1 Dimensionality Reduction

1.1 Principal Component Analysis

Minimizing error $\|x_n - \tilde{x}_n\|$ and maximizing variance and therefore revealing interesting information.

Covariance of the data with mean $\bar{x}$:

$$\Sigma = \frac{1}{N}\sum_{n=1}^{N}(x_n - \bar{x})(x_n - \bar{x})^\top$$

With the mean of the projected data being $u_i^\top \bar{x}$ we maximize the variance $u_i^\top \Sigma u_1$ such that $\|u_i\|_2 = 1$.

Using the Lagrangian of this optimization problem and setting the derivative to zero we get:

$$\Sigma u_1 = \lambda_1 u_1$$

Eigendecomposition Problem $\Sigma = U\Lambda U^\top$: To maximize the variance we simply choose the eigenvector with the larges associated eigenvalue. This is called the first (n^{th}) **principal direction**.

For $K \leq D$ dimensional projection space we choose K eigenvectors $\{u_1, \ldots, u_K\}$ with largest associated eigenvalues $\{\lambda_1, \ldots, \lambda_K\}$. We call these eigenvectors the **principal components** in a PCA of A.

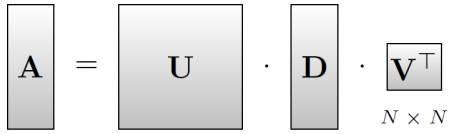
1.1.1 Matrix Viewpoint

When computing the projection of $\bar{X}$ on $U_k = [u_1, \ldots, u_K]$ (eigenvectors with the K highest eigenvalues of covariance matrix Σ) we get:

$$\begin{aligned} \bar{Z}_K &= U_K^\top \cdot \bar{X} && \text{(project on } U_k\text{)} \\ \tilde{\bar{X}} &= U_K \cdot \bar{Z}_K && \text{(original basis)} \\ \tilde{X} &= \tilde{\bar{X}} + M && \text{(re-add mean)} \end{aligned}$$

1.2 Singular Value Decomposition (SVD)

Every rectangular matrix has an SVD decomposition into a set of three matrix factors:



With $U, V^\top$ orthogonal and D diagonal. The non-zero elements in D on the diagonal are called the **singular values** and are equal to the square roots of the eigenvalues of $AA^\top$ and $A^\top A$. The corresponding eigenvectors are the columns of U and the rows of $V^\top$, respectively.

The first r columns of U are called the **left singular vectors** and form an orthogonal basis for the space spanned by the columns of the original matrix A. Similar with rows of $V^\top$ which form row space of A.

In Collaborative Filtering (Movie Rating):

U : User-to-concept affinity

V : Movie-to-concept affinity

D : Strength of concept

1.2.1 Closest Rank-k Matrix

With $A_k = \sum_{i=1}^{k} d_i u_i v_i^\top$ being a matrix with rank $k < r = rank(A)$ we have the closest rank-k approximation to A in the Euclidean matrix norm sense.

Magnitudes of the nonzero singular values provide a measure of approximation to A_k: $\|A - A_k\|_2 = d_{k+1}$. If square matrix $\rightarrow$ spectral norm: $\|A\|_2 = \sigma_{max}(A)$

1.2.2 Computing the SVD

Given a matrix $A \in \mathbb{R}^{M \times N}$ (assume $N < M$)

1. Eigenvalue-decomposition of $A^\top A$. Fill roots of eigenvalues into D.
2. Compute the eigenvectors of $A^\top A$. Place them (in the right order) along the columns of V. (rows of $V^\top$)
3. Compute the matrix U as AVD^{-1}. ($D_{ii}^{-1} = \frac{1}{D_{ii}}$)

If S is a real and symmetric matrix ($S = S^\top$) then $S = UDU^\top$.

2 Clustering

Assign data points to cluster and minimize the following cost function:

$$J(U, Z) = \|X - UZ\|_F^2 = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{k,n}\|x_n - u_k\|_2^2$$

Where $X = [x_1 \cdots x_N] \in \mathbb{R}^{D \times N}$, $U = [u_1 \cdots u_K] \in \mathbb{R}^{D \times K}$. We call the u_k the **centroids** and z_n the assignments of data points to clusters.

2.1 K-Means

Hard assignment: $Z \in \{0,1\}^{K \times N}$ with $\sum_k z_{k,n} = 1 \; \forall n$.

1. Initialize centroids
2. Assign data points to clusters.
   $k^*(x_n)$ index with the minimal distance:
   $$k^*(x_n) = \arg\min_k \left\{\|x_n - u_1\|_2^2 \quad \forall i\right\}$$
3. Update Cluster Centroids:
   Compute the mean/centroid of a cluster:
   $$u_k = \frac{\sum_{n=1}^{N} z_{k,n} x_n}{\sum_{n=1}^{N} z_{k,n}} \quad \forall k, k \in \{1, \ldots, K\}$$

Iterate until ($\mathcal{O}(KN)$ per iteration):
$$\|u_k^{(t)} - u_k^{(t-1)}\|_2^2 < \epsilon \quad \forall k \text{ with } (0 < \epsilon \ll 1)$$
or until $t = t_{finish}$

- K-means convergence is guaranteed
- Non-convex objective, local minima, sensitive to initializations. $\rightarrow$ restarts.

2.1.1 Clustering Stability

1. Generate pertrubed versions of the set
2. Apply algorithm on all versions
3. Pair-wise distance between clusterings
4. Compute the **instability** as the mean distance between all clusterings

Repeat for different numbers of clusters and choose the one that minimizes the instability.

The distance between two clusterings C and C' can be calculated as follows:

$$d = \min_{\pi} \|Z - \pi(Z')\|_0$$

where $\pi(Z')$ is one of the possible row permutations of Z' and $\|Z\|_0$ denotes the cardinality of Z.

2.2 Mixture Models (Soft Clustering)

Relax the hard constraint given by $Z \in \{0,1\}^{K \times N}$ with $\sum_k z_{k,n} = 1 \; \forall n$ from k-means with a soft one: $z_{k,n} \in [0,1]$ with $\sum_{k=1}^{K} z_{k,n} = 1 \quad \forall n$. Definition:

$$p(x) = \sum_{k=1}^{K} \pi_k p(x|\Theta_k)$$

2.2.1 Gaussian Mixture Models

Independent identically distributed points. Use **Expectation-Maximation** to find maximum likelihood solutions for models with latent variables. Find parameters π, μ, Σ.

$$\ln\, p(X|\pi, \mu, \Sigma) = \sum_{n=1}^{N} \ln\left\{\sum_{k=1}^{N} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)\right\}$$

$$\mu_k = \frac{1}{N_k}\sum_{n=1}^{N} \gamma(z_{k,n}) x_n, \; N_k = \sum_{n=1}^{N} \gamma(z_{k,n})$$

1. Initialize the means μ_k and mixing coefficients π_k. Set the Σ_k to the given covariances.
2. **E-Step** Evaluate the responsibilities:
   $$\gamma(z_{k,n}) = \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_j \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)}$$
3. **M-Step** Re-estimate the parameters:
   $$\mu_k^{new} = \frac{1}{N_k}\sum_{n=1}^{N} \gamma(z_{k,n} x_n)$$
   $$\pi_k^{new} = \frac{N_k}{N} \quad \text{where } N_k = \sum_{n=1}^{N} \gamma(z_{k,n})$$
4. Evaluate the log likelihood; check for convergence of parameters or log likelihood

Balance Complexity and Data Fit
Balance data fit (likelihood $p(X|\cdot)$) and complexity (i.e. free parameters $\kappa(\cdot)$)
Akaike Information Criterion (AIC):
$AIC(U,Z|x_1,\dots,x_N) = -\ln p(X|\cdot) + \kappa(U,Z)$

Bayesian Information Criterion (BIC):
$$BIC(U,Z|x_{ns}) = -\ln p(X|\cdot) + \frac{1}{2}\kappa(U,Z)\ln N$$

BIC criterion penalizes complexity more than AIC criterion. Most suitable number of clusters corresponds to the smalles AIC (BIC) value.

3 Multi Assignment Clustering

3.1 Binary Matrix Factorization

To infer a role-based access control system out of a discretionary one (one big user-permission matrix), we can use binary matrix factorization.

Min-Noise Approximation: Given K find the matrices $\hat{U}$, $\hat{Z}$ so that:

$$(\hat{U},\hat{Z}) = \arg\min_{U,Z} \|X - U \otimes Z\|_1$$
$$\text{with } U \in \mathbb{B}^{D\times K} \text{ and } Z \in \mathbb{B}^{K\times N}$$

Common methods for Boolean matrix factorization are:

Rounded SVD $X = U \cdot S \cdot V^\top$ (e. g. roles: $\hat{U} = (U_{(K)} > t_U)$). Very poor performance.
K-means with Hamming distance Use Hamming distance (0-norm) and restrict centroids u_k to boolean values. No multi assignments possible with k-means!
RoleMiner Roles are created by finding common sets of permissions between users. However is very sensitive to noise.
DBPsolver Approximate solution for the **Discrete Basis Problem** (Minimizing $\|X - U \otimes Z\|_F^2$ for given X)

3.1.1 RBAC

$X = U \otimes Z \Leftrightarrow x_{dn} = \bigvee_k \left[u_{dk} \wedge z_{kn}\right]$
SAC vs. MAC:

$$p(X|\beta,Z) = \prod_{n,d}(1-\beta_{dk_n})^{x_d n}(\beta_{dk_n})^{1-x_d n}$$
$$\text{MAC} = \prod_{n,d}(1-\prod_k \beta_{dk_n}^{z_{kn}})^{x_d n}(\prod_k \beta_{dk_n}^{z_{kn}})^{1-x_d n}$$

Mixtrue Noise Model:
$x_{dn} = (1-\xi_{dn})(U \otimes Z)_{dn} + \xi_{dn}\eta_{dn}$
ξ_{dn}: binary noise indicator
η_{dn}: binary random variable

4 Non-negative Matrix Factorization

Find $U, Z : X \approx U \cdot Z$:

$$\min_{U,Z} \quad J(U,Z) = \frac{1}{2}\|X - UZ\|_F^2$$
$$\text{s.t.} \quad u_{dk} \in [0,\infty) \forall d,k$$
$$z_{kn} \in [0,\infty) \forall k,n$$

Algorithm for Quadratic Cost Function:
1: $\mathbf{U} \leftarrow \text{rand}(D,K)$, $\mathbf{Z} \leftarrow \text{rand}(K,N)$
2: **for** i = 1:maxiter **do**
3: Update factors $\mathbf{U}$: $u_{dk} \leftarrow u_{dk}\frac{(\mathbf{XZ}^\top)_{dk}}{(\mathbf{UZZ}^\top)_{dk}}$.
4: Update coefficients $\mathbf{Z}$: $z_{kn} \leftarrow z_{kn}\frac{(\mathbf{U}^\top\mathbf{X})_{kn}}{(\mathbf{U}^\top\mathbf{UZ})_{kn}}$
5: **end for**

If we allow negative entries in U we get a **semi-NMF** algorithm iterating two steps:

1. Data matrix U is updated:
$$U = XZ^\top(ZZ^\top)^{-1}$$
2. Update Z:
$$z_{kn} \leftarrow z_{kn}\sqrt{\frac{(U^\top X)^+_{kn} + \left[(U^\top U)^- Z\right]_{kn}}{(U^\top X)^-_{kn} + \left[(U^\top U)^+ Z\right]_{kn}}}$$

With $a^+_{ij} := \max(0,a_{ij})$, $a^-_{ij} := \min(0,a_{ij})$.
Equivalent to K-means if Z is orthogonal.

5 Sparse Coding

Decompose original signal z into orthonormal matrix A and sparse signal $\hat{x}$ by truncating small values in x for $z = Ax$.

Overcompleteness ($L > D$): more atoms (dictionary elements) than dimensions. Therefore union of orthonormal bases $[U_1 \dots U_B]$ form new $U \in \mathbb{R}^{D\times(B\cdot D)} \Rightarrow$ Overcompleteness factor $= \frac{L}{D}$. Increases the linear dependency between atoms which is measured by **coherence**:
$$m(U) = \max_{i,j:i\neq j} \left|u_i^\top u_j\right|$$
which is 0 for orthogonal basis B and $\geq \frac{1}{\sqrt{D}}$ if atom u is added to B.

5.1 Matching Pursuit (MP) Algorithm

Minimize residual while selecting less than K atoms from the dictionary:
$$z^* = \arg\min_z \|x - Uz\|_2 \quad \text{s.t. } \|z\|_0 \leq K$$

MP-Algorithm:
$z \leftarrow 0$, $r \leftarrow z$ ▷ Initialization
while $\|z\|_0 < K$ **do**
▷ atom with maximum correlation to r
$d^* \leftarrow \arg\max_d |u_d^\top r|$
$z_{d^*} \leftarrow z_{d^*} + u_{u^*}^\top r$ ▷ update vector
$r \leftarrow r - (u_{d^*}^\top r)u_{d^*}$ ▷ update residual
end while

Exact recovery if $K < \frac{1}{2}\left(1 + \frac{1}{m(U)}\right)$. If coherence $m(U)$ small, explaining a generating atom with other atoms is not sparse. Therefore, sparse coding recovers support.

5.2 Sparse Coding for Inpainting

Sparse coding on known parts of the image which allows predition of missing parts by reconstruction from sparse code. Mask M with $m_{d,d} = 1$ if pixel d is known and 0 if missing.

$$z^* = \min_z \|z\|_0$$
$$\text{s.t.} \quad \|M(x - Uz)\|_2 < \sigma$$

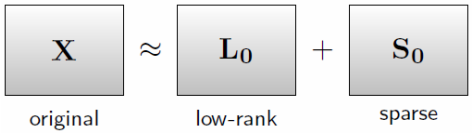
Reconstruction: $\hat{x} = Mx + (I - M)Uz^*$

5.3 Dictonary Learning

1. Coding: $Z^{t+1} \in \arg\min_z \|X - U^t \cdot Z\|_F^2$
2. Update: $U^{t+1} \in \arg\min_U \|X - U \cdot Z^{t+1}\|_F^2$

6 Robust PCA

Additive decomposition; minimize $rank(L) + \lambda \cdot card(S)$



Convex Relaxation ($\|\cdot\|_*$ nuclear norm; $\|\cdot\|_1$: sum of all absolute values):
$$\text{minimize} \quad \|L\|_* + \lambda\|S\|_1$$
$$\text{subject to} \quad L + S = X$$

6.1 Convexity

A set C is convex if the line segment between any two points in C lies in C. Function:
f is convex $:\Leftrightarrow \quad \forall x,y \in dom f, 0 \leq \theta \leq 1 :$
$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y)$$

Convex Optimization: minimize $f(x)$ s.t. $g_i(x) \leq 0, h_i(x) = 0$ with $f(x)$ convex objective function, $g_i(x)$ inequality constraint functions, $h_i(x)$ affine equality contrsint functions ($h_i(x) = a_i^\top x - b_i$)

Dual Problem: maximize $d(\lambda,\nu)$ s.t. $\lambda \geq 0$, (Lagrangian, Lagrange dual function)
$$L(x,\lambda,\nu) = f(x) + \sum_{i=1}^m \lambda_i g_i(x) + \sum_{i=1}^p \nu_i h_i(x)$$
$$d(\lambda,\nu) = \inf_x L(x,\lambda,\nu)$$

Dual Decomposition:
$x_i^{k+1} := \arg\min_{x_i} L_i(x_i,\nu^k), \quad \forall i$
$\nu^{k+1} := \nu^k + \alpha^k\left(\sum_{i=i}^N A_i x_i^{k+1} - b\right)$

Alternating Direction Method of Multipliers (ADMM) Minimize $f(x) + p(z) s.t. Ax + Bz = c$
$x^{k+1} := \arg\min_x L_\rho(x, z^k, \nu^k)$
$z^{k+1} := \arg\min_z L_\rho(x^{k+1}, z, \nu^k)$
$\nu^{k+1} := \nu^k + \rho(Ax^{k+1} + Bz^{k+1} - c)$

Robust PCA Solved with Principal Component Pursuit (PCP). Exact recovery with probability $1 - \mathcal{O}(n^{-10})$, PCP with $\lambda = \frac{1}{\sqrt{N}}$ is exact. L_0 of rank $\leq \rho_r n \mu^{-1}(\log n)^{-2}$ and S_0 of cardinality $m \leq \rho_s n^2$ (ρ_s, ρ_r const.)