Sparse Methods

Sanjiv Kumar, Google Research, NY
EECS-6898, Columbia University - Fall, 2010
What is Sparsity?

When a data item such as a vector or a matrix has only a few nonzero entries

- Sparsity in features $x$
- Sparsity in parameters $\alpha$
What is Sparsity?

When a data item such as a vector or a matrix has only a few nonzero entries

- Sparsity in features $x$
- Sparsity in parameters $\alpha$

Learning methods that utilize sparsity

- Kernel methods $\rightarrow$ want only a few coefficients nonzero in kernel sums
- Feature selection in linear methods using $L_1$ penalty e.g., Lasso
- Group feature selection e.g., Group Lasso
- Sparse Coding $\rightarrow$ Linear reconstruction of the data using a few elements from large dictionaries
- Sparse PCA and Sparse SVD
  - Find sparse vectors “closest” to the true singular vectors
- Many others…

Sparse methods lead to parsimonious (and interpretable) models in addition to being efficient for large scale learning
Sparse penalties

Total Loss \[ J(\alpha) = \sum_{i=1}^{n} l(x_i, y_i, \alpha) + R(\alpha) \]

Lasso or L₁
\[ R(\alpha) = \lambda \|\alpha\|_1 \] leads to sparse \( \alpha \)

Elastic net
\[ R(\alpha) = \lambda_1 \|\alpha\|_1 + \lambda_2 \|\alpha\|_2^2 \] allows important correlated variables to be picked

Group Lasso
\[ R(\alpha) = \lambda \sum_{l=1}^{L} \|\alpha_l\|_2 \] note: no square in the norm
allows groups of variables to be sparse

Sparse Group Lasso
\[ R(\alpha) = \lambda_1 \sum_{l=1}^{L} \|\alpha_l\|_2 + \lambda_2 \|\alpha\|_1 \] allows elements within groups to be sparse in addition to sparse groups

Multitask Lasso
\[ R(\alpha) = \lambda \sum_{l=1}^{L} \|\alpha_l\|_\infty \] encourages entire groups to have zero elements
more aggressive than group lasso
Lasso

Linear regression with L₁ penalty

$$\hat{\alpha} = \arg\min_{\alpha} \sum_{i=1}^{n} (y_i - \alpha^T x_i)^2 + \lambda \|\alpha\|_1 \quad x, \alpha \in \mathbb{R}^d$$

equivalent problem

$$\hat{\alpha} = \arg\min_{\alpha} \sum_{i=1}^{n} (y_i - \alpha^T x_i)^2 \text{ s.t. } \|\alpha\|_1 \leq C$$
Lasso

Linear regression with $L_1$ penalty

\[
\hat{\alpha} = \arg\min_{\alpha} \sum_{i=1}^{n} (y_i - \alpha^T x_i)^2 + \lambda \|\alpha\|_1 \quad x, \alpha \in \mathbb{R}^d
\]

equivalent problem
\[
\hat{\alpha} = \arg\min_{\alpha} \sum_{i=1}^{n} (y_i - \alpha^T x_i)^2 \text{ s.t. } \|\alpha\|_1 \leq C
\]

- Quadratic objective function with linear constraints $\rightarrow$ Quadratic Programming!
  
  • Does not scale well with problem size

- Can be solved using iterative (sub)gradient methods e.g., coordinate descent

- Leads to sparse $\alpha$, higher $C \rightarrow$ less sparse $\alpha$

- Related method: Forward Stagewise Procedure (Homotopy method)
Lasso

Linear regression with L₁ penalty

\[ \hat{\alpha} = \arg \min_\alpha \sum_{i=1}^{n} (y_i - \alpha^T x_i)^2 + \lambda \|\alpha\|_1 \quad x, \alpha \in \mathbb{R}^d \]

equivalent problem \[ \hat{\alpha} = \arg \min_\alpha \sum_{i=1}^{n} (y_i - \alpha^T x_i)^2 \text{ s.t. } \|\alpha\|_1 \leq C \]

- Quadratic objective function with linear constraints \( \rightarrow \) Quadratic Programming !
  - Does not scale well with problem size
- Can be solved using iterative (sub)gradient methods e.g., coordinate descent
- Leads to sparse \( \alpha \), higher \( C \) \( \rightarrow \) less sparse \( \alpha \)
- Related method: Forward Stagewise Procedure (Homotopy method)
  - Suppose data \( X \) and prediction \( y \) are centered, and \( \mu_t = X^T \alpha_t \)
  - Given current estimate of \( \alpha_t \) (starting with \( \alpha_0 = 0 \)), compute residual \( r_t = (y - \mu_t) \)
  - Compute data correlation with the residual vector \( c_t = X r_t \)
  - Find the direction of greatest correlation and take a small step

\[ j = \arg \max_l (c_{lt}) \Rightarrow \mu_{t+1} \leftarrow \mu_t + \epsilon \cdot \text{sgn}(c_{jt}) \cdot (X^j)^T \]

\( j \text{th row of } X \)
Group Lasso

Linear regression with $L_2$ (non-squared) penalty

- Suppose the features can be divided into $L$ groups
- Let $X_l$ be the data submatrix and $\alpha_l$ be the parameter chunk corresponding to $l^{th}$ group

$$\hat{\alpha} = \arg \min_{\alpha} \left\| y - \sum_{l=1}^{L} X_l^T \alpha_l \right\|_2^2 + \lambda \sum_{l=1}^{L} \| \alpha_l \|_2$$
Group Lasso

Linear regression with $L_2$ (non-squared) penalty

- Suppose the features can be divided into $L$ groups
- Let $X_l$ be the data submatrix and $\alpha_l$ be the parameter chunk corresponding to $l^{th}$ group

$$\hat{\alpha} = \arg\min_{\alpha} \|y - \sum_{l=1}^{L} X_l^T \alpha_l \|_2^2 + \lambda \sum_{l=1}^{L} \|\alpha_l\|_2$$

Subgradient equation

$$-X_l (y - \sum_{l=1}^{L} X_l^T \alpha_l) + \lambda s_l = 0 \quad l = 1, \ldots, L$$

$$s_l = \alpha_l / \|\alpha_l\| \quad \text{if } \alpha_l \neq 0, \text{any vector with } \|s_l\|_2 < 1 \text{ otherwise}$$
**Group Lasso**

Linear regression with $L_2$ (non-squared) penalty

- Suppose the features can be divided into $L$ groups
- Let $X_l$ be the data submatrix and $\alpha_l$ be the parameter chunk corresponding to $l$th group

$$\hat{\alpha} = \arg\min_{\alpha} \left\| y - \sum_{l=1}^{L} X_l^T \alpha_l \right\|^2_2 + \lambda \sum_{l=1}^{L} \| \alpha_l \|_2$$

Subgradient equation

$$-X_l (y - \sum_{l=1}^{L} X_l^T \alpha_l) + \lambda s_l = 0 \quad l = 1, \ldots, L$$

$$s_l = \alpha_l / \| \alpha_l \| \quad \text{if } \alpha_l \neq 0, \text{any vector with } \| s_l \|_2 < 1 \text{ otherwise}$$

if

$$\left\| X_l (y - \sum_{k \neq l} X_k^T \alpha_k) \right\| < \lambda \Rightarrow \alpha_l = 0$$

otherwise

$$\alpha_l = \left( X_l X_l^T + \lambda / \| \hat{\alpha}_l \| \right)^{-1} X_l r_l \quad \text{where} \quad r_l = y - \sum_{k \neq l} X_k^T \alpha_k$$
Group Lasso

Linear regression with L2 (non-squared) penalty
- Suppose the features can be divided into L groups
- Let $X_l$ be the data submatrix and $\alpha_l$ be the parameter chunk corresponding to $l^{th}$ group

$$\hat{\alpha} = \arg \min_{\alpha} \left\| y - \sum_{l=1}^{L} X_l^T \alpha_l \right\|_2^2 + \lambda \sum_{l=1}^{L} \| \alpha_l \|_2$$

Subgradient equation

$$-X_l (y - \sum_{l=1}^{L} X_l^T \alpha_l) + \lambda s_l = 0 \quad l = 1, ..., L$$

$$s_l = \alpha_l / \| \alpha_l \| \quad \text{if } \alpha_l \neq 0, \text{any vector with } \| s_l \|_2 < 1 \text{ otherwise}$$

if

$$\left\| X_l (y - \sum_{k \neq l} X_k^T \alpha_k) \right\| < \lambda \Rightarrow \alpha_l = 0$$

otherwise

$$\alpha_l = \left( X_l X_l^T + \lambda / \| \hat{\alpha}_l \| \right)^{-1} X_l r_l \quad \text{where } r_l = y - \sum_{k \neq l} X_k^T \alpha_k$$

if $X_l X_l^T = I$ and $v_l = X_l r_l$

$$\alpha_l = (1 - \lambda / \| v_l \|) v_l$$

Block-coordinate descent
Multitask Penalty

Want to learn \( K \) parameter vectors, one for each task \( k = 1, \ldots, K \)

\[
\hat{\alpha} = \arg \min_{\alpha} \sum_{i=1}^{n} l(x_i, \alpha) + \lambda \| \alpha \|_{1,\infty} \quad \alpha \in \mathbb{R}^{d \times K} \quad x \in \mathbb{R}^{d}
\]

\[
\| \alpha \|_{1,\infty} = \sum_{j=1}^{d} \max_{k} |\alpha_{jk}| \quad \text{sum of max absolute value in each row of } \alpha
\]
Multitask Penalty

Want to learn $K$ parameter vectors, one for each task $k = 1, \ldots, K$

$$\hat{\alpha} = \arg \min_{\alpha} \sum_{i=1}^{n} l(x_i, \alpha) + \lambda \|\alpha\|_{1,\infty} \quad \alpha \in \mathbb{R}^{d \times K} \quad x \in \mathbb{R}^{d}$$

$$\|\alpha\|_{1,\infty} = \sum_{j=1}^{d} \max_{k} |\alpha_{jk}|$$

sum of max absolute value in each row of $\alpha$

Equivalent

$$\hat{\alpha} = \arg \min_{\alpha} \sum_{i=1}^{n} l(x_i, \alpha) \quad \text{s.t.} \quad \|\alpha\|_{1,\infty} \leq C$$

Tends to make entire row of matrix $\alpha$ zero

Choice of loss function

Regression for $k^{th}$ task

$$l(x_i, y_i, \alpha_k) = \left\| y_i - \alpha_k^T x_i \right\|_2^2$$

Classification for $k^{th}$ task

$$l(x_i, y_i, \alpha_k) = \max(0, 1 - y_i \alpha_k^T x_i)$$

Can be solved directly using block coordinate descent or via projected subgradient
Projected Subgradient Method

Let’s focus on classification case with hinge loss

\[ \alpha_{t+1} = P_{\Omega}(\alpha_t - \eta_t g_t) \]

projection on convex set \( \Omega: \| \alpha \|_{1,\infty} \leq C \)

subgradient of \( \max(0, 1 - y_i \alpha_k^T x_i) \)

step size \( \eta_0 / \sqrt{t} \)
Projected Subgradient Method

Let's focus on classification case with hinge loss

\[ \alpha_{t+1} = P_\Omega(\alpha_t - \eta_t g_t) \]

projection on convex set \( \Omega: \|\alpha\|_{1,\infty} \leq C \) 

subgradient of \( \max(0, 1 - y_i \alpha_k^T x_i) \)

step size \( \eta_0 / \sqrt{t} \)

Subgradient for kth task

\[ g_t^k = \sum_{i:x_i=k, y_i \alpha_k^T x_i < 1} y_i x_i \]

Projection on constraint set

\[ \min_{\alpha \in \Omega} \|\alpha - \alpha'\|^2 = \min_{\alpha \in \Omega} \sum_{jk} (\alpha_{jk} - \alpha'_{jk})^2 \]

How to efficiently do projection of a matrix to a \( L_{1,\infty} \) ball?
Efficient Projection

Equivalent projection problem

- assuming all entries of $\alpha'$ are positive (can be relaxed)

$$\min_{\alpha, \mu} \sum_{jk} (\alpha_{jk} - \alpha'_{jk})^2$$

s.t. \( \forall j, k \quad \alpha_{jk} \leq \mu_j \)

$$\sum_j \mu_j = C$$

$$\forall j \quad \mu_j \geq 0$$
Efficient Projection

Equivalent projection problem

- assuming all entries of $\alpha'$ are positive (can be relaxed)

$$\min_{\alpha, \mu} \sum_{jk} (\alpha_{jk} - \alpha'_{jk})^2$$

s.t. $\forall j, k \quad \alpha_{jk} \leq \mu_j$

$$\sum_j \mu_j = C$$

$\forall j \quad \mu_j \geq 0$

Lagrangian

$$L = \sum_{jk} (\alpha_{jk} - \alpha'_{jk})^2 + \sum_{jk} \rho_{jk} (\alpha_{jk} - \mu)$$

$$+ \theta (\sum_j \mu_j - C) - \sum_{jk} \beta_{jk} \alpha_{jk} - \sum_j \gamma_j \mu_j$$
Efficient Projection

Equivalent projection problem
  – assuming all entries of $\alpha'$ are positive (can be relaxed)

$$\min_{\alpha, \mu} \sum_{j,k} (\alpha_{jk} - \alpha'_{jk})^2$$

s.t. $\forall j, k \quad \alpha_{jk} \leq \mu_j$

$$\sum_j \mu_j = C$$

$\forall j \quad \mu_j \geq 0$

Lagrangian

$$L = \sum_{j,k} (\alpha_{jk} - \alpha'_{jk})^2 + \sum_{j,k} \rho_{jk} (\alpha_{jk} - \mu)$$

$$+ \theta (\sum_j \mu_j - C) - \sum_{j,k} \beta_{jk} \alpha_{jk} - \sum_j \gamma_j \mu_j$$

Suppose $\hat{\mu}$ is the optimal value of $\mu$

$$\alpha'_{jk} \geq \hat{\mu}_j \Rightarrow \alpha_{jk} = \mu_j$$

$$\alpha'_{jk} \leq \hat{\mu}_j \Rightarrow \alpha_{jk} = \alpha'_{jk}$$

$$\hat{\mu}_j = 0 \Rightarrow \alpha_{jk} = 0$$

Optimal value of $\mu$ can be obtained in $O(dK\log dK)$ time!
Synthetic Data, only 10% features relevant for any task prediction

\[ K = 60, \; d = 200 \]

Quattoni et al. [14]
Sparse Coding

**Linear reconstruction of data using a few elements of a dictionary**

- A data vector \( x \in \mathbb{R}^d \) is reconstructed using a dictionary \( D \) containing \( k \) elements (sometimes called “basis” vectors or “atoms”)

\[
x \approx \sum_{j=1}^{k} \alpha^j d_j = DA
\]

\( D \in \mathbb{R}^{d \times k}, \alpha \in \mathbb{R}^k \)

number of nonzero elements \( \|\alpha\|_0 << k \)

\[
x \approx \alpha^1 d_1 + \alpha^2 d_2 + \alpha^3 d_3
\]
Sparse Coding

Linear reconstruction of data using a few elements of a dictionary

- A data vector $x \in \mathbb{R}^d$ is reconstructed using a dictionary $D$ containing $k$ elements (sometimes called “basis” vectors or “atoms”)

$$x \approx \sum_{j=1}^{k} \alpha^j d_j = D\alpha, \quad D \in \mathbb{R}^{d \times k}, \alpha \in \mathbb{R}^k$$

number of nonzero elements $\|\alpha\|_0 \ll k$

Properties

- The dictionary elements usually not orthogonal (unlike PCA)
- Dictionary may be overcomplete, i.e., number of elements are more than the data dimensionality $d \rightarrow$ elements are not linearly independent
- Learned dictionaries usually lead to more more compact representation than predefined ones e.g., based on wavelets
Sparse Coding

Linear reconstruction of data using a few elements of a dictionary
- A data vector $x \in \mathbb{R}^d$ is reconstructed using a dictionary $D$ containing $k$ elements (sometimes called “basis” vectors or “atoms”)

$$x \approx \sum_{j=1}^{k} \alpha^j d_j = D\alpha \quad D \in \mathbb{R}^{d \times k}, \alpha \in \mathbb{R}^k$$

number of nonzero elements $\|\alpha\|_0 << k$

Properties
- The dictionary elements usually not orthogonal (unlike PCA)
- Dictionary may be overcomplete, i.e., number of elements are more than the data dimensionality $d \rightarrow$ elements are not linearly independent
- Learned dictionaries usually lead to more more compact representation than predefined ones e.g., based on wavelets

Learning Problem
- Offline: Given a dataset $X$, learn dictionary $D$
- Online: Given a dictionary $D$ and input vector $x$, learn coefficients $\alpha$
Sparse Coding Formulation

Given a training set $X = [x_1, \ldots, x_n]$, $x_i \in \mathbb{R}^d$ learn dictionary $D$

$$X \approx D\alpha \quad X \in \mathbb{R}^{d \times n}, D \in \mathbb{R}^{d \times k}, \alpha \in \mathbb{R}^{k \times n}$$

Total penalized loss

$$\min_{D, \alpha} \sum_{i=1}^{n} \left[ \frac{1}{2} \| x_i - D\alpha_i \|_2^2 + \lambda \| \alpha_i \|_1 \right] \quad \alpha_i \in \mathbb{R}^k$$
Sparse Coding Formulation

Given a training set $X = [x_1, \ldots, x_n]$, $x_i \in \mathbb{R}^d$ learn dictionary $D$

\[ X \approx D\alpha \quad X \in \mathbb{R}^{d \times n}, D \in \mathbb{R}^{d \times k}, \alpha \in \mathbb{R}^{k \times n} \]

Total penalized loss

\[
\min_{D, \alpha} \sum_{i=1}^{n} \left[ \frac{1}{2} \|x_i - D\alpha_i\|_2^2 + \lambda \|\alpha_i\|_1 \right] \quad \alpha_i \in \mathbb{R}^k
\]

- Given $D$, obtaining best $\alpha$ equivalent to solving Lasso
- Nonconvex in both $D$ and $\alpha$ but convex in one if the other is fixed
- Since scales of $D$ and $\alpha$ are arbitrary, dictionary elements are constrained

\[
C = \{D \in \mathbb{R}^{d \times k}, \text{s.t. } d_j^T d_j \leq 1 \quad \forall \ j = 1, \ldots, k\}
\]
Sparse Coding Formulation

Given a training set $X = [x_1,...,x_n]$, $x_i \in \mathbb{R}^d$ learn dictionary $D$

$$X \approx D\alpha \quad X \in \mathbb{R}^{d \times n}, D \in \mathbb{R}^{d \times k}, \alpha \in \mathbb{R}^{k \times n}$$

Total penalized loss

$$\min_{D,\alpha} \sum_{i=1}^{n} [(1/2)\|x_i - D\alpha_i\|_2^2 + \lambda \|\alpha_i\|_1] \quad \alpha_i \in \mathbb{R}^k$$

- Given $D$, obtaining best $\alpha$ equivalent to solving Lasso
- Nonconvex in both $D$ and $\alpha$ but convex in one if the other is fixed
- Since scales of $D$ and $\alpha$ are arbitrary, dictionary elements are constrained

$$C = \{D \in \mathbb{R}^{d \times k}, \text{s.t.} \ D_j^T d_j \leq 1 \quad \forall \ j = 1,...,k\}$$

- Matrix form

$$\min_{D \in C, \alpha} \left[ (1/2)\|X - D\alpha\|_2^2 + \lambda \|\alpha\|_{1,1} \right] \quad \alpha_{1,1} = \sum_{ij} |a_{ij}|$$

- Usually solved via alternate minimization of $D$ and $\alpha$
  - For large-scale problems (i.e., large $n$), most time spent in getting $\alpha$
Dictionary Learning

Suppose $\alpha$ is given, then how to learn best $D$?

- Due to scale constraints, iterative methods for solving $D$
- For large-scale learning, projected stochastic gradient decent

$$D_t = P_C[D_{t-1} - \eta_t \nabla l(x_t, D_{t-1})]$$

projection on constraint set

if $\|d_j\|_2 > 1$ then $d_j \leftarrow d_j / \|d_j\|_2$

step size

$a/(t+b)$

random sample from training set, can be extended to mini-batch

$$l(x_t, D_{t-1}) = (1/2)\|x_t - D_{t-1}\alpha\|_2^2$$
Suppose $\alpha$ is given, then how to learn best $D$?

- Due to scale constraints, iterative methods for solving $D$
- For large-scale learning, **projected stochastic gradient decent**

$$D_t = P_C[D_{t-1} - \eta_t \nabla l(x_t, D_{t-1})]$$

$l(x_t, D_{t-1}) = (1/2)\|x_t - D_{t-1}\alpha\|_2^2$

- Need to **tune** the step size
- May have **slow convergence**
- Can one **exploit the structure** of the sparse coding problem to get faster learning without much dependence on step size
- Alternative: use **second order** information in stochastic updates
Online Dictionary Learning

Key Idea: The loss function at time $t$ aggregates information from samples drawn up to time $t$

$$ l_t(D) = \frac{1}{2t} \sum_{i=1}^{t} \| x_i - D\alpha_i \|_2^2 $$

best $\alpha$ for point $x_i$ using the dictionary $D_{i-1}$

averaging over previous sparse reconstructions
Online Dictionary Learning

Key Idea: The loss function at time $t$ aggregates information from samples drawn up to time $t$

$$l_t(D) = (1/2t)\sum_{i=1}^{t} \left\| x_i - D\alpha_i \right\|^2_2$$

averaging over previous sparse reconstructions

$$= (1/2t)(\text{Tr}[D^TDA_t] - \text{Tr}[D^TB_t])$$

$$A_t = \sum_{i=1}^{t} \alpha_i\alpha_i^T \quad B_t = \sum_{i=1}^{t} x_i\alpha_i^T$$

best $\alpha$ for point $x_i$ using the dictionary $D_{i-1}$
Online Dictionary Learning

Key Idea: The loss function at time $t$ aggregates information from samples drawn up to time $t$

$$l_t(D) = (1/2t)\sum_{i=1}^{t} \|x_i - D\alpha_i\|_2^2$$

- At each step, given $D_{t-1}$, $\alpha_t$ is obtained using $x_t$ via any method, e.g., coordinate descent with soft thresholding, LARS with Cholesky etc.
- Dictionary learning based on block coordinate descent
- Does not require learning rate tuning
- With increasing $t$, the storage cost of $x_i$ and $\alpha_i$, $i < t$ can be expensive
  - No need to store these explicitly, only need to store $A_t$ and $B_t$
Complete Algorithm

Given training set $X$, coefficient $\lambda$, initial dictionary $D_0$ and iterations $T$

$A_0 = 0, B_0 = 0$

For $t = 1, \ldots, T$

1. Draw a random point $x_t$ from $X$

2. Find best $\alpha_t$, for $x_t$ given $D_{t-1}$

$$\alpha_t = \arg \min_\alpha [(1/2)\|x_t - D_{t-1}\alpha\|_2^2 + \lambda \|\alpha\|_1]$$

use Lasso / LARS / iterative thresholding

3. Update $A_t \leftarrow A_{t-1} + \alpha_t \alpha_t^T$  
   $B_t \leftarrow B_{t-1} + x_t \alpha_t^T$
**Complete Algorithm**

Given training set $X$, coefficient $\lambda$, initial dictionary $D_0$ and iterations $T$

$A_0 = 0, B_0 = 0$

For $t = 1, \ldots, T$

1. Draw a random point $x_t$ from $X$
2. Find best $\alpha_t$, for $x_t$ given $D_{t-1}$

$$\alpha_t = \arg \min_{\alpha} \left[ (1/2) \|x_t - D_{t-1}\alpha\|_2^2 + \lambda \|\alpha\|_1 \right]$$

   use Lasso / LARS / iterative thresholding

3. Update $A_t \leftarrow A_{t-1} + \alpha_t a_t^T$  $B_t \leftarrow B_{t-1} + x_t \alpha_t^T$

4. Update dictionary using block coordinate descent (starting with $D_{t-1}$)

$$d_j \leftarrow \frac{1}{A_{jj}} (b_j - Da_j) + d_j$$

   if $\|d_j\|_2 > 1 \Rightarrow d_j \leftarrow d_j / \|d_j\|_2$

   or using Newton’s method by first creating a Lagrangian using constraints

   • Need to invert $k \times k$ matrix (fine for small $k$)

What happens if $T > n$? Various heuristics to remove the contribution of older $\alpha$’s!
Dictionary Learning Experiment

\[ n = 1.25M \quad \text{image patches} \]
\[ d = 256, \quad k = 1024 \]
Dictionary Learning Experiment

Inpainting, e.g., remove text from images

\[ n = 7M \text{ patches, } d = 432, \ k = 256 \]
Training time: 8 mins, 2.4GHz/8 cores

Mairal et al. [13]
Sparse PCA

Key Idea: Find sparse vectors that maximize the variance of the projected data or minimize mean squared error of reconstruction

Benefits: Less storage and computation, Interpretability

Formulation 1

– Maximize variance subject to sparsity constraints
Sparse PCA

Key Idea: Find sparse vectors that maximize the variance of the projected data or minimize mean squared error of reconstruction.

Benefits: Less storage and computation, Interpretability.

Formulation 1

- Maximize variance subject to sparsity constraints.

Suppose data matrix $X$ is centered, i.e., zero mean

$$
\hat{D} = \arg \max_D \text{Tr}[D^T XX^T D] \quad X \in \mathbb{R}^{d \times n}, D \in \mathbb{R}^{d \times k}
$$

subject to

$$
D^T D = I
$$

$$
\sum_{j=1}^k \|d_j\|_1 \leq C
$$

- Usually each $d_j$ is learned incrementally by enforcing orthonormality and sparsity constraints.
- Problem is non-convex and computationally expensive.
Sparse PCA

Key Idea: Find **sparse** vectors that maximize the variance of the projected data or minimize mean squared error of reconstruction

Formulation 2
- Minimize squared reconstruction error subject to sparsity constraints

Recall PCA formulation

\[
\hat{D} = \arg \min_D \left\| X - DD^T X \right\|_2^2 \quad \text{subject to} \quad D^T D = I
\]
Sparse PCA

Key Idea: Find sparse vectors that maximize the variance of the projected data or minimize mean squared error of reconstruction

Formulation 2
- Minimize squared reconstruction error subject to sparsity constraints

Recall PCA formulation
\[
\hat{D} = \arg\min_D \left\| X - DD^T X \right\|_2^2 \quad \text{subject to} \quad D^T D = I
\]

Sparse PCA formulation
\[
\hat{B}, \hat{D} = \arg\min_{B,D} \left\| X - BD^T X \right\|_2^2 + \lambda_1 \sum_{j=1}^k \|d_j\|_2^2 + \lambda_2 \sum_{j=1}^k \|d_j\|_1 \quad \text{subject to} \quad B^T B = I
\]

- If no L1 penalty, solution the same as PCA
- Solved via alternate minimization, using LARS and SVD
- Much more expensive than PCA
Sparse PCA

Key Idea: Find sparse vectors that maximize the variance of the projected data or minimize mean squared error of reconstruction

Formulation 3

- Minimize square reconstruction via dictionary learning view

\[
\min_{D,\alpha} \sum_{i=1}^{n} [(1/2)\|x_i - Da_i\|_2^2 + \lambda\|a_i\|_1] \quad \text{sparse } \alpha
\]

subject to \( \|d_j\|_2^2 + \gamma\|d_j\|_1 \leq 1 \quad j = 1,\ldots,k \quad \text{sparse } d_j \)
Sparse PCA

Key Idea: Find sparse vectors that maximize the variance of the projected data or minimize mean squared error of reconstruction

Formulation 3

– Minimize square reconstruction via dictionary learning view

\[
\min_{D, \alpha} \sum_{i=1}^{n} [(1/2)\|x_i - Da_i\|_2^2 + \lambda \|a_i\|_1] \quad \text{sparse } \alpha
\]

subject to \( \|d_j\|_2^2 + \gamma \|d_j\|_1 \leq 1 \quad j = 1, ..., k \) \quad \text{sparse } d_j

– Solved via alternate minimization as for sparse coding

– Dictionary learning step is modified as

\[
d_j' \leftarrow \frac{1}{A_{jj}} (b_j - Da_j) + d_j
\]

subject to \( d_j \leftarrow \arg\min_d \|d' - d\|_2^2 \) \quad \text{s.t. } \|d\|_2^2 + \gamma \|d\|_1 \leq 1 \quad j = 1, ..., k

No orthonormality constraints on \( d_j \)!
Dictionary Learning Experiment

\[ n = 2414 \quad \text{face images} \]
\[ d = 256, \quad k = 49 \]

(a) PCA  \quad (b) SPCA, \tau = 70\% \quad (c) Dictionary Learning

Mairal et al. [13]
References