

Basics

Scalar product: $\mathbf{x}^T \mathbf{y} = \|\mathbf{x}\|_2 \cos \theta$

L_2 – Norm: $\|x\|_2 = \sqrt{x^T x}$

Spectral Norm: $\|A\|_2 = \sigma_{max}(A)$

Nuclear Norm: $\|A\|_* = \sum_{i=1}^{\min\{m,n\}} \sigma_i \quad (\sigma : \text{Singular Value})$

Frobenius Norm: $\|A\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2}$

Statistics

Expectation: $\mathbb{E}[\mathbf{X}] = (\mathbb{E}[X_1], \dots, \mathbb{E}[X_D])^T$

Covariance: $Cov[X, Y] = \int_{\mathcal{X}} \int \mathcal{Y} p(x, y) (x - \mu_X)(y - \mu_Y) dx dy$
 $= E_{X,Y}[(x - \mu_X)(y - \mu_Y)]$

Cov Matrix Σ : $\Sigma_{i,j} := Cov[\mathbf{X}_i, \mathbf{X}_j]$

Gaussian Distribution

$g(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

With μ being the mean and σ the standard deviation.

$g(\mathbf{x}; \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^D \cdot |\Sigma|^{\frac{1}{2}}}} e^{\frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1}(\mathbf{x}-\mu)}$

D Dimensions, Σ Covariance Matrix

Bayes’ Theorem

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} = \frac{P(B|A) \cdot P(A)}{\sum_i P(B|A_i) \cdot P(A_i)}$$

Eigenvalue Decomposition

$\exists u : Au = \lambda u \quad A = U \cdot \Lambda \cdot U$

If $A^T \cdot A = A \cdot A^T$ then $A = U \cdot \Lambda \cdot U^{-1} = U \cdot \Lambda \cdot U^T$

Convex

A function $f : \mathbb{R}^n \mapsto \mathbb{R}$ is *convex* if $dom\ f$ is a a convex set and if for all $x, y \in dom\ f$, and $\theta \in [0, 1]$ we have:

$f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y).$

Singular Value Decomposition

Let $A \in \mathbb{R}^{m \times n}$. A can be decomposed as

$$A = \underbrace{U}_{\mathbb{R}^m \times m} \cdot \underbrace{D}_{\mathbb{R}^m \times n} \cdot \underbrace{V^T}_{\mathbb{R}^n \times n}$$

Principal Component Analysis

Minimize error $\|x_n - \tilde{x}_n\|_2$ of point x_n and it’s approximation \tilde{x}_n . Algorithm, for $\mathbf{X} \in \mathbb{R}^{D \times N}$:

- 1. Compute the zero-centered data $\bar{\mathbf{X}}$ by subtracting the mean of the sample $\bar{\mathbf{X}} = \mathbf{X} - \mathbf{M}$.
- 2. Calculate the covariance matrix $\Sigma = \frac{1}{N} \bar{\mathbf{X}} \bar{\mathbf{X}}^T$

- 3. Compute the eigenvectors \mathbf{U} and eigenvalues Λ of the covariance matrix
- 4. Compute the projection of $\bar{\mathbf{X}}$ on the largest k principal components $\mathbf{U}_k = [u_1, \dots, u_k]$ by $\bar{\mathbf{Z}}_k = \mathbf{U}_k^T \bar{\mathbf{X}}$

To approximate $\bar{\mathbf{X}}$ we return to the original basis by $\tilde{\bar{\mathbf{X}}} = \mathbf{U}_k \bar{\mathbf{Z}}_k$

K-means

$$J(\mathbf{U}, \mathbf{Z}) = \|\mathbf{X} - \mathbf{UZ}\|_2^2 = \sum_{n=1}^N \sum_{k=1}^K z_{k,n} \|\mathbf{x}_n - \mathbf{u}_k\|_2^2$$

Data: $\mathbf{X} = [\mathbf{x}_1 \dots \mathbf{x}_N] \in \mathbb{R}^{D \times N}$, centroids:
 $\mathbf{U} = [\mathbf{u}_1 \dots \mathbf{u}_K] \in \mathbb{R}^{D \times K}$ and the assignment $\mathbf{Z} \in \{0, 1\}^{K \times N}$

Algorithm

- 1. Initiate $\mathbf{u}_1^{(0)}, \dots, \mathbf{u}_K^{(0)}$ (random choice or data points from the set)
- 2. *Cluster assignment.* Solve $\forall n$:
 $k^*(\mathbf{x}_n) = \arg \min \left\{ \|\mathbf{x}_n - \mathbf{u}_1^{(t)}\|_2^2, \dots, \|\mathbf{x}_n - \mathbf{u}_K^{(t)}\|_2^2 \right\}.$
Then, $z_{k^*(\mathbf{x}_n),n}^{(t)} = 1$ and $z_{j,n}^{(t)} \forall k, k \in \{1, \dots, K\}$
- 3. *Centroid update.*
$$\mathbf{u}_k^{(t)} = \frac{\sum_{n=1}^N z_{k,n}^{(t)} \mathbf{x}_n}{\sum_{n=1}^N z_{k,n}^{(t)}} \quad \forall k, k \in \{1, \dots, K\}$$
- 4. Increment t . Repeat step 2 until $\|\mathbf{u}_k^{(t)} - \mathbf{u}_k^{t-1}\|_2^2 < \varepsilon \forall k$ or until $t = t_{\text{finish}}$

Convergence is guaranteed, optimizes a non-convex objective \implies we can only guarantee to find a local minimum.

Stability

Cluster data and train classifier and test permuted output.

$$r := \frac{1}{N} \min_{\pi \in P_K} \left\{ \sum_{i=1}^N \mathbb{I}_{\{\pi(\varphi(x'_i)) \neq z'_i\}} \right\}$$

Given K clusters of equal size, a random assignment yields $r_{\text{rand}}) \frac{K-1}{K}$. The stability is thus defined as:

$$stab := 1 - \frac{r}{r_{\text{rand}}} = \begin{cases} 1 & \text{No inconsistent assignments} \\ 0 & \text{The output is random.} \end{cases}$$

Test clustering stability by generating perturbed versions of the set and applying the clustering algorithm.

Gaussian Mixture Model

Mixture of K probaility densities is defined as:

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k p(\mathbf{x}|\theta_k) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$$

$$p(\mathbf{X}|\pi, \mu, \Sigma) = \prod_{n=1}^N p(\mathbf{x}_n) = \prod_{n=1}^N \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n|\mu_k, \Sigma_k).$$

We want to find the parameters that maximize the likelihood:

$$(\hat{\mathbf{x}}, \hat{\mu}, \hat{\Sigma}) \in \arg \max_{\pi, \mu, \Sigma} \sum_{n=1}^N \log \left\{ \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n|\mu_k, \Sigma_k) \right\}$$

Algo: Initialize μ_k and π_k . Set the Σ_k

Eval: $\gamma(z_{k,n}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\mu_k, \Sigma_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}_n|\mu_j, \Sigma_j)}$

Update: $\mu_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{k,n}) \mathbf{x}_n, \pi_k^{\text{new}} = \frac{N_k}{N}, N_k = \sum_{n=1}^N \gamma(z_{k,n})$

RBAC

Given user-permission matrix \mathbf{X} , find roles \mathbf{U} and assignments \mathbf{Z} such that

$$\mathbf{X} = \mathbf{U} \bigotimes \mathbf{Z} \iff x_{dn} = \bigvee_k [u_{dk} \wedge z_{kn}]$$

$$p(\mathbf{X}|\beta, \mathbf{Z}) = \prod_{n,d} \left(1 - \prod_k \beta_{dk}^{z_{kn}} \right)^{x_{dn}} \left(\prod_k \beta_{dk}^{z_{kn}} \right)^{1-x_{dn}} \quad (\text{mult. assign})$$

Noise model: $x_{dn} = (1 - \xi_{dn})(\mathbf{U} \otimes \mathbf{Z})_{dn} + \xi_{dn} \nu_{dn}$. Final Model:

$$p(\mathbf{X}|\mathbf{Z}, \beta, \varepsilon, r) = \prod_{n,d} (\varepsilon r + (1 - \varepsilon)(1 - \beta_{d,\mathcal{L}_n}))^{x_{dn}} \cdot (\varepsilon(1 - r) + (1 - \varepsilon)\beta_{d,\mathcal{L}_n})^{1-x_{dn}}$$

ε : Noise probability, r probability of noisy bits to be 1 and β probabilities of role-permission assignments \mathbf{U} to be 0.

Not convex! Use an EM-type algorithm to maximize the function.

Evaluating a Matrix Reconstruction

The deviation is the fraction of wrongly predicted definitions.

Deviation: $\frac{1}{N \cdot D} \|\mathbf{X} - \hat{\mathbf{U}} \hat{\otimes} \hat{\mathbf{Z}}\|_1$

Coverage: $Cov = \frac{|\{(i,j)|\hat{x}_{i,j} = x_{i,j} = 1\}|}{|\{(i,j)|x_{i,j} = 1\}|}$

Non-Negative MF

Given Document-term matrix $\mathbf{X} \in \mathbb{R}_+^{D \times N}$. We want a NMF for which holds:

$$\mathbf{X} \approx \mathbf{UZ} \quad \text{with } \mathbf{U} \in \mathbb{R}_+^{D \times K} \text{ and } \mathbf{Z} \in \mathbb{R}_+^{K \times N}$$

Probabilistic LSI

In order to generate a tuple (*document, word*):

- Sample document according to $P(\text{document})$
- Sample word according to $P(\text{word}|\text{document})$
- Assume a factorization
$$P(\text{word}|\text{document}) = \sum_{\text{topic}} P(\text{word}|\text{topic})P(\text{topic}|\text{document})$$

which can be written as

$$P(d - th\ word, n - th\ document) = x_{dn} = (\mathbf{UZ})_{dn}$$

The pLSI is computed using a non-negative \mathbf{X} and a quadratic cost function:

$$\min_{\mathbf{U}, \mathbf{Z}} J(\mathbf{U}, \mathbf{Z}) = \frac{1}{2} \|\mathbf{X} - \mathbf{UZ}\|_2^2 \quad u_{dk}, z_{kn} \in \mathbb{R}_0^+,$$

Algorithm:

$$\mathbf{U} = rand(D, K), \mathbf{Z} = rand(K, N)$$

for $i = 1\ maxiter$ do

$$\text{Update factors } \mathbf{U} : u_{dk} = u_{dk} \frac{(\mathbf{XZ}^T)_{dk}}{(\mathbf{UZZ}^T)_{dk}}$$

$$\text{Update coefficients } \mathbf{Z} : z_{kn} = z_{kn} \frac{(\mathbf{U}^T \mathbf{X})_{kn}}{(\mathbf{U}^T \mathbf{UZ})_{kn}}$$

This leads to $\mathbf{X} \approx \mathbf{UZ}$ when $K < N$.

Show monotonic convergence

To prove that some non-negative $J(z)$ is guaranteed to converge we can follow these steps

- Define *auxiliary function* $G(z, z')$ for $J(z)$ such that:
 $G(z, z') \geq J(z)$ and $G(z, z) = J(z)$
- Find a local minimum of G by following repeatedly
 $z^{t+1} = \arg \min_z G(z, z^t)$
- The sequence $\{z^t\}$ is converging to a local minimum of $J(z)$
 $J(z^{t+1}) \leq G(z^{t+1}, z^t) \leq G(z^t, z^t) = J(z^t)$

Example:

$$G(\mathbf{Z}, \mathbf{Z}^t) = J(\mathbf{Z}^t) + (\mathbf{Z} - \mathbf{Z}^t) \nabla J(\mathbf{Z}^t) + \frac{1}{2} (\mathbf{Z} - \mathbf{Z}^t)^T M (\mathbf{Z} - \mathbf{Z}^t)$$

$$M_{kn}(Z^t) = \delta_{kn} \frac{(\mathbf{U}^T \mathbf{UZ}^t)_k}{\mathbf{Z}_k^t} \quad \delta_{kn} : \text{Kronecker delta}$$

Sparse Coding

$$f = \sum_{l=1}^L z_l \underbrace{u_l}_{\text{base}} = \sum_{l=1}^L \underbrace{\langle f, u_l \rangle}_{\text{signal}} u_l \quad \text{Compression: } \hat{f} = \sum_{k \in \sigma} z_k u_k$$

$$\text{Error: } \|f - \hat{f}\|^2 = \sum_{k \notin \sigma} |\langle f, u_k \rangle|^2$$

Compressive Sensing

$$\mathbf{x} = \mathbf{UZ} \quad \mathbf{U} \text{ basis}$$

$$\mathbf{y} = \mathbf{Wx} = \mathbf{WUZ} := \mathbf{\Theta z} \quad \mathbf{\Theta} = \mathbf{WU} \in \mathbb{R}^{M \times D}$$

- All elements in $w_{i,j}$ of Matrix \mathbf{W} are i.i.d. random variables with a Gaussian distribution with zero mean and variance $\frac{1}{D}$.
- $M : M \leq cK \log\left(\frac{D}{K}\right)$, where c is some constant.

Since \mathbf{x} and \mathbf{z} are both unknown, \mathbf{z} can be reconstructed with

$$z^* = \arg \min_z \|z\|_0 \quad \text{s.t. } \mathbf{\Theta z} = \mathbf{y}$$

NP hard, solve this with Matching Pursuit.

Coherence

Increasing the overcompleteness factor $\frac{L}{D}$: Increases the sparsity of the coding and increases the linear dependency between atoms.

$$\text{coherence: } m(\mathbf{U}) = \max_{i,j:i \neq j} |\mathbf{u}_i^T \mathbf{u}_j|$$

- $m(\mathbf{B}) = 0$ for an orthogonal basis \mathbf{B}
- $m([\mathbf{B}\mathbf{u}]) \geq \frac{1}{\sqrt{D}}$ if atom \mathbf{u} added to \mathbf{B}

Overcompleteness and noise

$$\text{Overcompleteness: } z^* = \arg \min_z \|\mathbf{z}\|_0$$

$$\text{s.t. } \mathbf{x} = \mathbf{UZ}$$

$$\text{Noise: } z^* = \arg \min_z \|\mathbf{z}\|_0 \quad \mathbf{x} = \mathbf{UZ} + n \text{ with } n \sim \mathcal{N}(0, \sigma^2)$$

$$\text{s.t. } \|\mathbf{x} - \mathbf{UZ}\|_2 < \sigma$$

Matching Pursuit (MP)

Greedy algorithm: Approximate NP hard problem iteratively.

Applied to sparse coding:

- Start with zero vector $\mathbf{z} = \mathbf{0}$ and residual $\mathbf{r}^0 = \mathbf{x}$
- At each iteration t , take a step in the direction of the atom $\mathbf{u}_{d^*(t)}$ that maximally reduces the residual $\|\mathbf{x} - \mathbf{UZ}\|_2$.
Criteria: $d^*(t) = \arg \max_d |\langle \mathbf{t}^t, \mathbf{u}_d \rangle|$
Update: $z_{d^*} \leftarrow z_{d^*} + \mathbf{u}_{d^*}^T \mathbf{r}$, $\mathbf{r} \leftarrow \mathbf{r} - (\mathbf{u}_{d^*}^T \mathbf{r}) \mathbf{u}_{d^*}$

Exact recovery when $K < \frac{1}{2} \left(1 + \frac{1}{m(\mathbf{U})}\right)$ (K : # non-Zero el.)

Dictionary Learning

Factorize training set $\mathbf{X} \in \mathbb{R}^{D \times N}$ into a dictionary $\mathbf{U} \in \mathbb{R}^{D \times L}$ and sparsity constraint $\mathbf{Z} \in \{0, 1\}^{L \times N}$ such that:

$(\mathbf{U}^*, \mathbf{Z}^*) \in \arg \min_{\mathbf{U}, \mathbf{Z}} \|\mathbf{X} - \mathbf{U} \cdot \mathbf{Z}\|_F^2$ (not convex in \mathbf{U} and \mathbf{Z}) but convex in either \mathbf{U} or \mathbf{Z} .

- Coding step*: $\mathbf{Z}^{t+1} \in \arg \min_{\mathbf{Z}} \|\mathbf{X} - \mathbf{U}^t \cdot \mathbf{Z}\|_F^2$, subject to \mathbf{Z} being sparse and fixed \mathbf{U} such that:
 $\|\mathbf{x}_n - \mathbf{U}^t \mathbf{z}\|_2 \leq \sigma \|\mathbf{x}_n\|_2$
- Dictionary update*: $\mathbf{U}^{t+1} \in \arg \min_{\mathbf{U}} \|\mathbf{X} - \mathbf{U}^t \cdot \mathbf{Z}\|_F^2$, subject to $\|\mathbf{u}_l\|_2 = 1 \forall l \in [1, L]$ and fixed \mathbf{Z} .
Approximation: update one atom at a time for all $l = 1, \dots, L$:

(a) Set $\tilde{\mathbf{U}} = [\mathbf{u}_1^t \dots \mathbf{u}_l \dots \mathbf{u}_L^t]$ (fix all atoms except \mathbf{u}_l).

(b) Isolate \mathbf{R}_l^t , the residual that is due to atom \mathbf{u}_l :
 $\|\mathbf{X} - \tilde{\mathbf{U}} \cdot \mathbf{Z}^{t+1}\|_F^2 = \|\mathbf{R}_l^t - \mathbf{u}_l (\mathbf{z}_l^{t+1})^T\|_F^2$

(c) Find \mathbf{u}_l^* that minimizes \mathbf{R}_l^t , subject to $\|\mathbf{u}_l^*\|_2 = 1$.
with SVD of \mathbf{R}_l^t : $\mathbf{R}_l^t = \tilde{\mathbf{U}} \mathbf{S} \tilde{\mathbf{V}}^T = \sum_i s_i \tilde{\mathbf{u}}_i \tilde{\mathbf{v}}_i^T$

RPCA

Convex Optimization

Minimize $f(x)$

subject to $g_i(x) \leq 0$, $i = 1, \dots, m$ and $h_i(x) = 0$, $i = 1, \dots, p$

Lagrange Multipliers

$$\text{minimize } f(x) + \sum_{i=1}^m I_-(g_i(x)) + \sum_{i=1}^p I_0(h_i(x)) \quad (\text{unconstrained pr.})$$

$$\diamond I_- = \begin{cases} 0 & u \leq 0 \\ \infty & u > 0 \end{cases} \quad \diamond I_0 = \begin{cases} 0 & u = 0 \\ \infty & u \neq 0 \end{cases}$$

Approximate $I_-(u)$ linearly with $\lambda_i u$, $\lambda_i \geq 0$, and $I_0(u)$ with $\nu_i u$:

$$L(x, \lambda, \nu) = f(x) + \sum_{i=1}^m \lambda_i g_i(x) + \sum_{i=1}^p \nu_i h_i(x) \quad \underbrace{d(\lambda, \nu) = \inf_x L(x, \lambda, \nu)}_{\text{dual function}}$$

$$\text{Lagrange dual problem: } \begin{cases} \text{maximize} & d(\lambda, \nu) \\ \text{subject to} & \lambda \geq 0 \end{cases}$$

Convex Optimization Problem

$$\text{Minimize } f(x) \quad \text{subject to } Ax = b$$

$$\text{Lagrangian: } L(x, \nu) = f(x) + \nu^T (Ax - b)$$

$$\text{Dual function: } d(\nu) = \inf_x L(x, \nu)$$

$$\text{Dual problem: maximize } d(\nu)$$

$$\text{Recover optimal } x: x^* = \arg \min_x L(x, \nu^*)$$

Method of Multipliers

Augmented Lagrangian:

$$L_\rho(x, \nu) = f(x) + \nu^T (Ax - b) + \frac{\rho}{2} \|Ax - b\|_2^2$$

$$\text{Step: } x^{k+1} := \arg \min_x L_\rho(x, \nu^k)$$

$$\nu^{k+1} := \nu^k + \rho(Ax^{k+1} - b)$$

Choose ρ as step size, since x^{k+1} minimizes $L_\rho(x, \nu^k)$:

$$0 = \nabla_x L_\rho(x^{k+1}, \nu^k) = \nabla_x f(x^{k+1}) + \underbrace{A^T(\nu^k + \rho(Ax^{k+1} - b))}_{A^T \nu^{k+1}}$$

Alternating Direction Method of Multipliers

The augmented Lagrangian L_ρ is not separable anymore \implies we can't parallelize x -minimization.

Minimize: $f(x) + p(z)$ subject to $Ax + Bz = c$ f, p convex.

$$L_\rho(x, z, \nu) = f(x) + p(z) + \nu^T (Ax + Bz - c) + \frac{\rho}{2} \|Ax + Bz - c\|_2^2$$

$$\text{Steps: } x^{k+1} := \arg \min_x L_\rho(x, z^k, \nu^k)$$

$$z^{k+1} := \arg \min_z L_\rho(x^{k+1}, z, \nu^k)$$

$$\nu^{k+1} := \nu^k + \rho(Ax^{k+1} + Bz^{k+1} - c)$$

Conditions: $Ax^* + Bz^* - c = 0$ (Primal Feasibility)

$$\nabla f(x^*) + A^T \nu^* = 0 \text{ (Dual Feasibility)}$$

$$\nabla p(z^*) + B^T \nu^* = 0$$

Robust PCA

Decompose matrix into low-rank (\mathbf{L}_0) and sparse (\mathbf{S}_0) part:

$$\mathbf{X} \approx \mathbf{L}_0 + \mathbf{S}_0.$$

Use convex relaxation:

$$\text{Minimize } \|\mathbf{L}\|_* + \lambda \|\mathbf{S}\|_1, \quad \text{subject to } \mathbf{L} + \mathbf{S} = \mathbf{X}.$$

$$\mathbf{L}_0 : n \times n, \text{ of rank}(\mathbf{L}_0) \leq \rho_r n \mu^{-1} (\log n)^{-2}$$

$$\mathbf{S}_0 : n \times n, \text{ random sparsity pattern of cardinality } m \leq \rho_s n^2$$

With probability $1 - \mathcal{O}(n^{-10})$, PCP with $\lambda = \frac{1}{\sqrt{n}}$ is exact.