1 Essentials

Derivatives

$\frac{\partial}{\partial \mathbf{x}}(\mathbf{a}^\top \mathbf{x}) = \mathbf{a}$ $\quad \frac{\partial}{\partial \mathbf{x}}(\mathbf{A}\mathbf{x}) = \mathbf{A}^\top$ $\quad \frac{\partial}{\partial \mathbf{x}}(\mathbf{x}^\top \mathbf{A}) = \mathbf{A}$

$\frac{\partial}{\partial \mathbf{x}}(\mathbf{x}^\top \mathbf{x}) = 2\mathbf{x}$ $\quad \frac{\partial}{\partial \mathbf{x}}(\mathbf{x}^\top \mathbf{A}\mathbf{x}) = (\mathbf{A}+\mathbf{A}^\top)\mathbf{x}$

$\frac{\partial}{\partial \mathbf{X}}(\mathbf{c}^\top \mathbf{X}\mathbf{b}) = \mathbf{c}\mathbf{b}^\top$ $\quad \frac{\partial}{\partial \mathbf{X}}(\mathbf{c}^\top \mathbf{X}^\top \mathbf{b}) = \mathbf{b}\mathbf{c}^\top$

$\frac{\partial}{\partial \mathbf{X}}(\|\mathbf{X}\|_F^2) = 2\mathbf{X}$ $\quad \frac{\partial}{\partial \mathbf{X}}(log(det(\mathbf{X}))) = (\mathbf{X}^\top)^{-1}$

$\frac{\partial}{\partial \mathbf{a}}((\mathbf{x}-\mathbf{a})^\top \mathbf{W}(\mathbf{x}-\mathbf{a})) = -2\mathbf{W}(\mathbf{x}-\mathbf{a})$

$\frac{\partial}{\partial \mathbf{X}}(\mathbf{a}^\top \mathbf{X}^{-1}\mathbf{b}) = -(\mathbf{X}^\top)^{-1}\mathbf{a}\mathbf{b}^\top(\mathbf{X}^\top)^{-1}$

$\frac{\partial}{\partial \mathbf{X}}(Tr(\mathbf{A}\mathbf{X})) = \frac{\partial}{\partial \mathbf{X}}(Tr(\mathbf{X}\mathbf{A})) = \mathbf{A}^\top$

$\frac{\partial}{\partial \mathbf{X}}(Tr(\mathbf{A}\mathbf{X}^\top)) = \frac{\partial}{\partial \mathbf{X}}(Tr(\mathbf{X}^\top\mathbf{A})) = \mathbf{A}$

$\frac{\partial}{\partial \mathbf{X}}(Tr(\mathbf{X}^\top \mathbf{A}\mathbf{X})) = (\mathbf{A}+\mathbf{A}^\top)\mathbf{X}$

Jacobian

$J_F = \begin{bmatrix} \nabla_x^\top F_1 \\ \dots \\ \nabla_x^\top F_m \end{bmatrix}$ where $F : \mathbb{R}^n \to \mathbb{R}^m$ for $x \in \mathbb{R}^n$

LinAlg

- A is **psd** if $v^\top A v \geq 0$, note: if $A = B^\top B$ then A is psd, psd $\Leftrightarrow \lambda \geq 0$
- **orthogonal matrix**: $U^\top U = UU^\top = \mathbb{1}$
1.unit length columns/rows 2.orthogonal columns/rows 3. square matrix
properties: preserve norm $\|Ux\|_2^2 = \|x\|_2^2$, $det(U) = \pm 1$, full rank because invertible.
- $\langle \mathbf{x}, \mathbf{y} \rangle = \|\mathbf{x}\|_2 \cdot \|\mathbf{y}\|_2 \cdot \cos(\theta)$
- **spectral theorem**: Matrix A is diagonalizable by orthogonal matrix iff A is symmetric.
- A is degenerate/non-invertible $\Leftrightarrow det(A) = 0$.
- **rank-nullity theorem**: $dim(kernel(A)) + dim(range(A)) = n$ where n is input dimension.
- **determinant**: $det(diag(a_1, ..., a_n)) = a_1 * ... * a_n$
$det(\begin{bmatrix} a & b \\ c & d \end{bmatrix}) = ad - bc$
- **cauchy-schwarz**: $x^\top y \leq \|x\|_2 \|y\|_2$
- **trace** of a matrix is equal to the sum of its eigenvalues. Trace operator is linear and cyclic.
- eigenvectors corresponding to distinct eigenvalues of a symmetric matrix are orthogonal to each other.
- a function is convex iff its Hessian ∇^2 is psd.
- f convex $\Leftrightarrow \forall \lambda \in [0,1], x, y \in X. f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y)$
- **jensens inequality**: $E[f(X)] \geq f(E[X])$ for f convex ($\leq$ for f concave).

Eigendecomposition

$\Sigma = U \Lambda U^\top$ where $\Lambda = diag(\lambda_1, ..., \lambda_m)$ with eigenvalues $\lambda_1 \geq ... \geq \lambda_m$, $U = (u_1 | ... | u_m)$ eigenvectors orthogonal. Only exists if Σ is symmetric.

Eigenvalue / -vectors

Eigenvalue Problem: $\mathbf{A}\mathbf{x} = \lambda \mathbf{x}$

1. solve $\det(\mathbf{A} - \lambda \mathbb{1}) \stackrel{!}{=} 0$ resulting in $\{\lambda_i\}_i$
2. $\forall \lambda_i$: solve $(\mathbf{A} - \lambda_i \mathbb{1})\mathbf{x}_i = \mathbf{0}$, for $\mathbf{x}_i$.

Norms

$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n \mathbf{a}_{i,j}^2} = \sqrt{trace(A^\top A)} = \sqrt{trace(AA^\top)} = \sqrt{\sum_{i=1}^{\min\{m,n\}} \sigma_i^2}$

$\|\mathbf{A}\|_2 = sup\{\|Ax\| : \|x\| = 1\} = \sigma_1$

$\|\mathbf{M}\|_\star = \sum_{i=1}^{\min(m,n)} \sigma_i$

Probability / Statistics

- $Var(X) = E[(X-\mu)^2]$
- sample variance: $\frac{1}{N}\sum_{i=1}^N (x_i - \overline{x})^2$

2 Principle Component Analysis

$\mathbf{X} \in \mathbb{R}^{D \times N}$. N observations.

1. Empirical Mean: $\overline{\mathbf{x}} = \frac{1}{N}\sum_{n=1}^N \mathbf{x}_n$.
2. Center Data: $\overline{\mathbf{X}} = \mathbf{X} - [\overline{\mathbf{x}}, \dots, \overline{\mathbf{x}}] = \mathbf{X} - \mathbf{M}$.
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 $\mathbf{U}_K = [u_1, ..., u_K]$.
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 $\mathbf{U}_k, \overline{\mathbf{Z}}_K, \overline{\mathbf{x}}$.
$\mathbf{U}_k \in \mathbb{R}^{D \times K}, \Sigma \in \mathbb{R}^{D \times D}, \overline{\mathbf{Z}}_K \in \mathbb{R}^{K \times N}, \overline{\mathbf{X}} \in \mathbb{R}^{D \times N}$

- Reconstruction error = sum of discarded eigenvalues.
- The transformed dataset $\overline{\mathbf{Z}}_K$ has diagonal covariance matrix.

Iterative View

Residuals $r_i = x_i - \tilde{x}_i = (\mathbb{1} - uu^T)x_i$

Cov. matrix of residuals: $\frac{1}{n}\sum_{i=1}^n r_i r_i^\top = \frac{1}{n}\sum_{i=1}^n (\mathbb{1} - uu^T) x_i x_i^T (\mathbb{1} - uu^T)^T = ... = \Sigma - \lambda uu^T$

1. Find principal eigenvector of $(\Sigma - \lambda uu^T)$
2. which is the 2nd principal eigenvector of Σ
3. iterating to get d principal eigenvector of Σ

- the principal eigenvector of Σ points in the direction with largest variance of the data ($argmax_{\|u\|=1} u^\top \Sigma u$).

Power iteration

find principal eigenvector of A: $v_{t+1} = \frac{Av_t}{\|Av_t\|}$, $\lim_{t \to \infty} v_t = u_1$

3 SVD

$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^\top = \sum_{i=1}^{s=min\{m,n\}} \sigma_i u_i (v_i)^\top$

$\mathbf{A} \in \mathbb{R}^{m \times n}, \mathbf{U} \in \mathbb{R}^{m \times m}, \mathbf{D} \in \mathbb{R}^{m \times n}, \mathbf{V} \in \mathbb{R}^{n \times n}$

$\mathbf{U}, \mathbf{V}$ orthogonal, $D = diag(\sigma_1, ..., \sigma_s), \sigma_1 \geq ... \geq \sigma_s \geq 0$. Rank($\mathbf{A}$) =#non-zero singular values.

Eckart-Young Theorem

$\min_{rank(B)=K} \|A - B\|_F^2 = \|A - A_k\|_F^2 = \sum_{r=k+1}^{rank(A)} \sigma_r^2$

note: use $Tr(A^\top A)$ def of F-norm to prove.

$\min_{rank(B)=K} \|A - B\|_2 = \|A - A_k\|_2 = \sigma_{k+1}$

4 Matrix Approximation & Reconstruction

problem of SVD is unobserved entries. Want to only consider I = observed entries.

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

Convex Relaxation

- $rank(B) \geq \|B\|_*$ for $\|B\|_2 \leq 1$. proof: $rank(B) = \#\{\sigma_i > 0\} = \sum_{i:\sigma_i > 0} 1 \geq \sum_{i:\sigma_i > 0} \sigma_i = \|B\|_*$ since $\sigma_1 = \|B\|_2 \leq 1$. Thus $Q_k = \{B : rank(B) \leq k\} \subseteq P_k = \{B : \|B\|_* \leq k\}$ a convex relaxation (in fact convex hull). •Singular Value Tresholding: if $A = UDV^\top$ then $shrink_\tau(A) = UD_\tau V^\top$ where $D_\tau = diag(max\{0, \sigma_i - \tau\})$. $B_{t+1} = B_t + \eta_t \pi(A - shrink_\tau(B_t))$ where π zeros out unobserved entries.

Alternating Least Squares

reparametrize $B = UV$ where $U \in \mathbb{R}^{mxk}, V \in \mathbb{R}^{kxn}$. If we have product of two matrices, rank cannot be bigger than smallest rank occuring in it. $min[\sum_{(i,j)\in I}(a_{ij} - u_i^\top v_j)^2]$. ALS = optimize over u_i's while keeping v_j's fixed and vice versa.

5 Non-Negative Matrix Factorization

pLSA

•co-occurence matrix $X = x_{ij}$ #occurences of word w_j in doc d_i. •$p(w|d) = \sum_z p(w, z|d) = \sum_z p(w|d,z)p(z|d) = \sum_z p(w|z)p(z|d)$.
•log-likelihood: $\sum_{i,j} x_{i,j} \log p(w_j|d_i)$

E-Step (optimal q):

$q_{zij} = p(z|w_j, d_i) = \frac{p(w_j|z)p(z|d_i)}{\sum_{k=1}^K p(w_j|k)p(k|d_i)}$

M-Steps:

$p(z|d_i) = \frac{\sum_j x_{ij} q_{zij}}{\sum_j x_{ij}}, p(w_j|z) = \frac{\sum_i x_{ij} q_{zij}}{\sum_{i,l} x_{il} q_{zil}}$

NMF Algorithm for quadratic cost function

$\mathbf{X} \in \mathbb{Z}_{\geq 0}^{N \times M}$, NMF: $\mathbf{X} \approx \mathbf{U}^\top \mathbf{V}, x_{ij}$

$\min_{\mathbf{U},\mathbf{V}} J(\mathbf{U}, \mathbf{V}) = \frac{1}{2}\|\mathbf{X} - \mathbf{U}^\top \mathbf{V}\|_F^2 = \frac{1}{2}\sum(x_{ij} - u_i^\top v_j)^2$

s.t. $\forall i, j, z : u_{zi}, v_{zj} \geq 0$

1. init: $\mathbf{U}, \mathbf{V} = rand()$ 2. repeat for $maxIters$:
3. upd. $(\mathbf{V}\mathbf{V}^\top)\mathbf{U} = \mathbf{V}\mathbf{X}^\top$, proj. $u_{zi} = \max\{0, u_{zi}\}$
4. upd. $(\mathbf{U}\mathbf{U}^\top)\mathbf{V} = \mathbf{U}\mathbf{X}$, proj. $v_{zj} = \max\{0, v_{zj}\}$

•vector form:
$(\sum_i u_i u_i^\top) v_j = \sum_i x_{ij} u_i$, $(\sum_j v_j v_j^\top) u_i = \sum_j x_{ij} v_j$

note for derivation: If want matrix form, use trace def. If want vector only form, use sum representation of objective.

6 Word Embeddings

Skip-gram 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).

- **add context embeddings:** more flexibility $\log p_\theta(w|w') = \langle x_{w'}, y_w \rangle + b_w$, word embeddings y_w, context embeddings $x_{w'}$.
- **negative sampling (logistic classification):** avoids having to compute normalization Z $\sum_{(i,j)\in\Delta^+} \log \sigma(x_i^\top y_j) + \sum_{(i,j)\in\Delta^-} \log \sigma(-x_i^\top y_j >)$

GloVe

Co-occurence Matrix:
$\mathbf{N} = (n_{ij}) \in \mathbb{R}^{|V| \times |C|}$ = # of word w_i in context w_j

Objective: (Weighted Square Loss) $H(\theta; \mathbf{N})$
$= \sum_{n_{ij} > 0} f(n_{ij})(\log n_{ij} - \log \tilde{p}_\theta(w_i|w_j))^2$
with $\tilde{p}_\theta(w_i|w_j) = \exp[\langle \mathbf{x}_i, \mathbf{y}_j \rangle + b_i + c_j]$ unnormalized! and $f(n) = \min\{1, (\frac{n}{n_{max}})^\alpha\}, \alpha \in (0;1]$.

1. sample (i,j) u.a.r, s.t. $n_{ij} > 0$
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. $\mathbf{y}_j^{new} \leftarrow \mathbf{y}_j + 2\eta f(n_{ij})(\log n_{ij} - \langle \mathbf{x}_i, \mathbf{y}_j \rangle)\mathbf{x}_i$

7 K-means Algorithm

Hard assignments $Z \in \{0,1\}^{NxK}$, Centroids $U = (u_1|..|u_K) \in \mathbb{R}^{DxK}$, data $X \in \mathbb{R}^{DxN}$

Target: $\min_{\mathbf{U},\mathbf{Z}} J(\mathbf{U}, \mathbf{Z}) = \|\mathbf{X} - \mathbf{U}\mathbf{Z}^\top\|_F^2 = \sum_{n=1}^N \sum_{k=1}^K \mathbf{z}_{n,k} \|\mathbf{x}_n - \mathbf{u}_k\|_2^2$

1. **Assign** data points to closest centroid: $k^*(\mathbf{x}_n) = \arg\min_k \{\|\mathbf{x}_n - \mathbf{u}_k\|_2\}$. Set $\mathbf{z}_{k^*,n} = 1$, and for $l \neq k^*$ $\mathbf{z}_{l,n} = 0$.
2. **Update** centroids: $\mathbf{u}_k = \frac{\sum_{n=1}^N z_{n,k}\mathbf{x}_n}{\sum_{n=1}^N z_{n,k}}$.
3. Repeat until Z doesn't change anymore.

Computational cost: $O(K \cdot N \cdot D)$

8 Gaussian Mixture Models (GMM)

let $\boldsymbol{\theta}_k = (\boldsymbol{\mu}_k, \Sigma_k)$, $p(\mathbf{x}; \theta_k) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k)$

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

1. sample cluster index $j \sim Categorical(\pi)$
2. given j, sample data $x \sim$ Normal(μ_j, Σ_j)

Latent variables: data point x_i belongs to cluster z_i. $p(z_i = j) = \pi_j$

Max. Likelihood Estimation (MLE):
$\arg\max_\theta \sum_{n=1}^N \log\left(\sum_{k=1}^K \pi_k p(\mathbf{x}_n; \theta_k)\right)$
$\geq \sum_{n=1}^N \sum_{k=1}^K q_{k,n}[\log p(\mathbf{x}_n; \theta_k) + \log \pi_k - \log q_{k,n}]$

Expectation-Maximization (EM) for GMM

E-Step: (posterior over latent variables)
$q_{k,n}^* = \Pr(z_n = k|x_n) = \frac{1}{Z}p(z_n = k)p(x_n|z_n = k) =$

$\frac{\pi_k p(x_n;\theta_k)}{\sum_{l=1}^{K}\pi_l p(x_n;\theta_l)}$

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

Gaussian distribution

Standard deviation σ, Mean μ

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

Covariance matrix Σ, Mean μ

$p(x;\mu,\Sigma) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}$

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: $\mathrm{AIC}(\theta|\mathbf{X}) = -\log p_\theta(\mathbf{X}) + \kappa(\theta)$
Bayesian Information Criterion: $\mathrm{BIC}(\theta|\mathbf{X}) = -\log p_\theta(\mathbf{X}) + \frac{1}{2}\kappa(\theta)\log N$
# of free params for GMM: $\kappa(\theta) = KD + K\frac{D(D+1)}{2} + (K-1)$.

9 Sparse Coding

Orthogonal Basis

For x and orthogonal $U = (u_1|..|u_D) \in \mathbb{R}^{DxD}$. Encoding $\mathbf{z} = \mathbf{U}^\top\mathbf{x}$. Decoding $\mathbf{x} = \mathbf{U}\mathbf{z} = \mathbf{U}\mathbf{U}^\top\mathbf{x} = \mathbf{x}$. Tresholding $\hat{\mathbf{x}} = \mathbf{U}\hat{\mathbf{z}}$, $\hat{z}_i = z_i$ if $|z_i| > \epsilon$ else 0. Reconstr. Error $\|\mathbf{x}-\hat{\mathbf{x}}\|^2 = \|\sum_{d\notin\sigma}\langle\mathbf{x},\mathbf{u}_d\rangle\mathbf{u}_d\|^2 = \sum_{d\notin\sigma}\langle\mathbf{x},\mathbf{u}_d\rangle^2$. Proof: $\|\sum x_i\|^2 = \sum\|x_i\|^2$ if x_i orthogonal. Advantages: efficient inverse, energy/length preservation.

Haar wavelets:

scaling: $[1,1,1,1]$, mother wavelet: $[1,1,-1,-1]$, dilated: $[1,-1,0,0]$, translated: $[0,0,1,-1]$.

Fourier vs Wavelets:

•Fourier: good for periodic signals, global support, no time info/only frequencies themselves.
•Wavelets: good for localized signals like abrupt changes/irregularities, represents a signal in time and frequency domain.

Overcomplete Basis

$\mathbf{U} \in \mathbb{R}^{D\times L}$ for # atoms $= L > D = \dim(\text{data})$. $x \in \mathbb{R}^D, z \in \mathbb{R}^L$ because $Uz = x$. Encoding ill-posed problem $\rightarrow$ add constraint $\mathbf{z}^\star \in \arg\min_\mathbf{z}\|\mathbf{z}\|_0$ s.t. $\mathbf{x} = \mathbf{U}\mathbf{z}$. NP-hard and non-convex $\rightarrow$ approx with Basis pursuit i.e relax to 1-norm (convex) or with Matching Pursuit.

Coherence

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

Matching Pursuit (MP) a greedy approximation algorithm. 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 max correlation $d^\star = \arg\max_d|\langle\mathbf{u}_d,\mathbf{r}\rangle|$ 4. update coefficients: $z_{d^\star} \leftarrow z_{d^\star} + \langle\mathbf{u}_{d^\star},\mathbf{r}\rangle$ 5. update residual: $\mathbf{r} \leftarrow \mathbf{r} - \langle\mathbf{u}_{d^\star},\mathbf{r}\rangle\mathbf{u}_{d^\star}$.

Compressive Sensing: Compress data while gathering: • $\mathbf{x} \in \mathbb{R}^D$, K-sparse in o.n.b. $\mathbf{U}$. $\mathbf{y} \in \mathbb{R}^M$ corresponds to M lin. combinations/measurements of signal; $\mathbf{y} = \mathbf{W}\mathbf{x} = \mathbf{W}\mathbf{U}\mathbf{z} = \theta\mathbf{z}$, $\theta \in \mathbb{R}^{M\times D}$ • Reconstruct $\mathbf{x} \in \mathbb{R}^D$ from $\mathbf{y}$; find $\mathbf{z}^\star \in \arg\min_\mathbf{z}\|\mathbf{z}\|_0$, s.t. $\mathbf{y} = \theta\mathbf{z}$ (e.g. with MP). Given $\mathbf{z}$, reconstruct $\mathbf{x}$ via $\mathbf{x} = \mathbf{U}\mathbf{z}$
Sufficient conditions: • $\mathbf{W}$ = Gaussian random projection, i.e. $w_{ij} \sim \mathcal{N}(0,\frac{1}{D})$ • $\mathrm{M} \geq cKlog(\frac{D}{K})$, where c is some constant

Dictionary Learning

Adapt the dictionary to signal characteristics. Objective: $(\mathbf{U}^\star,\mathbf{Z}^\star) \in \arg\min_{\mathbf{U},\mathbf{Z}}\|\mathbf{X}-\mathbf{U}\cdot\mathbf{Z}\|_F^2$

K-SVD (Iter Greedy Minimization): 1. Coding step: $\mathbf{Z}^{t+1} \in \arg\min_\mathbf{Z}\|\mathbf{X}-\mathbf{U}^t\mathbf{Z}\|_F^2 = \sum_{i=1}^{N}\|x_i - U^t z_i\|^2$ s.t $\|z_i\|_0 \leq K$. Use any pursuit algorithm. 2. Dict update step: $\mathbf{U}^{t+1} \in \arg\min_\mathbf{U}\|\mathbf{X}-\mathbf{U}\mathbf{Z}^{t+1}\|_F^2$, s.t $\forall l \in [L] : \|\mathbf{u}_l\|_2 = 1$. idea: update one atom u_l at a time. $\min_{u_l}\|\mathbf{X}-\mathbf{U}\mathbf{Z}\|_F^2 = \min_{u_l}\|X - (\sum_{e\neq l}u_e z_e^\top + u_l z_l^\top)\|_F^2 = \min_{u_l}\|\mathbf{R}_l - \mathbf{u}_l(\mathbf{z}_l)^\top\|_F^2$ where z_l is the l-th row of matrix Z and R_l is the residual due to atom u_l. Use SVD $\mathbf{R}_l = \sum_i\sigma_i\tilde{u}_i\tilde{v}_i^\top$ then $\mathbf{u}_l^* = \tilde{\mathbf{u}}_1$ and $\mathbf{z}_l^* = \tilde{\mathbf{v}}_1$ (use power iteration for efficiency).

10 Neural Networks

$F^\sigma : \mathbb{R}^n \rightarrow \mathbb{R}^m, F_j^\sigma(x) = \sigma(w_j^\top x)$ for $j = 1,..,m$

Activation fuctions: logistic function $\sigma(x) = \frac{1}{1+e^{-x}}$, $\sigma'(x) = \sigma(x)(1-\sigma(x))$, ReLu $\phi(z) = max(0,z)$

Output layer: linear regression: $\hat{\mathbf{y}} = \mathbf{W}^L\mathbf{x}^{L-1}$
binary classification (logistic):
$\hat{y_1} = \mathrm{P}[Y=1|\mathbf{x}] = \frac{1}{1+\exp[-\langle\mathbf{w}_1^L,\mathbf{x}^{L-1}\rangle]}$
multiclass (soft-max):
$\hat{y_k} = \mathrm{P}[Y=k|\mathbf{x}] = \frac{\exp[\langle\mathbf{w}_k^L,\mathbf{x}^{L-1}\rangle]}{\sum_{m=1}^{K}\exp[\langle\mathbf{w}_m^L,\mathbf{x}^{L-1}\rangle]}$.

Loss function squared loss: $\frac{1}{2}(y-\hat{y})^2$
cross-entropy loss: $-y\log\hat{y} - (1-y)\log(1-\hat{y})$.
Regularization: add l_2-regularizer to objective or add drop-out layers.
Units and Layers: layer-to-layer fwd. prop. notation: $\mathbf{x}^{(l)} = \sigma^{(l)}\left(\mathbb{W}^{(l)}\mathbf{x}^{(l-1)}\right)$ where $y = x^{(L)}$ is the output activation vector.

Backpropagation

•Use SGD to optimize over weights: $\theta \leftarrow \theta - \eta\nabla_\theta l(y_t; y(x_t;\theta))$ for $t = \{1,..,T\}$ •We want to know $\partial l/\partial w_{ij}^{(l')}$ i.e how does changing weights affect the loss. •three steps: 1. how does output y affect loss 2. how do activities of units affect each other resp. y. 3. how do weights affect activities of units. •1. $\nabla_y l = \frac{\partial}{\partial y}l(y^*,y) = ..$
2. $\frac{\partial\mathbf{x}^{(l)}}{\partial\mathbf{x}^{(l-n)}} = \mathbf{J}^{(l)}\cdot\frac{\partial\mathbf{x}^{(l-1)}}{\partial\mathbf{x}^{(l-n)}} = \mathbf{J}^{(l)}\cdot\mathbf{J}^{(l-1)}\cdots\mathbf{J}^{(l-n+1)}$ where $\mathbf{x}$ = prev. layer activation, $\mathbf{x}^+$ = next layer activation. Jacobian matrix $\mathbf{J} = J_{ij}$ of mapping $\mathbf{x} \rightarrow \mathbf{x}^+$, $\mathbf{x}_i^+ = \sigma(\mathbf{w}_i^\top\mathbf{x})$, $J_{ij} = \frac{\partial\mathbf{x}_i^+}{\partial\mathbf{x}_j} = w_{ij}\cdot\sigma'(\mathbf{w}_i^\top\mathbf{x})$.
3. $\frac{\partial x_i^+}{\partial w_{ij}} = \sigma'(w_i^\top x)x_j$

•Perform forward pass to compute activities for all units. Compute gradient of objective wrt output layer activites. Propagate gradient info back from output to inputs. Compute local gradients of activities wrt weights.

Convolutional Neural Networks

•**Convolution step**: primary purpose is to extract features from the input image. Parameter/weight sharing = a kernel is used on multiple locations of the image with the same weights. Sparse interactions = by making kernel smaller than input. Discrete convolution operator $s[i,j] = (I * K)[i,j] = \sum_m\sum_n I[m,n]K[i-m,j-n]$ where I is the image and K the kernel. Note that arguments are commutative.
•**Pooling step**: reduce dim of each feature map e.g by max, sum, average over a predefined spatial neighborhood. Why? Scale invariant representation of image, less params/less overfitting.

11 Deep Unsupervised Learning

Autoencoders

learn low-dim representation $z \in \mathbb{R}^d$ for given data. Linear autoencoder with weights $C \in \mathbb{R}^{dxm}$ (encoder) and $D \in \mathbb{R}^{mxd}$ (decoder). objective min $\frac{1}{2n}\sum_{i=1}^{n}\|x_i - DCx_i\|^2$. Frobenius norm optimal approx (in this case) via SVD $X^\top X = U\Sigma V^\top$ then $C^* = U_d^\top$ and $D^* = U_d$ (first d columns of U).

Variational Autoencoders (VAEs)

Put a gaussian prior on distribution of continous latent vector z: $p(z_l) \sim \mathcal{N}(\mu_l,\Sigma_l)$ for $l \in \{1,..L\}$. This allows to easily generate new data points.
log-likelihood: $\log p_\theta(x) = \log\int p_\theta(x|z)p(z)dz \geq E_q[\log p_\theta(x|z)] + E_q[\log\frac{p(z)}{q(z)}]$ where $-D_{KL}(q\|p) = E_q[\log\frac{p(z)}{q(z)}]$. KL-divergence tells how much two distributions diverge. Optimal $q(z) = p(z|x)$.
NN approach: 1. recognition/inference model: learns variational distribution $q(z)$ i.e given x it returns params of normal distribution (μ_l,Σ_l) for $l = 1,..,L$ from which we than can sample the z_l's. 2. generative model: implements $p_\theta(x|z)$ and deterministically maps z to x (reconstruction).

Autoregressive Models

generate output one variable at a time based on chain rule $p(x_{1:m}) = \Pi_{i=1}^{m}p(x_t|x_{1:t-1})$.
PixelCNN: nxn image with pixels $x_1,..,x_{n^2}$. Generate pixel x_i by conditioning on previously generated pixels $x_1,..,x_{i-1}$. Use a masking filter for implem.
RNN:observed sequence $x_1,..,x_T$ and corresponding labels $y_1,..,y_T$. Use feedbackloop $h_t = f(h_{t-1},x_t)$ with hidden state h_t. LSTM units to avoid vanishing gradient problem.
PixelRNN: use RNN for mapping $x_1,..x_{i-1}$ to x_i. Row LSTM: convolute along each row from top to bottom (triangular receptive field i.e loss of context). Diagonal BiLSTM: convolute along the diagonal (receptive field includes all previously generated pixels).

12 Notes

•Proof that eigenvectors of symmetric matrix are orthogonal: $\lambda\langle x,y\rangle = \langle\lambda x,y\rangle = \langle Ax,y\rangle = \langle x,A^\top y\rangle = \langle x,Ay\rangle = \langle x,\mu y\rangle = \mu\langle x,y\rangle \Rightarrow (\lambda-\mu)\langle x,y\rangle = 0$ where $\lambda \neq \mu$.
•K-means vs GMM: k-means has hard assignments, cheaper to train (less params). GMM has soft assginments, more expressive because cluster is described by a MVN i.e shape is defined by an arbitrary covariance martix and not restricted to spherical clusters. GMM is generative model i.e we can do outlier detection, generate data points, uncertainty estimation.
•SVD and PCA: if $A = UDV^\top$. Columns of U are eigenvectors of $AA^\top$. Columns of V are eigenvectors of $A^\top A$. Eigenvalues of $A^\top A$ and $AA^\top$ are singular values^2 of A.
•orthogonal Haar Basis for 4-dim signals:

$$U = \frac{1}{2}\begin{bmatrix}1 & 1 & \sqrt{2} & 0\\ 1 & 1 & -\sqrt{2} & 0\\ 1 & -1 & 0 & \sqrt{2}\\ 1 & -1 & 0 & -\sqrt{2}\end{bmatrix}$$