$$(P_0) \;\; \min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ subject to } \mathbf{y} = D\mathbf{x} \qquad (1)$$

$$(P_{0,\varepsilon}) \;\; \min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ subject to } \|\mathbf{y} - D\mathbf{x}\|_2 \leq \varepsilon \qquad (2)$$

  where $\|\cdot\|_0$ is the ℓ_0-norm, counting the nonzero entries of a vector

The choice of the dictionary

- Can either be chosen as a prespecified set of function (i.e., non-adaptive) or designed by adapting its content to fit a given set of signal examples
- Prespecified transform matrix: wavelets, curvelets, contourlets, steerable wavelet filters, short-time Fourier transforms, random matrices, and more
- K-SVD: learn a dictionary D from training examples
- Compressive sensing: use random matrices

Sparse coding

- Sparse coding: Computing the representation coefficients $\mathbf{x}$ based on the given signal $\mathbf{y}$ and the dictionary D
- Commonly referred as as atom decomposition and requires formulation of (1) or (2)
- Exact determination of sparest representation proves to be an NP-hard problem
- Typically done by a "pursuit algorithm" that finds an approximate solution
  - matching pursuit (MP) and orthogonal matching pursuit (OMP) algorithms: require inner products between signals and dictionary columns
  - basis pursuit (BP) algorithms: a convexification of the problems in (1) or (2) by replacing the ℓ_0-norm with an ℓ_1-norm with iterative methods
  - The focal underdetermined system solver (FOCUSS) is very similar using the ℓ_p-norm with $p \leq 1$ although the overall problem becomes non-convex
  - BP and FOCUSS algorithms can also be motivated based on maximum a posteriori (MAP) estimation

Matching pursuit

- Greedy algorithm that finds best matching projection of multidimensional data onto an overcomplete dictionary D
- Each such dictionary D is a collection of waveforms $(\phi_\gamma)_{\gamma \in \Gamma}$ with γ a parameter

$$\mathbf{y} = \sum_{\gamma \in \Gamma} \alpha_\gamma \phi_\gamma, \text{ or } \quad \mathbf{y} = \sum_{i=1}^{m} \alpha_{\gamma_i} \phi_{\gamma_i} + R^{(m)}$$

as an approximate decomposition with residual $R^{(m)}$
- Start with an initial approximation $\mathbf{y}^{(0)} = 0$ and residual $R^{(0)} = \mathbf{y}$, build up a sequence of sparse approximations stepwise
- At step k, identify the atom that best correlates with the residual (by sweeping all samples), and then add to the current approximation a scalar multiple of that atom, so that $\mathbf{y}^{(k)} = \mathbf{y}^{(k-1)} + \alpha_k \phi_{\gamma k}$ where $\alpha_k = \langle R^{(k-1)}, \phi_{\gamma_k} \rangle$ and $R^{(k)} = \mathbf{y} - \mathbf{y}^{(k)}$
- After m steps, obtain the representation in (7) with residual $R = R^{(m)}$

Orthogonal matching pursuit

- When the dictionary is orthogonal (e.g., orthogonal wavelet), MP recovers the underlying sparse structure well
- Computational complexity of MP for encoder is high
- Improvements include the use of approximate dictionary representations and suboptimal ways of choosing the best match at each iteration (atom extraction)
- Orthogonal matching pursuit (OMP): an extra step of orthogonalization in MP
- Take all m terms that have entered at step m and solve the least squares problem

$$\min_{(\alpha_i)} \|\mathbf{y} - \sum_{i=1}^{m} \alpha_i \phi_{\gamma_i}\|_2$$

  for coefficients $(\alpha_i^{(m)})$
- Then forms the residual $\overline{R}^{[m]} = \mathbf{y} - \sum_{i=1}^{m} \alpha_i^{(m)} \phi_{\gamma_i}$ which will be orthogonal to all terms currently in the model

Basis pursuit

- Matching pursuit can be viewed as a greedy approximation to solve

$$\min \|\boldsymbol{\alpha}\|_0 \quad \text{subject to} \quad \Phi\boldsymbol{\alpha} = \mathbf{y}$$

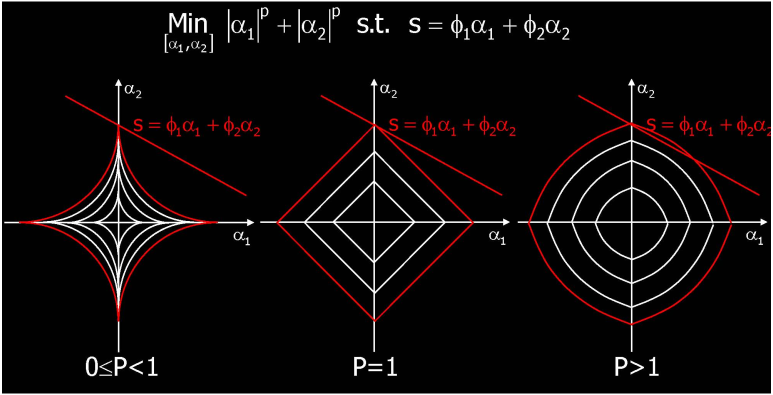
- Basis pursuit: A principle for decomposing a signals into an optimal superposition of dictionary elements
- Approximate sparsity with ℓ_1-norm
- Optimal in the sense of having smallest ℓ_1-norm among all such decompositions

$$\min \|\boldsymbol{\alpha}\|_1 \quad \text{subject to} \quad \Phi\boldsymbol{\alpha} = \mathbf{y}$$

- A convex optimization problem that can be solved via linear programming

Why ℓ_1-norm?

- Consider a two-dimensional case



Design of dictionaries

- There is an intriguing relation between sparse representation and clustering (i.e., vector quantization)
- In clustering, a set of descriptive vectors $\{\mathbf{d}_k\}_{k=1}^K$ is learned, and each sample is represented by one of these vectors (based on distance metric e.g., ℓ_2-norm)
- Can think of this as an extreme sparse representation, where only one atom is allowed in the signal decomposition
- K-means algorithm, also known as the generalized Lloyd (GLA) algorithm, is the most commonly used procedure for clustering
- Dictionary learning can be considered as generalization of K-means algorithm:
  - given $\{\mathbf{d}_k\}_{k=1}^K$, assign the training examples to their nearest neighbor
  - given that assignment, update $\{\mathbf{d}_k\}_{k=1}^K$ to better fit the examples

Maximum likelihood methods

- Formulate the problem with Gaussian distributions

$$\mathbf{y} = D\mathbf{x} + \mathbf{v}$$

where $\mathbf{v}$ are white Gaussian white noise, and

$$p(Y|D) = \prod_{i=1}^{N} p(\mathbf{y}_i|D)$$

, and consider $\mathbf{x}$ as the hidden variables

$$\begin{aligned} p(\mathbf{y}_i|D) &= \int p(\mathbf{y}_i, \mathbf{x}|D)d\mathbf{x} = \int p(\mathbf{y}_i|\mathbf{x}, D)p(\mathbf{x})d\mathbf{x} \\ &= C \int \exp(\tfrac{1}{2\sigma^2}\|D\mathbf{x} - \mathbf{y}_i\|^2)p(\mathbf{x})d\mathbf{x} \end{aligned}$$

where C is a constant

- The prior distribution is assumed to be zero-mean with Cauchy or Laplace distribution

Maximum likelihood methods (cont'd)

- Assuming the prior is with Laplace distribution
$$\begin{aligned} p(\mathbf{y}_i|D) &= \int p(\mathbf{y}_i|\mathbf{x}, D)p(\mathbf{x})d\mathbf{x} \\ &= C\int \exp(\tfrac{1}{2\sigma^2}\|D\mathbf{x} - \mathbf{y}_i\|^2)\exp(\lambda\|\mathbf{x}\|_1)d\mathbf{x} \end{aligned}$$
- Difficult to evaluate but can be simplified with
$$\begin{aligned} D &= \underset{D}{\operatorname{argmax}} \textstyle\sum_{i=1}^N \max_{\mathbf{x}_i} p(\mathbf{y}_i, \mathbf{x}_i|D) \\ &= \underset{D}{\operatorname{argmin}} \textstyle\sum_{i=1}^N \min_{\mathbf{x}_i} \|D\mathbf{x}_i - \mathbf{y}_i\|^2 + \lambda\|\mathbf{x}_i\|_1 \end{aligned} \tag{3}$$
- This problem does not penalize the entries of D as it does for of $\mathbf{x}_i$, thereby the solution tends to increase the dictionary entries
- An iterative method was suggested: first calculate the coefficients $\mathbf{x}_i$ using a simple gradient descent procedure and then update the dictionary using
$$D^{(n+1)} = D^{(n)} - \eta\sum_{i=1}^{N}(D^{(n)}\mathbf{x}_i - \mathbf{y}_i)\mathbf{x}_i^\top$$
- Related to independent component analysis (ICA) which maximizes the mutual information between inputs (samples) and outputs (coefficients)

Method of optimal directions (MOD)

- Follow closely the K-means outline with a sparse coding stage that uses either OMP or FOCUSS followed by an update of the dictionary
- Assume that the sparse coding for each example is known, we define the errors $\mathbf{e}_i = \mathbf{y}_i - D\mathbf{x}_i$, the overall representation error is

$$\|E\|_F^2 = \|[\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_N]\|_F^2 = \|Y - DX\|_F^2$$

- Assume X is fixed, we can seek an update to D such that the above error is minimized by taking derivative of the above equation w.r.t. D, $(Y - DX)X^\top = 0$, and have

$$D^{(n+1)} = YX^{(n)^\top}(X^{(n)}X^{(n)^\top})^{-1}$$

- Related to the maximum likelihood methods

K-means algorithm for vector quantization

- A codebook that includes K codewords (representatives, prototypes) is used to represent a family of vectors (signals) $Y = \{\mathbf{y}_i\}_{i=1}^N$ ($N \gg K$) by nearest neighbor assignment
- Efficient compression or description of signals as clusters
- The dictionary of VQ codewords, $C = [\mathbf{c}_1, \ldots, \mathbf{c}_K]$ is typically trained using the K-means algorithm
- When C is given, each signal is represented as its closest codeword (using ℓ_2 norm), i.e., $\mathbf{y}_i = C\mathbf{x}_i$ where $\mathbf{x}_i = \mathbf{e}_j$ is a canonical vector (trivial basis) with all zero entries except a one in the j-th position

$$\forall k \neq j \quad \|\mathbf{y}_i - C\mathbf{e}_j\|_2^2 \leq \|\mathbf{y}_i - C\mathbf{e}_k\|_2^2$$

- The mean square error is $r_i^2 = \|\mathbf{y}_i - C\mathbf{x}_i\|_2^2$, and the overall MSE is $E = \sum_{i=1}^K r_i^2 = \|Y - CX\|_2^2$
- The VQ training process is to find a codebook C that minimizes E subject to X

$$\min_{C,X} \|Y - CX\|_F^2 \quad \text{subject to} \quad \forall i \;\; \mathbf{x}_i = \mathbf{e}_k \text{ for some } k \tag{4}$$

K-SVD: Generalizing the K-means

- The sparse representation problem can be viewed as a generalization of the VQ problem (4) in which we allow each input signal to be represented by a linear combination

$$\min_{D,X} \|Y - DX\|_F^2 \ \text{ subject to } \ \forall i \ \|\mathbf{x}_i\|_0 \leq T_0 \tag{5}$$

, or

$$\min_{D,X} \|Y - DX\|_F^2 \ \text{ subject to } \ \|Y - DX\|_F^2 \leq \varepsilon \tag{6}$$

- Minimize (5) iteratively by first fix D and find the coefficient matrix X using any pursuit method, and then search for a better dictionary
- It update one column at a time, fixing all the other columns, and find a new column d_k and new values for its coefficients that best reduce the MSE
- The process of updating only one column of D at a time is a problem having a straightforward solution based on SVD

Updating dictionary

- Assume that both X and D are fixed, and want to add on column in the dictionary $\mathbf{d}_k$ and the coefficients of k-th row of X is $\mathbf{x}_T^k$ (different from the vector $\mathbf{x}_k$ which is the k-th column in X)
- The objective function can be rewritten as

$$
\begin{aligned}
\|Y - DX\|_F^2 &= \left\|Y - D_{j=1}^{K} \mathbf{d}_j \mathbf{x}_T^j\right\|_F^2 \\
&= \left\|(Y - \textstyle\sum_{j \neq k} \mathbf{d}_j \mathbf{x}_T^j) - \mathbf{d}_k \mathbf{x}_T^k\right\|_F^2 \\
&= \left\|E_k - \mathbf{d}_k \mathbf{x}_T^k\right\|_F^2
\end{aligned}
$$

- Decompose DX to the sum of K rank-1 matrices where $K-1$ terms are fixed and the k-th term remains in question
- It would be tempting to suggest the use of SVD to find alternative $\mathbf{d}_k$ and $\mathbf{x}_T^k$
- The SVD finds the closest rank-1 matrix that approximate E_k
- However, this minimization does not take sparsity into consideration

Updating dictionary (cont'd)

- One remedy to enforce sparsity is to favor the dictionary atoms that have been used frequently
- Define $\boldsymbol{\omega}_k$ as the group of indices pointing to examples $\{\mathbf{y}_i\}$ that use atom $\mathbf{d}_k$, i.e., those where $\mathbf{x}_T^k(i)$ is nonzero
$$\boldsymbol{\omega}_k = \{i | 1 \leq i \leq K,\ \mathbf{x}_T^k(i) \neq 0\}$$
- Define Ω_k as a matrix of size $N \times |\boldsymbol{\omega}_k|$ with ones on the $(\boldsymbol{\omega}_k(i), i)$-th entries and zeros elsewhere
- When multiplying $\mathbf{x}_R^k = \mathbf{x}_T^k \Omega_k$, this shrinks the row vector $\mathbf{x}_T^k$ by discarding of the zero entries, resulting with the row vector $\mathbf{x}_R^k$ of length $|\boldsymbol{\omega}_k|$
- Similarly, $Y_k^R = Y\Omega_k$ creates a matrix of size $n \times |\boldsymbol{\omega}_k|$ that includes a subset of examples that are currently using the $\mathbf{d}_k$ atom
- Same for $E_k^R = E_k \Omega_k$, implying a selection of error columns that correspond to examples that use the atom $\mathbf{d}_k$
- The equivalent minimization
$$\|E_k \Omega_k - \mathbf{d}_k \mathbf{x}_T^k \Omega_k\|_F^2 = \|E_k^R - \mathbf{d}_k \mathbf{x}_R^k\|_F^2$$
which can now be solved by SVD

Updating dictionary (cont'd)

- Taking the restricted matrix E_k^R, SVD decomposes it to $E_k^R = U\Sigma V^\top$
- Define the solution for $\widetilde{\mathbf{d}_k}$ as the first column of U, and the coefficient vector $\mathbf{x}_R^k$ as the fist column of V multiplied by σ_1
- In the K-SVD algorithm, one needs to sweep through the columns and use always the most updated coefficients as they emerge from the SVD steps

The K-SVD algorithm

Initialize: Normalize columns of the dictionary matrix $D^{(0)} \in \mathbb{R}^{n \times K}$

for $J = 1, 2, \ldots$ **do**

Sparse coding: Use any pursuit algorithm to compute the representation vector $\mathbf{x}_i$ for each example $\mathbf{y}_i$, by approximating the solution of

$$i = 1, \ldots, N, \quad \min_{\mathbf{x}_i} \|\mathbf{y}_i - D\mathbf{x}\|_2^2 \quad \text{subject to} \quad \|\mathbf{x}_i\|_0 \leq T_0$$

Codebook update: For each column $k = 1, \ldots, K$ in $D^{(J-1)}$

- Define the group of examples that use this atom, $\boldsymbol{\omega}_k = \{i | 1 \leq i \leq N, \ \mathbf{x}_T^k(i) \neq 0\}$
- Compute the overall representation error $E_k = Y - \sum_{j \neq k} \mathbf{d}_j \mathbf{x}_T^j$
- Restrict E_k by choosing only the columns corresponding to $\boldsymbol{\omega}_k$ and obtain E_k^R
- Apply SVD decomposition $E_k^R = U \Sigma V^\top$. Choose the updated dictionary column $\widetilde{\mathbf{d}}_k$ to be the first column of U. Update the coefficient vector $\mathbf{x}_R^k$ to be the first column of V multiplied by σ_1

end for