0 Essentials

Matrix/Vector

Vec Spaces: $range(\mathbf{A}) = C(\mathbf{A}) = \{\mathbf{z} \mid \exists \mathbf{x} : \mathbf{z} = \mathbf{A}\mathbf{x}\}$
$rank(\mathbf{A}) = dim(C(\mathbf{A})) = dim(R(\mathbf{A}))$
$N(\mathbf{A}) = \{\mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{0}\}$
$N(\mathbf{A}) \perp R(\mathbf{A}),\ N(\mathbf{A}^\top) \perp C(\mathbf{A})$

Rank-null Thrm: $rank(\mathbf{A}) + dim(N(\mathbf{A})) = cols$

Orthog Matrix: $\mathbf{Q}^{-1} = \mathbf{Q}^\top, \mathbf{Q}\mathbf{Q}^\top = \mathbf{Q}^\top\mathbf{Q} = \mathbf{I}$ preserves inner product, norm, distance, angle, rank, matrix orthogonality

Dot Product: $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^\top \mathbf{y} = \sum_{i=1}^N x_i y_i = \|\mathbf{x}\|\|\mathbf{y}\|\cos\theta$. • $\langle \mathbf{x}+\mathbf{y}, \mathbf{z} \rangle = \langle \mathbf{x}, \mathbf{z} \rangle + \langle \mathbf{y}, \mathbf{z} \rangle$
• $\langle \mathbf{x} \pm \mathbf{y}, \mathbf{x} \pm \mathbf{y} \rangle = \langle \mathbf{x}, \mathbf{x} \rangle \pm 2\langle \mathbf{x}, \mathbf{y} \rangle + \langle \mathbf{y}, \mathbf{y} \rangle$
• $\langle \mathbf{u}, \mathbf{v} \rangle \mathbf{v} = (\mathbf{v}\mathbf{v}^\top)\mathbf{u}$

Outer Product: $\mathbf{u}\mathbf{v}^\top$, $(\mathbf{u}\mathbf{v}^\top)_{i,j} = \mathbf{u}_i \mathbf{v}_j$

Trace: $trace(\mathbf{XYZ}) = trace(\mathbf{ZXY})$

Transpose: $(\mathbf{A}^\top)^{-1} = (\mathbf{A}^{-1})^\top$, $(\mathbf{AB})^\top = \mathbf{B}^\top \mathbf{A}^\top$

Norms

- $\|\mathbf{x}\|_0 = |\{x_i \mid x_i \neq 0\}|$ # non-zero elems
- $\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^N x_i^2} = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$
- $\|\mathbf{u} - \mathbf{v}\|_2 = \sqrt{(\mathbf{u}-\mathbf{v})^\top(\mathbf{u}-\mathbf{v})}$
- $\|\mathbf{x}\|_p = \left(\sum_{i=1}^N |x_i|^p\right)^{\frac{1}{p}}$
- $\|\mathbf{A}\|_F = \sqrt{\sum\sum a_{i,j}^2} = \sqrt{trace(\mathbf{A}^T\mathbf{A})} = \sqrt{\sum_{i=1}^{\min\{m,n\}} \sigma_i^2} = \|\sigma(\mathbf{A})\|_2$
- $\|\mathbf{A}\|_G = \sqrt{\sum_{ij} g_{ij} a_{ij}^2}$ (weighted Frobenius)
- $\|\mathbf{A}\|_1 = \sum_{i,j} |a_{i,j}|$
- $\|\mathbf{A}\|_2 = \sigma_{\max}(\mathbf{A}) = \|\sigma(\mathbf{A})\|_\infty$
- $\|\mathbf{A}\|_p = \max \|\mathbf{A}\mathbf{v}\|_p : \|\mathbf{v}\|_p = 1$ max stretch
- $\|\mathbf{A}\|_* = \sum_{i=1}^{\min(m,n)} \sigma_i = \|\sigma(\mathbf{A})\|_1$ (nuclear)

Derivatives

• $\frac{\partial}{\partial \mathbf{x}}(\mathbf{b}^\top \mathbf{x}) = \frac{\partial}{\partial \mathbf{x}}(\mathbf{x}^\top \mathbf{b}) = \mathbf{b}$ • $\frac{\partial}{\partial \mathbf{x}}(\mathbf{x}^\top \mathbf{x}) = 2\mathbf{x}$
• $\frac{\partial}{\partial \mathbf{x}}(\mathbf{x}^\top \mathbf{A}\mathbf{x}) = (\mathbf{A}^\top + \mathbf{A})\mathbf{x}$ • $\frac{\partial}{\partial \mathbf{x}}(\mathbf{b}^\top \mathbf{A}\mathbf{x}) = \mathbf{A}^\top \mathbf{b}$
• $\frac{\partial}{\partial \mathbf{X}}(\mathbf{c}^\top \mathbf{X}\mathbf{b}) = \mathbf{c}\mathbf{b}^\top$ • $\frac{\partial}{\partial \mathbf{X}}(\mathbf{c}^\top \mathbf{X}^\top \mathbf{b}) = \mathbf{b}\mathbf{c}^\top$
• $\frac{\partial}{\partial \mathbf{x}}(\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2) = 2\mathbf{A}^\mathbf{T}(\mathbf{A}\mathbf{x} - \mathbf{b})$ • $\frac{\partial}{\partial \mathbf{x}}(\|\mathbf{x} - \mathbf{b}\|_2) = \frac{\mathbf{x}-\mathbf{b}}{\|\mathbf{x}-\mathbf{b}\|_2}$ • $\frac{\partial}{\partial \mathbf{x}}(\|\mathbf{x}\|_2^2) = \frac{\partial}{\partial \mathbf{x}}(\mathbf{x}^\top \mathbf{x}) = 2\mathbf{x}$ • $\frac{\partial}{\partial \mathbf{X}}(\|\mathbf{X}\|_F^2) = 2\mathbf{X}$ • $\frac{d}{dx}\log(x) = \frac{1}{x}$

Probability / Statistics

• $P(x) := Pr[X = x] := \sum_y P(x,y)$ • $P(x,y) = P(x|y)P(y)$ • $\forall y \in Y : \sum_x P(x|y) = 1$
• $P(x|y) = \frac{P(y|x)P(x)}{P(y)}$ (Bayes' rule)
• $P(x_1, \ldots, x_n) = \prod_{i=1}^n P(x_i)$ (iff i.i.d.) • $E[X] := \sum_{x \in X} p(x)x$ • $Var[X] := E[(X - \mu_x)^2] := E(X^2) - E(X)^2$ • $Var[aX + bY] := a^2 Var[X] + b^2 Var[Y] + 2ab\,Cov[X,Y]$

Eigendecomposition

Full rank $\mathbf{A} = \mathbf{X}\mathbf{\Lambda}\mathbf{X}^{-1}$ with $\mathbf{A}, \mathbf{X} \in \mathbb{R}^{N \times N}$.
Symmetric $\mathbf{S} = \mathbf{S}^\top = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top$ ($\mathbf{Q}$ orthogonal).

Singular Value Decomposition

$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^\top + .. + \sigma_r \mathbf{u}_r \mathbf{v}_r^\top$ for $r = rank(\mathbf{A})$
$\mathbf{A} \in \mathbb{R}^{N \times P}, \mathbf{U} \in \mathbb{R}^{N \times N}, \Sigma \in \mathbb{R}^{N \times P}, \mathbf{V} \in \mathbb{R}^{P \times P}$
$\mathbf{U}^\top \mathbf{U} = I = \mathbf{V}^\top \mathbf{V}$ ($\mathbf{U}, \mathbf{V}$ orthonormal)
$\mathbf{U}$ cols eigenvectors of $\mathbf{A}\mathbf{A}^\top$, $\mathbf{V}$ cols eigenvectors of $\mathbf{A}^\top\mathbf{A}$, $\Sigma = diag(\sigma_i)$ singular values.
1. find $\mathbf{A}^\top\mathbf{A}$.
2. find eigenvalues solving $det(\mathbf{A}^\top\mathbf{A} - \lambda\mathbf{I}) = 0$, sort them in descending order to get singular values $\sigma_i = \sqrt{\lambda_i}$ and set $\Sigma = diag(\sigma_i)$.
3. find eigenvectors of $\mathbf{A}^\top\mathbf{A}$ from the eigenvalues, and set them as columns of $\mathbf{V}$.
4. calculate the missing matrix: $\mathbf{U} = \mathbf{A}\mathbf{V}\Sigma^{-1}$.
5. normalize each column of $\mathbf{U}$ and $\mathbf{V}$.

1 Linear Autoencoder

Encoder $C \in \mathbb{R}^{K,N}$, decoder $D \in \mathbb{R}^{N,K}$ to approx identity map $\mathbf{x} \approx \mathbf{D}\mathbf{C}\mathbf{x}$ for $K << N$.

Echart-Young Theorem

Best rank-k approx (non-convex domain) of $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top$ given by k largest eigenvectors
$$\arg\min_{rank(B)=k} \|\mathbf{A} - \mathbf{B}\|_F^2 = \mathbf{A_k} = \mathbf{U}\Sigma_\mathbf{k}\mathbf{V}^\top$$
$\|A - A_k\|_F^2 = \sum_{r=k+1}^{rank(A)} \sigma_r^2, \quad \|A - A_k\|_2 = \sigma_{k+1}$

Optimal Linear Autoencoder via SVD

By EY thrm, optimal reconstruction of $\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^\top$ given by $C^* = \mathbf{A}\mathbf{U}_k^\top$, $D^* = \mathbf{U}_k\mathbf{A}^{-1}$ for any invertible $\mathbf{A}$. To reduce d.o.f. and ambiguity, do **weight sharing** and pick $\mathbf{D} = \mathbf{C}^\top$.

2 Principal Component Analysis

$\mathbf{X} \in \mathbb{R}^{D \times N}$. N observations, K rank.
1. Empirical Mean: $\overline{\mathbf{x}} = \frac{1}{N}\sum_{n=1}^N \mathbf{x}_n$.
2. Center Data: $\overline{\mathbf{X}} = \mathbf{X} - [\overline{\mathbf{x}}, \ldots, \overline{\mathbf{x}}] = \mathbf{X} - \mathbf{M}$.
3. Cov.: $\Sigma = \frac{1}{N}\sum_{n=1}^N (\mathbf{x}_n - \overline{\mathbf{x}})(\mathbf{x}_n - \overline{\mathbf{x}})^\top = \frac{1}{N}\overline{\mathbf{X}}\overline{\mathbf{X}}^\top$.
4. Eigenvalue Decomposition: $\Sigma = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$.
5. Select $K < D$, only keep $\mathbf{U}_K, \lambda_K$.
6. Transform data onto new Basis: $\overline{\mathbf{Z}}_K = \mathbf{U}_K^\top \overline{\mathbf{X}}$.
7. Reconstruct to original Basis: $\tilde{\overline{\mathbf{X}}} = \mathbf{U}_k \overline{\mathbf{Z}}_K$.
8. Reverse centering: $\tilde{\mathbf{X}} = \tilde{\overline{\mathbf{X}}} + \mathbf{M}$.
For compression save $\mathbf{U}_k, \overline{\mathbf{Z}}_K, \overline{\mathbf{x}}$.
$\mathbf{U}_k \in \mathbb{R}^{D \times K}, \Sigma \in \mathbb{R}^{D \times D}, \overline{\mathbf{Z}}_K \in \mathbb{R}^{K \times N}, \overline{\mathbf{X}} \in \mathbb{R}^{D \times N}$

Iterative View

Residual $\mathbf{r_i} : \mathbf{x_i} - \tilde{\mathbf{x}}_\mathbf{i} = (\mathbf{I} - \mathbf{u}\mathbf{u}^\top)\mathbf{x_i}$
Cov of r: $\frac{1}{n}\sum_{i=1}^n (I - uu^T)x_i x_i^T (I - uu^T)^T = (I - uu^T)\Sigma(I - uu^T)^T = \Sigma - 2\Sigma uu^T + uu^T\Sigma uu^T = \Sigma - \lambda uu^T$
1. Find principal eigenvector of $(\Sigma - \lambda uu^T)$
2. which is the second eigenvector of Σ
3. iterating to get d principal eigenvector of Σ

Power Method

Power iteration: $v_{t+1} = \frac{Av_t}{\|Av_t\|}$, $\lim_{t \to \infty} v_t = u_1$
Assuming $\langle u_1, v_0 \rangle \neq 0$ and $|\lambda_1| > |\lambda_j| (\forall j \geq 2)$

Comparison with Linear Autoencoder

PCA clarifies importance of data centering, gives unique repr. (except for eigenvectors with same λ_i). Linear autoencoder spans same space but gives arbitrary basis (often not interpretable). PCA with power iter easy and robust, good for small k (finds one component at a time). PCA via SVD good for mid-sized problems. For large datasets, autoencoder trainable via backprop: extensible and SGD efficient.

3 Matrix Approximation & Reconstruction

$\min_{rank(B)=k}[\sum_{(i,j)\in I}(a_{ij} - b_{ij})^2], I = \{(i,j) : obs.\}$

Alternating Least Squares

$f(U, v_i) = \sum_{(i,j)\in I}(a_{i,j} - \langle u_j, v_i \rangle)^2$
$f(u_i, V) = \sum_{(i,j)\in I}(a_{i,j} - \langle u_j, v_i \rangle)^2$
Convex when fixed one.

Convex Optimization

$f : \mathbb{R}^D \to \mathbb{R}$ is convex, if $dom\ f$ is a convex set (i.e. points on the line between any two points are also in the set), and if $\forall \mathbf{x}, \mathbf{y} \in dom\ f$, $\forall \alpha \in [0,1]$: $f(\alpha\mathbf{x} + (1-\alpha)\mathbf{y}) \leq \alpha f(\mathbf{x}) + (1-\alpha)f(\mathbf{y})$.
Convex $\iff$ local=global

Convex Relaxation

Replace non-convex rank constraints by convex norm constraints (superset). Then project optimum back (hopefully still optimal).
$\min_{\mathbf{B} \in P_k} \|\mathbf{A} - \mathbf{B}\|_G^2, P_k = \{\mathbf{B} : \|\mathbf{B}\|_* \leq k\} \supseteq Q_k = \{\mathbf{B} : rank(\mathbf{B}) \leq k\}$ (as tightest convex lower-bound $rank(\mathbf{B}) \geq \|\mathbf{B}\|_* : \|\mathbf{B}\|_2 \leq 1$)

SVD Thresholding

$\mathbf{B}^* = shrink_\tau(\mathbf{A}) = \arg\min_\mathbf{B} \{\|\mathbf{A} - \mathbf{B}\|_F^2 + \tau\|\mathbf{B}\|_*\}$
Then with SVD $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V_T}, \mathbf{D} = diag(\sigma_i)$, holds $\mathbf{B}^* = \mathbf{U}\mathbf{D}_\tau\mathbf{V}^\mathbf{T}, \mathbf{D}_\tau = diag(\max\{0, \sigma_i - \tau\})$
Iteration: $\mathbf{B}_{t+1} = \mathbf{B}_t + \eta_t \Pi(\mathbf{A} - shrink_\tau(\mathbf{B}_t))$

4 Non-Negative Matrix Factorization

$\mathbf{X} \in \mathbb{Z}_{\geq 0}^{N \times M}$, NMF: $\mathbf{X} \approx \mathbf{U}^\top\mathbf{V}, x_{ij} = \sum_z u_{zi} v_{zj} = \langle \mathbf{u}_i \mathbf{v}_j \rangle$ Decompose object into features: topics, face parts, etc.. $\mathbf{u}$ weights on parts, $\mathbf{v}$ parts (bases). More interpretable (PCA: holistic repre.).

Jensen's ineq. concave : $\varphi\left(\frac{\sum_i a_i x_i}{\sum_i a_i}\right) \geq \frac{\sum_i a_i \varphi(x_i)}{\sum_i a_i}$

EM for MLE for pLSA (NO global opt guarantee)

Context Model: $p(w|d) = \sum_{z=1}^K p(w|z)p(z|d)$
Conditional independence assumption $(*)$: words depend on topics only, not on docs
$p(w|d) = \sum_z p(w|d,z)p(z|d) \stackrel{*}{=} \sum_z p(w|z)p(z|d)$
Symmetric parametrization:
$p(w,d) = \sum_z p(z)p(w|z)p(d|z)$
Log-Likelihood: $L(\mathbf{U}, \mathbf{V}) = \sum_{i,j} x_{i,j} \log p(w_j|d_i)$
$= \sum_{(i,j)\in X} \log \sum_{z=1}^K p(w_j|z)p(z|d_i)$ (concave)
$p(w_j|z) = v_{zj}$, $p(z|d_i) = u_{zi}$, $\sum_j^N v_{zj} = \sum_z^K u_{zi} = 1$
E-Step (optimal q: posterior of z over (d_i, w_j)):
$q_{zij} = \frac{p(w_j|z)p(z|d_i)}{\sum_{k=1}^K p(w_j|k)p(k|d_i)} := \frac{v_{zj}u_{zi}}{\sum_{k=1}^K v_{kj}u_{ki}}$, $\sum_z q_{zij} = 1$
M-Steps:
$p(z|d_i) = \frac{\sum_j x_{ij} q_{zij}}{\sum_j x_{ij}}$, $p(w_j|z) = \frac{\sum_i x_{ij} q_{zij}}{\sum_{i,l} x_{il} q_{zil}}$

Latent Dirichlet Allocation

To sample a new document, we need to extend X and U^T with a new row, s.t. $X = U^T V$. (While pLSA fixes both dimensions)
For each d_i sample topic weights
$\mathbf{u}_i \sim$Dirichlet(α): $p(u_i|\alpha) = \prod_{z=1}^K u_{zi}^{\alpha_k - 1}$, then topic $z^t \sim$Multi(u_i), word $w^t \sim$Multi(v_{z^t})
Multinom. obsv. model on word-count vec:
$p(\mathbf{x}|V, u) = \frac{l!}{\prod_j \mathbf{x}_j!}\prod_j \pi_j^{\mathbf{x}_j}$ where $\pi_j = \sum_z v_{zj} u_z$, $l = \sum_j x_j$
Bayesian averaging over $\mathbf{u}$: $p(\mathbf{x}|\mathbf{V}, \alpha) = \int p(\mathbf{x}|\mathbf{V}, \mathbf{u})p(\mathbf{u}|\alpha)d\mathbf{u}$

NMF Algorithm for quadratic cost function

$\min_{\mathbf{U},\mathbf{V}} J(\mathbf{U}, \mathbf{V}) = \frac{1}{2}\|\mathbf{X} - \mathbf{U}^\top\mathbf{V}\|_F^2$ (non-negativity) s.t. $\forall i, j, z : u_{zi}, v_{zj} \geq 0$
Comparison with pLSA:
1. sampling model: Gaussian vs multinomial 2. objective: quadratic vs KL divergence 3. constraints: not normalized
Alternating least squares:
1. init: $\mathbf{U}, \mathbf{V} = rand()$
2. repeat 3~4 for $maxIters$:
3. upd. $(\mathbf{V}\mathbf{V}^\top)\mathbf{U} = \mathbf{V}\mathbf{X}^\top$, proj. $u_{zi} = \max\{0, u_{zi}\}$
4. update $(\mathbf{U}\mathbf{U}^\top)\mathbf{V} = \mathbf{U}\mathbf{X}$, proj. $v_{zj} = \max\{0, v_{zj}\}$

5 Word Embeddings

Distributional Model:
$p_\theta(w|w') = $ Pr[w occurs in context of w']
Log-likelihood:
$L(\theta; \mathbf{w}) = \sum_{t=1}^T \sum_{\Delta \in I} \log p_\theta(w^{(t+\Delta)}|w^{(t)})$
Latent Vector Model: $w \to (\mathbf{x}_w, b_w) \in \mathbb{R}^{D+1}$
$p_\theta(w|w') = \frac{\exp[\langle \mathbf{x}_w, \mathbf{x}_{w'} \rangle + b_w]}{\sum_{v \in V}\exp[\langle \mathbf{x}_v, \mathbf{x}_{w'} \rangle + b_v]}$ (soft-max).
Modifications:
$\log p_\theta(w|w') = \langle y_w, x_{w'} \rangle + b_w$, word y_w, ctx $x_{w'}$
use GloVe objective
negative sampling (logistic classification)

GloVe (Weighted Square Loss)

Co-oc Matrix: $(n_{ij}) \in \mathbb{R}^{|V| \times |C|} = w_i$ # in ctx w_j
Objective: $\mathcal{H}(\theta; \mathbf{N}) = $
$= \sum_{n_{ij}>0} f(n_{ij})[\log n_{ij} - \log\exp[\langle \mathbf{x}_i, \mathbf{y}_j \rangle + b_i + c_j]]^2$
with $f(n) = \min\{1, (\frac{n}{n_{max}})^\alpha\}$, $\alpha \in (0;1]$.

unnormalized distr. → 2-sided loss function (no norm., fast)
1. sample (i,j) unif. rand. such that $n_{ij} > 0$
2. $\mathbf{x}_i^{new} \leftarrow \mathbf{x}_i + 2\eta f(n_{ij})(\log n_{ij} - \langle \mathbf{x}_i, \mathbf{y}_j \rangle)\mathbf{y}_j$
3. $\mathbf{y}_j^{new} \leftarrow \mathbf{y}_j + 2\eta f(n_{ij})(\log n_{ij} - \langle \mathbf{x}_i, \mathbf{y}_j \rangle)\mathbf{x}_i$

Discussion
Word embeddings can model analogies and relatedness, but antonyms are usually not well captured.

6 Data Clustering & Mixture Models

KMeans

Target: $\min_{\mathbf{U},\mathbf{Z}} J(\mathbf{U},\mathbf{Z}) = \|\mathbf{X} - \mathbf{U}\mathbf{Z}\|_F^2$ $= \sum_{n=1}^N \sum_{k=1}^K \mathbf{z}_{k,n} \|\mathbf{x}_n - \mathbf{u}_k\|_2^2$
1. **Initiate:** choose K centroids $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_K]$
2. **Cluster Assign:** data points to clusters. $k^\star(\mathbf{x}_n) = \arg\min_k \{\|\mathbf{x}_n - \mathbf{u}_k\|_2\}$ returns cluster $k^\star$, whose centroid $\mathbf{u}_{k^\star}$ is closest to data point $\mathbf{x}_n$. Set $\mathbf{z}_{k^\star,n} = 1$, and for $l \neq k^\star$ $\mathbf{z}_{l,n} = 0$.
3. **Update centroids:** $\mathbf{u}_k = \frac{\sum_{n=1}^N z_{k,n}\mathbf{x}_n}{\sum_{n=1}^N z_{k,n}}$.
4. Repeat from step 2, stops if $\|\mathbf{Z} - \mathbf{Z}^{\text{new}}\|_0 = \|\mathbf{Z} - \mathbf{Z}^{\text{new}}\|_F^2 = 0$.

Computational cost: $O(k \cdot n \cdot d)$

Gaussian Mixture Models (GMM)

Gaussian $p(x) = \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{(x-\mu)^2}{2\sigma^2})$ Multivariate $p(x;\mu;\Sigma) = \frac{1}{|\Sigma|^{\frac{1}{2}}(2\pi)^{\frac{D}{2}}} exp[-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)]$

For GMM let $\boldsymbol{\theta}_k = (\boldsymbol{\mu}_k, \Sigma_k)$; $p_{\theta_k}(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k)$, $\boldsymbol{\theta} = (\pi, \theta_1, .., \theta_k)$

Mixture Models: $p_\theta(\mathbf{x}) = \sum_{k=1}^K \pi_k p_{\theta_k}(\mathbf{x})$

Assignment variable (generative model): $z_{ij} \in \{0,1\}$, $\sum_{j=1}^k z_{ij} = 1$
$\Pr(z_k = 1) = \pi_k \Leftrightarrow p(\mathbf{z}) = \prod_{k=1}^K \pi_k^{z_k}$

Complete data distribution: $p_\theta(\mathbf{x},\mathbf{z}) = \prod_{k=1}^K \left(\pi_k p_{\theta_k}(\mathbf{x})\right)^{z_k}$

Posterior Probabilities: $\Pr(z_k = 1|\mathbf{x}) = \frac{\Pr(z_k=1)p(\mathbf{x}|z_k=1)}{\sum_{l=1}^K \Pr(z_l=1)p(\mathbf{x}|z_l=1)} = \frac{\pi_k p_{\theta_k}(\mathbf{x})}{\sum_{l=1}^K \pi_l p_{\theta_l}(\mathbf{x})}$

post $p(A|B) = \frac{\text{prior } p(A) \times \text{likelihood } p(B|A)}{\text{evidence } p(B)}$

Likelihood of observed data X: $p_\theta(\mathbf{X}) = \prod_{n=1}^N p_\theta(\mathbf{x}_n) = \prod_{n=1}^N \left(\sum_{k=1}^K \pi_k p_{\theta_k}(\mathbf{x}_n)\right)$

Max. Likelihood Estimation (MLE): $\arg\max_\theta \sum_{n=1}^N \log\left(\sum_{k=1}^K \pi_k p_{\theta_k}(\mathbf{x}_n)\right)$
$\geq \sum_{n=1^N} \sum_{k=1}^K q_k[\log p_{\theta_k}(\mathbf{x}_n) + \log \pi_k - \log q_k]$
with $\sum_{k=1}^K q_k = 1$ by Jensen Inequality.

Generative Model
1. sample cluster index $j \sim Categorical(\pi)$
2. given j, sample data $x \sim \text{Normal}(\mu_j, \Sigma_j)$

Expectation-Maximization (EM) for GMM
E-Step: $\Pr[z_{k,n} = 1|\mathbf{x}_n] = q_{k,n} = \frac{\pi_k^{(t-1)}\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k^{(t-1)}, \Sigma_k^{(t-1)})}{\sum_{j=1}^K \pi_j^{(t-1)}\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j^{(t-1)}, \Sigma_j^{(t-1)})}$

M-Step: $\boldsymbol{\mu}_k^{(t)} := \frac{\sum_{n=1}^N q_{k,n}\mathbf{x}_n}{\sum_{n=1}^N q_{k,n}}$, $\pi_k^{(t)} := \frac{1}{N}\sum_{n=1}^N q_{k,n}$
$\Sigma_k^{(t)} = \frac{\sum_{n=1}^N q_{k,n}(\mathbf{x}_n - \boldsymbol{\mu}_k^{(t)})(\mathbf{x}_n - \boldsymbol{\mu}_k^{(t)})^\top}{\sum_{n=1}^N q_{k,n}}$

Discussion K-means vs. EM
hard assignment vs soft. spherical clusters shapes vs covariance matrix. fast vs slow and more iteration. K-means can be used as initialization for EM.
K-means as a special case of GMM with covariances $\Sigma_j = \sigma^2 I$. in the limit of $\sigma \to 0$, recover K-means (hard assignments).

Model Order Selection (AIC / BIC for GMM)
Trade-off between data fit (i.e. likelihood $p(\mathbf{X}|\theta)$) and complexity (i.e. # of free parameters $\kappa(\cdot)$). For choosing K:
Akaike Information Criterion: $\text{AIC}(\theta|\mathbf{X}) = -\log p_\theta(\mathbf{X}) + \kappa(\theta)$
Bayesian Information Criterion: $\text{BIC}(\theta|\mathbf{X}) = -\log p_\theta(\mathbf{X}) + \frac{1}{2}\kappa(\theta)\log N$
# of free params, fixed covariance matrix: $\kappa(\theta) = K \cdot D + (K-1)$ (K: # clusters, D: $\dim(\text{data}) = \dim(\mu_i)$, $K-1$: π of # free clusters), full covariance matrix: $\kappa(\theta) = K(D + \frac{D(D+1)}{2}) + (K-1)$.
Compare AIC/BIC for different K – the smaller the better. BIC penalizes complexity more.

7 Neural Networks

Activation: $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ sigmoid $s(x) = \frac{1}{1+e^{-x}}$, $s'(x) = s(x)(1-s(x))$, ReLU $\max(0,x)$

Neurons: $F_\sigma(\mathbf{x};\mathbf{w}) = \sigma(w_0 + \sum_{i=1}^M x_i w_i)$.

Output: linear regression $\mathbf{y} = \mathbf{W}^L\mathbf{x}^{L-1}$, binary (logistic) $y_1 = \text{P}[Y=1|\mathbf{x}] = \frac{1}{1+\exp(-\mathbf{w}^T\mathbf{x}^{L-1})}$, multiclass (soft-max) $y_k = \text{P}[Y=k|\mathbf{x}] = \frac{\exp(\mathbf{w}_k^T\mathbf{x}^{L-1})}{\sum_{m=1}^K \exp(\mathbf{w}^T\mathbf{x}^{L-1})}$. **Loss function** $l(y,\hat{y})$: squared loss $\frac{1}{2}(y-\hat{y})^2$, cross-entropy loss $-y\log\hat{y} - (1-y)\log(1-\hat{y})$. **Units and Layers**: layer-to-layer fwd. prop. notation: $\mathbf{x}^l = \sigma^l\left(\mathbb{W}^{(l)}\mathbf{x}^{(l-1)}\right)$. L-layer network: $\mathbf{y} = \sigma^{(L)}\left(\mathbf{W}^(L)\sigma^{(L-1)}\left(\cdots\left(\sigma^{(1)}\left(\mathbf{W}^{(1)}\mathbf{x}\right)\cdots\right)\right)\right)$

Backpropagation
Layer-to-layer Jacobian: $\mathbf{x}$ = prev. layer activation, $\mathbf{x}^+$ = next layer activation. Jacobian matrix $\mathbf{J} = J_{ij}$ of mapping $\mathbf{x} \to \mathbf{x}^+$, $\mathbf{x}_i^+ = \sigma(\mathbf{w}_i^\top\mathbf{x})$, $J_{ij} = \frac{\partial \mathbf{x}_i^+}{\partial \mathbf{x}_j} = w_{ij} \cdot \sigma'(\mathbf{w}_i^\top\mathbf{x})$. Across multiple layers: $\frac{\partial \mathbf{x}^{(l)}}{\partial \mathbf{x}^{(l-n)}} = \mathbf{J}^{(l)} \cdot \frac{\partial \mathbf{x}^{(l-1)}}{\partial \mathbf{x}^{(l-n)}} = \mathbf{J}^{(l)} \cdot \mathbf{J}^{(l-1)} \cdots \mathbf{J}^{(l-n+1)}$ and then back prop. $\nabla_{\mathbf{x}^{(l)}}^\top \ell = \nabla_{\mathbf{y}}^\top \ell \cdot \mathbf{J}^{(L)} \cdots \mathbf{J}^{(l+1)}$

Weights: $\frac{\partial l}{\partial w_{ij}^{(l)}} = \frac{\partial l}{\partial x_i^{(l)}} \frac{\partial x_i^{(l)}}{\partial w_{ij}^{(l)}}$, $\frac{\partial x_i^l}{\partial w_{ij}^l} = \sigma'([\mathbf{w}_i^{(l)}]^T\mathbf{x}^{(l-1)}) \cdot x_j^{(l-1)}$ (sensitivity of down-stream unit · activation of up-stream unit)

Gradient Descent (or Deepest Descent)
Gradient: $\nabla f(\mathbf{x}) := \left(\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}_1}, \ldots, \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}_D}\right)^\top$
1. init: $\mathbf{x}^{(0)} \in \mathbb{R}^D$
2. for $t = 0$ to $maxIter$:
3. $\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \gamma\nabla f(\mathbf{x}^{(t)})$, usually $\gamma \approx \frac{1}{t}$

Stochastic Gradient Descent (SGD)
Assume **Additive Objective**: $f(x) = \frac{1}{N}\sum_{n=1}^N f_n(x)$
1. init: $\mathbf{x}^{(0)} \in \mathbb{R}^D$
2. for $t = 0$ to $maxIter$:
3. sample $n \in_{u.a.r.} \{1, \ldots, N\}$
4. $\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \gamma\nabla f_n(\mathbf{x}^{(t)})$, typically $\gamma \approx \frac{1}{t}$.

Neural Networks for Images (CNN)
Translation invariance of images → neurons compute same fct, shift invariant filters; weights defined as filter masks, e.g. convolution: $F_{n,m}(\mathbf{x};\mathbf{w}) = \sigma(b + \sum_{k=-2}^2\sum_{l=-2}^2 w_{k,l}x_{n+k,m+l})$. Use {max, avg}-pooling to reduce dim. of convolution and extract interesting features.

8 Generative Models

Autoregressive
Image $p(\mathbf{x}) = \Pi_i^{n^2} p(x_i|x_1, \cdots, x_{i-1})$

Variational Autoencoder
$D_{KL}(P\|Q) = \sum_i P(i)\log\frac{P(i)}{Q(i)} = \mathbb{E}_i[\frac{\log P_i}{\log Q_i}]$ (0: similar)
Elbo $\log p_\theta(x) \geq \mathbb{E}z \sim Q[\log P\theta(x|z)] - D^{KL}(Q(z|x)\|P(z))$
Q enc. posterior distr., $P(z)$ prior distr. on latent var z, P_g likelihood of dec. generated $\mathbf{x}$
Jointly trained: enc. optimize regularizer term, sample $\mathbf{z} \sim Q$, feed to dec., produce $\hat{x}$ to max. reconstruction quality. Both terms diff'able, can use SGD to train end-to-end.

Generative Adversarial Networks
Optimal Bayes classifier: posterior $q_\theta = p/(p + p_\theta)$ (often inaccessible) Generator vs classifier (to fool)

9 Sparse Coding

Orthogonal Basis
Pros: fast inverse; preserves energy. For $\mathbf{x}$ and orthog. mat. $\mathbf{U}$ compute $\mathbf{z} = \mathbf{U}^\top\mathbf{x}$. Approx $\hat{\mathbf{x}} = \mathbf{U}\hat{\mathbf{z}}$, $\hat{z}_i = z_i$ if $|z_i| > \epsilon$ else 0. Reconstruction Error $\|\mathbf{x} - \hat{\mathbf{x}}\|^2 = \sum_{d \notin \sigma}\langle \mathbf{x}, \mathbf{u}_d\rangle^2$. **Choice of base** depends on signal. Fourier for global, wavelet for local support. PCA basis optimal for given Σ. Stripes & check patterns: hi-freq in Fourier.

Haar Wavelets (form orthogonal basis)
scaling fcn $\phi(x) = [1,1,1,1]$, mother $W(x) = [1,1,-1,-1]$, dilated $W(2x) = [1,-1,0,0]$, translated $W(2x-1) = [0,0,1,-1]$

Overcomplete Basis
$\mathbf{U} \in \mathbb{R}^{D \times L}$ for # atoms $= L > D = \dim(\text{data})$. Decoding involved → add constraint $\mathbf{z}^\star \in \arg\min_{\mathbf{z}} \|\mathbf{z}\|_0$ s.t. $\mathbf{x} = \mathbf{U}\mathbf{z}$. NP-hard → approximate with 1-norm (convex) or with MP.
Coherence • $m(\mathbf{U}) = \max_{i,j:i\neq j} |\mathbf{u}_i^\top\mathbf{u}_j|$ • $m(\mathbf{B}) = 0$ if $\mathbf{B}$ orthogonal matrix • $m([\mathbf{B},\mathbf{u}]) \geq \frac{1}{\sqrt{D}}$ if atom $\mathbf{u}$ is added to orthogonal basis $\mathbf{B}$ (o.n.b. = orthonormal base)

Matching Pursuit (MP) approximation of $\mathbf{x}$ onto $\mathbf{U}$, using K entries. Objective: $\mathbf{z}^\star \in \arg\min_{\mathbf{z}} \|\mathbf{x} - \mathbf{U}\mathbf{z}\|_2$, s.t. $\|\mathbf{z}\|_0 \leq K$ 1. init: $z \leftarrow 0, r \leftarrow x$ 2. while $\|\mathbf{z}\|_0 < K$ do 3. select atom with smallest angle $i^\star = \arg\max_i |\langle \mathbf{u}_i, \mathbf{r}\rangle|$ 4. update coefficients: $z_{i^\star} \leftarrow z_{i^\star} + \langle \mathbf{u}_{i^\star}, \mathbf{r}\rangle$ 5. update residual: $\mathbf{r} \leftarrow \mathbf{r} - \langle \mathbf{u}_{i^\star}, \mathbf{r}\rangle\mathbf{u}_{i^\star}$.
Exact recovery when: $K < 1/2(1 + 1/m(\mathbf{U}))$
Compressive Sensing: Compress data while gathering: • $\mathbf{x} \in \mathbb{R}^D$, K-sparse in o.n.b. $\mathbf{U}$. $\mathbf{y} \in \mathbb{R}^M$ with $y_i = \langle \mathbf{w}_i, \mathbf{x}\rangle$: M lin. combinations of signal; $\mathbf{y} = \mathbf{W}\mathbf{x} = \mathbf{W}\mathbf{U}\mathbf{z} = \boldsymbol{\Theta}\mathbf{z}$, $\Theta \in \mathbb{R}^{M \times D}$ • Reconstruct $\mathbf{x} \in \mathbb{R}^D$ from $\mathbf{y}$; find $\mathbf{z}^\star \in \arg\min_{\mathbf{z}} \|\mathbf{z}\|_0$, s.t. $\mathbf{y} = \boldsymbol{\Theta}\mathbf{z}$ (e.g. with MP, or convex it with 1-norm: canbe eq!). Given $\mathbf{z}$, reconstruct $\mathbf{x} = \mathbf{U}\mathbf{z}$
Any orthogonal $\mathbf{U}$ sufficient if: • $\mathbf{W}$ = Gaussian random projection, i.e. $w_{ij} \sim \mathcal{N}(0, \frac{1}{D})$ • M $\geq cKlog(\frac{D}{K})$, where c is some constant

10 Dictionary Learning

Adapt the dictionary to signal characteristics. Objective: $(\mathbf{U}^\star, \mathbf{Z}^\star) \in \arg\min_{\mathbf{U},\mathbf{Z}} \|\mathbf{X} - \mathbf{U} \cdot \mathbf{Z}\|_F^2$ not jointly convex but convex in 1 argument.

Matrix Factorization by Iter Greedy Minimization 1. Coding step: $\mathbf{Z}^{t+1} \in \arg\min_{\mathbf{Z}} \|\mathbf{X} - \mathbf{U}^t\mathbf{Z}\|_F^2$ subject to $\mathbf{Z}$ being sparse ($\mathbf{z}_n^{t+1} \in \arg\min_{\mathbf{z}} \|\mathbf{z}\|_0$ s.t.$\|\mathbf{x}_n - \mathbf{U}^t\mathbf{z}\|_2 \leq \sigma\|\mathbf{x}_n\|_2$)
2. Dict update step: $\mathbf{U}^{t+1} \in \arg\min_{\mathbf{U}} \|\mathbf{X} - \mathbf{U}\mathbf{Z}^{t+1}\|_F^2$, subj to $\forall l \in [L] : \|\mathbf{u}_l\|_2 = 1$. (set $\mathbf{U} = [\mathbf{u}_1^t \cdots \mathbf{u}_l \cdots \mathbf{u}_L^t]$, $\min_{u_l} \|\mathbf{X} - \mathbf{U}\mathbf{Z}^{t+1}\|_F^2 = \min_{u_l} \|\mathbf{R}_l^t - \mathbf{u}_l(\mathbf{z}_l^{t+1})^\top\|_F^2$ with $\mathbf{R}_l^t = \tilde{\mathbf{U}}\Sigma\tilde{\mathbf{V}}^\top$ by $\mathbf{u}_l^* = \tilde{\mathbf{u}}_1$)