1 Essentials

1.1 Matrix/Vector Derrivatives

• $\nabla_x(u\cdot v) = u\cdot\nabla_x(v) + v\cdot\nabla_x(u)$ • $\nabla_x(g(u)) = \nabla_u(g(u))\cdot\nabla_x(u)$ • $\nabla_x(a^Tx) = \nabla_x(x^Ta) = a$ • $\nabla_x(b^TAx) = A^Tb$ • $\nabla_x(x^TAx) = (A+A^T)x =_{\text{sym.}} 2Ax$ • $\nabla_x(a^Txx^Tb) = (ab^T+ba^T)x$ • $\nabla_x\|x\|^2 = 2x$ • $\nabla_x\|x\| = \frac{x}{\|x\|}$

1.2 Norms

l_0: $\|\mathbf{x}\|_0 := |\{i|x_i\neq 0\}|$ **Nuclear:** $\|\mathbf{X}\|_\star = \sum_{i=1}^{\min(m,n)}\sigma_i$ **p-norm:** $\|\mathbf{x}\|_p := \left(\sum_{i=1}^N |x_i|^p\right)^{\frac1p}$ **Frobenius:** $\|\mathbf{A}\|_F := \sqrt{\sum_{i,j}|\mathbf{A}_{i,j}|^2} = \sqrt{\sum_{i=1}^{\min\{m,n\}}\sigma_i^2}$ (σ_i is the i-th singularvalue)

1.3 Eigenvalue / -vectors

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

1. solve $\det(\mathbf{A}-\lambda\mathbf{I}) \overset{!}{=} 0$ resulting in $\{\lambda_i\}_i$
2. $\forall\lambda_i$: solve $(\mathbf{A}-\lambda_i\mathbf{I})\mathbf{x}_i = \mathbf{0}$, $\mathbf{x}_i$ is the i-th eigenvector.
3. (opt.) normalize eigenvector q_i: $q_i^{\text{norm}} = \frac{1}{\|q_i\|_2}q_i$.

1.4 Eigendecomposition

$\mathbf{A}\in\mathbb{R}^{N\times N}$, $\mathbf{A}=\mathbf{U}\Lambda\mathbf{U}^{-1}$, $\mathbf{Q}\in\mathbb{R}^{N\times N}$, $\Lambda = diag(\lambda_1,\dots,\lambda_n)$, $\mathbf{U}^T=\mathbf{U}^{-1}$, $\mathbf{A}$ symmetric then $\mathbf{A}^{-1}=\mathbf{U}\Lambda^{-1}\mathbf{U}^{-1}$

1.5 Probability / Statistics

• $P(x|y) := \frac{P(x,y)}{P(y)}$, if $P(y)>0$ • $\sum_{x\in X}P(x|y)=1$ • $P(x,y) = P(x|y)P(y)$ • $P(x|y) = \frac{P(y|x)P(x)}{P(y)}$ (Bayes' rule) • $P(x|y) = P(x) \Leftrightarrow P(y|x)=P(y)$ (iff X,Y independent) • $P(x_1,\dots,x_n) = \prod_{i=1}^n P(x_i)$ (iff IID)

2 Dimensionality Reduction / PCA

$\mathbf{X}\in\mathbb{R}^{D\times N}$. N observations, K properties. Target: $\tilde{\mathbf{X}}\in\mathbb{R}^{K\times N}$.

1. Empirical Mean: $\bar{\mathbf{x}} = \frac1N\sum_{n=1}^N\mathbf{x}_n$
2. Center Data: $\overline{\mathbf{X}} = \mathbf{X} - [\bar{\mathbf{x}},\dots,\bar{\mathbf{x}}] = \mathbf{X}-\mathbf{M}$
3. Cov. Matrix: $\Sigma = \frac1N\sum_{n=1}^N(\mathbf{x}_n-\bar{\mathbf{x}})(\mathbf{x}_n-\bar{\mathbf{x}})^\top = \frac1N\overline{\mathbf{X}}\overline{\mathbf{X}}^\top$
4. Eig.Dec.: $\Sigma = \mathbf{U}\Lambda\mathbf{U}^\top$
5. Select $K<D$, keep first K ew. and ev. $\Rightarrow \mathbf{U}_K,\lambda_K$
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 $\mathbf{U}_k,\overline{\mathbf{Z}}_K,\bar{\mathbf{x}}$.
• $\mathbf{U}_k\in\mathbb{R}^{D\times K},\Sigma\in\mathbb{R}^{D\times D},\overline{\mathbf{Z}}_K\in\mathbb{R}^{K\times N},\overline{\mathbf{X}}\in\mathbb{R}^{D\times N}$

3 SVD

• $\mathbf{A} = \mathbf{UDV}^\top = \sum_{k=1}^{\text{rank}(\mathbf{A})} d_{k,k}u_k(v_k)^\top$ • $\mathbf{A}\in\mathbb{R}^{N\times P},\mathbf{U}\in\mathbb{R}^{N\times N},\mathbf{D}\in\mathbb{R}^{N\times P},\mathbf{V}\in\mathbb{R}^{P\times P}$ • $\mathbf{U}^\top\mathbf{U} = \mathbf{V}^\top\mathbf{V} = \mathbf{I}$ (cols. orthonormal) • cols. of $\mathbf{U}$ are ev. of $\mathbf{AA}^\top$ (row simil. result of PCA of $\mathbf{A}$), $\mathbf{V}$ of $\mathbf{A}^\top\mathbf{A}$ (col. simil.), $\mathbf{D} = diag(\sigma_i)$, $\sigma_i^2=\lambda_i$ for u_i, v_i • cols. of $\mathbf{V}$ where $\sigma_i=0$ span $null(A)$, $\mathbf{U}$ where $\sigma_i>0$ span $range(A)$ • $\mathbf{A}$ sym. then $\mathbf{A}=\mathbf{UDU}^\top$ and u_i ev. of $\mathbf{A}$ • $\mathbf{A}_k = \sum_{i=1}^k u_i\sigma_iv_i^T$ • $min_{rank(B)=k}\|A-B\|_2 = \|A-A_k\|_2 = \sigma_{k+1}$ • $min_{rank(B)=k}\|A-B\|_F^2 = \|A-A_k\|_F^2 = \sigma_{k+1}^2+\dots+\sigma_r^2$

1. calculate $\mathbf{A}^\top\mathbf{A}$.
2. calculate eigenvalues of $\mathbf{A}^\top\mathbf{A}$, the square root of them, in descending order, are the diagonal elements of $\mathbf{D}$.
3. calculate eigenvectors of $\mathbf{A}^\top\mathbf{A}$ using the eigenvalues resulting in the columns of $\mathbf{V}$.
4. calculate the missing matrix: $\mathbf{U}=\mathbf{AVD}^{-1}$. Can be checked by calculating the eigenvectors of $\mathbf{AA}^\top$.
5. normalize each column of $\mathbf{U}$ and $\mathbf{V}$.

4 K-means Algorithm

Target: $\min_{\mathbf{U},\mathbf{Z}} J(\mathbf{U},\mathbf{Z}) = \sum_{n=1}^N\sum_{k=1}^K \mathbf{z}_{k,n}\|\mathbf{x}_n-\mathbf{u}_k\|_2^2 = \|\mathbf{X}-\mathbf{UZ}\|_F^2$ 1. $\mathbf{U} = [\mathbf{u}_1^{(0)},\dots,\mathbf{u}_k^{(0)}]$ 2. $k^\star(\mathbf{x}_n) = \arg\min_k\{\|\mathbf{x}_n-\mathbf{u}_k^{(t-1)}\|_2\}$ Set $\mathbf{z}_{j,n}^{(t)}=1$ if $j=k^\star$ else 0. 3. $\mathbf{u}_k^{(t)} = \frac{\sum_{n=1}^N z_{k,n}^{(t)}\mathbf{x}_n}{\sum_{n=1}^N z_{k,n}^{(t)}}$.

4. stops if $\|u_k^{(t-1)}-u_k^{(t)}\|<\varepsilon\ \forall k$.

4.1 Clustering Stability

mulitple runs return similar clusters • dist. between clust. (same data): $d(C,C') := \min_\Pi\frac12\|Z-\Pi(Z')\|_F^2$, $\Pi(Z')$ = row perm. of Z' • arbitrary sets $\mathbf{X},\mathbf{X}'$ of size N,N': $r := \frac{1}{N'}\min_\Pi\{\sum_{n=1}^{N'}\mathbb{I}_{\{\Pi(\phi(x_n'))\neq z_n'\}}\}$ (ϕ: multi-class classifier trained on $(\mathbf{X},\mathbf{Z})$) • for K clusters: stability $:= 1-\frac{r}{r_{rand}}$ (1 good, 0 bad), rand. clust. of equal size: $r_{rand} = \frac{K-1}{K}$.

5 Gaussian Mixture Models (GMM)

For GMM let $\theta_k = (\mu_k,\Sigma_k)$; $p_{\theta_k}(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\mu_k,\Sigma_k) = \frac{1}{\sqrt{(2\pi)^n|\Sigma_k|}}exp(-\frac12(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k))$

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

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

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

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

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

MLE: $\arg\max_\theta\sum_{n=1}^N\log\left(\sum_{k=1}^K\pi_kp_{\theta_k}(\mathbf{x}_n)\right)$

$\log\left(\sum_{k=1}^K q_k\frac{\pi_kp_{\theta_k}(\mathbf{x}_n)}{q_k}\right)\geq\sum_{k=1}^K q_k[\log p_{\theta_k}(\mathbf{x}_n)+\log\pi_k-\log q_k]$ with $\sum_{k=1}^K q_k=1$ by Jensens inequality

5.1 Expectation-Maximization (EM) for GMM

1. Initialize $\pi_k^{(0)},\mu_k^{(0)},\Sigma_k^{(0)}$ for $k=1,\dots,K$ and $t=1$.
2. E-Step: $q_{k,n}^\star = Pr[z_{k,n}=1|\mathbf{x}_n]$
3. M-Step: $\mu_k^\star := \frac{\sum_{n=1}^N q_{k,n}\mathbf{x}_n}{\sum_{n=1}^N q_{k,n}}$ & $\pi_k^\star := \frac1N\sum_{n=1}^N q_{k,n}$ & $\Sigma_k^\star = \frac{\sum_{n=1}^N q_{k,n}(\mathbf{x}_n-\mu_k)(\mathbf{x}_k-\mu_k)^\top}{\sum_{n=1}^N q_{k,n}}$
4. stop if $\|\log p_{\theta_{(t-1)}} - \log p_{\theta_{(t)}}\|<\varepsilon$

5.2 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**: $\text{BIC}(\theta|\mathbf{X}) = -\log p_\theta(\mathbf{X})+\frac12\kappa(\theta)\log N$ • # of free params: fixed covariance matrix: $\kappa(\theta) = K\cdot D+(K-1)$ (K: # clusters, D: dim(data) = dim(μ_i), $K-1$: # 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.

6 Word Embeddings

Distributional Model: $p_\theta(w|w') = \Pr[w \text{ occurs close to } w']$

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: $w\mapsto(\mathbf{x}_w,b_w)\in\mathbb{R}^{D+1}$

$p_\theta(w|w') = \frac{\exp[\langle\mathbf{x}_w,\mathbf{x}_{w'}\rangle+b_w]}{\sum_{v\in V}\exp[\langle\mathbf{x}_v,\mathbf{x}_{w'}\rangle+b_v]}$. Modifications: • split vocab in main vocab V, context vocab C: $p_\theta(w|w') = \langle x_{w'},y_w\rangle+b_w$, word embed. x_w, context embed. y_w • use GloVe objective

6.1 GloVe (Weighted Square Loss)

Co-occurence Matrix: $\mathbf{N} = (n_{ij})\in\mathbb{R}^{|V|\cdot|C|}\leftrightarrow \#w_i$ in c'txt w_j

Objective: $H(\theta;\mathbf{N}) = \sum_{n_{ij}>0}f(n_{ij})(\log n_{ij}-\log\exp[\langle\mathbf{x}_i,\mathbf{y}_j\rangle+b_i+d_j])^2$ with $f(n) = \min\{1,(\frac{n}{n_{max}})^\alpha\}$, $\alpha\in(0;1]$.

unnormalized distribution → two-sided loss function

SGD: 1. $\mathbf{x}_i^{new}\leftarrow\mathbf{x}_i+2\eta f(n_{ij})(\log n_{ij}-\langle\mathbf{x}_i,\mathbf{y}_j\rangle)\mathbf{y}_j$

2. $\mathbf{y}_j^{new}\leftarrow\mathbf{y}_j+2\eta f(n_{ij})(\log n_{ij}-\langle\mathbf{x}_i,\mathbf{y}_j\rangle)\mathbf{x}_i$

7 Non-Negative Matrix Factorization (NMF) / pLSA

Context Model: $p(w|d) = \sum_{z=1}^K p(w|z)p(z|d)$

Conditional independence assumption (*): $p(w|d) = \sum_z p(w,z|d) = \sum_z p(w|d,z)p(z|d) \overset{*}{=} \sum_z p(w|z)p(z|d)$

Symmetric parameterization: $p(w,d) = \sum_z p(z)p(w|z)p(d|z)$

7.1 EM for pLSA:

1. $\mathbf{X} = x_{i,j}$ = # occ. of w_j in doc. d_i
2. Log-Likelihood: $L(\mathbf{U},\mathbf{V}) = \sum_{i,j}x_{i,j}\log p(w_j|d_i) = \sum_{(i,j)\in X}\log\sum_{z=1}^K p(w_j|z)p(z|d_i) = \sum_{(i,j)\in X}\log\sum_{z=1}^K v_{zj}u_{zi}$

3. E-Step (optimal q): $q_{zij} = \frac{v_{zj}u_{zi}}{\sum_{k=1}^{K} v_{kj}u_{ki}}$

4. M-Steps: $p(z|d_i) = \frac{\sum_j x_{ij}q_{zij}}{\sum_j x_{ij}}$ & $p(w_j|z) = \frac{\sum_i x_{ij}q_{zij}}{\sum_{i,l} x_{il}q_{zil}}$

7.2 NMF Algorithm for quadratic cost function

• $\mathbf{X} \in \mathbb{Z}_{\geq 0}^{N\times M}$ • NMF: $\mathbf{X} \approx \mathbf{U}^\top \mathbf{V}, x_{ij} = \sum_z u_{zi}v_{zj} = \langle \mathbf{u}_i, \mathbf{v}_j \rangle$ $\min_{\mathbf{U},\mathbf{V}} J(\mathbf{U},\mathbf{V}) = \frac{1}{2}\|\mathbf{X} - \mathbf{U}^\top\mathbf{V}\|_F^2$ s.t. $\forall i,j,z\ u_{zi}, v_{zj} \geq 0$ 1. init: $\mathbf{U},\mathbf{V} = rand()$ 2. repeat for *maxIters*: 3. update $\mathbf{U}$: $(\mathbf{V}\mathbf{V}^\top)\mathbf{U} = \mathbf{V}\mathbf{X}^\top$ 4. project $u_{zi} = \max\{0, u_{zi}\}$ 5. update $\mathbf{V}$: $(\mathbf{U}\mathbf{U}^\top)\mathbf{V} = \mathbf{U}\mathbf{X}$ 6. project $v_{zj} = \max\{0, v_{zj}\}$

8 Convolutional Neural Networks

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 classification; $y_1 = \mathrm{P}[Y=1|\mathbf{x}] = \frac{1}{1+\exp[-\langle \mathbf{w}_1^L, \mathbf{x}^{L-1}\rangle]}$, multiclass; $y_k = \mathrm{P}[Y=k|\mathbf{x}] = \frac{\exp[\langle \mathbf{w}_k^L, \mathbf{x}^{L-1}\rangle]}{\sum_{m=1}^{K}\exp[\langle \mathbf{w}_m^L, \mathbf{x}^{L-1}\rangle]}$. **Loss function** $l(y,\hat{y})$: squared loss; $\frac{1}{2}(y-\hat{y})^2$, cross-entropy loss; $-y\log\hat{y} - (1-y)\log(1-\hat{y})$.

8.1 Neural Networks for Images

Translation invariance of images → neurons compute same fct, shift invariant filters; weights defined as filter masks, e.g. convolution: $F_{n,m}(\mathbf{x};\mathbf{w}) = \sigma(b + \sum_{k=-2}^{2}\sum_{l=-2}^{2} w_{k,l}x_{n+k,m+1})$. To reduce dimension of convolution, use {max, avg}-pooling

9 Optimization

9.1 Coordinate Descent (update the d-th coord. per step)

1. init: $\mathbf{x}^{(0)} \in \mathbb{R}^D$ 2. for $t=0$ to *maxIter*: 3. sample u.a.r. $d \sim \{1,\ldots,D\}$ 4. $u^\star = \arg\min_{u\in\mathbb{R}} f(x_1^{(t)},..,x_{d-1}^{(t)},u,x_{d+1}^{(t)},..,x_D^{(t)})$ 5. $\mathbf{x}_d^{(t+1)} = u^\star$ and $\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t)}$ for $i \neq d$

9.2 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*: $\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \gamma\nabla f(\mathbf{x}^{(t)})$, usually $\gamma \approx \frac{1}{t}$

9.3 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 u.a.r. $n \sim \{1,\ldots,N\}$ 4. $\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \gamma\nabla f_n(\mathbf{x}^{(t)})$, usually stepsize $\gamma \approx \frac{1}{t}$.

9.4 Projected Gradient Descent (Constrained Opt.)

minimize $f(x), x \in Q$ (constraint). **Project** x onto Q: $P_Q(\mathbf{x}) = \arg\min_{y\in Q}\|\mathbf{y}-\mathbf{x}\|$, **Projected Gradient Update**: $\mathbf{x}^{(t+1)} = P_Q[\mathbf{x}^{(t)} - \gamma\nabla f(\mathbf{x}^{(t)})]$, $\mathbf{x}^{(t+1)}$ is unique if Q convex.

9.5 Lagrangian Multipliers

Minimize $f(\mathbf{x})$ s.t. $g_i(\mathbf{x}) \leq 0,\ i=1,..,m$ (**inequality constr.**) and $h_i(\mathbf{x}) = \mathbf{a}_i^\top\mathbf{x} - b_i = 0,\ i=1,..,p$ (**equality constraint**)

Lagrangian: $L(\mathbf{x},\lambda,\nu) := f(\mathbf{x}) + \sum_{i=1}^{m}\lambda_i g_i(\mathbf{x}) + \sum_{i=1}^{p}\nu_i h_i(\mathbf{x})$ **Dual function:** $D(\lambda,\nu) := \inf_{\mathbf{x}} L(\mathbf{x},\lambda,\nu) \in \mathbb{R}$ **Dual Problem:** $\max_{\lambda,\nu} D(\lambda,\nu)$ s.t. $\lambda \geq \mathbf{0}$. Note: $\max_{\lambda,\nu} D(\lambda,\nu) \leq \min_{\mathbf{x}} f(\mathbf{x})$, equality if *dom f* and f convex

9.6 Convex Optimization

Q convex: $\forall \mathbf{x},\mathbf{y} \in Q : \forall 0 \leq \alpha \leq 1 : \alpha\mathbf{x} + (1-\alpha)\mathbf{y} \in Q$ $f : \mathbb{R}^D \to \mathbb{R}$ is convex, if *dom f* is a convex set, and if $\forall \mathbf{x},\mathbf{y} \in dom\ f : \forall 0 \leq \alpha \leq 1$: $f(\alpha\mathbf{x} + (1-\alpha)\mathbf{y}) \leq \alpha f(\mathbf{x}) + (1-\alpha)f(\mathbf{y})$. local=global min, **Convergence**: $f(\mathbf{x}^{(t)}) - f(\mathbf{x}^*) \leq \frac{c}{t}$. **Subgradient** $g \in \mathbb{R}^D$ of f at $\mathbf{x}$: $f(\mathbf{y}) \geq f(\mathbf{x}) + g^\top(\mathbf{y}-\mathbf{x})\ \forall\mathbf{y}$ Convergence: $f(\mathbf{x}^{(t)}) - f(\mathbf{x}^*) \leq \frac{c}{\sqrt{t}}$.

10 Sparse Coding

10.1 Orthogonal Basis

For $\mathbf{x}$ and orthogonal $\mathbf{U}$ compute $\mathbf{z} = \mathbf{U}^\top\mathbf{x}$. Approx $\hat{\mathbf{x}} = \mathbf{U}\hat{\mathbf{z}}$, $\hat{z}_i = z_i$ if $|z_i| > \varepsilon$ else 0. Energy preserving $\|Uz\| = \|z\|$. Reconstruction Error $\|\mathbf{x} - \hat{\mathbf{x}}\|^2 = \sum_{d\notin\sigma}\langle \mathbf{x}, \mathbf{u}_d\rangle^2$.

10.2 Overcomplete Basis

$\mathbf{U} \in \mathbb{R}^{D\times L}$ for dim(data) $= L > D =$ # atoms. 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)

10.3 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 2. Dictionary update step: $\mathbf{U}^{t+1} \in \arg\min_{\mathbf{U}}\|\mathbf{X} - \mathbf{U}\mathbf{Z}^{t+1}\|_F^2$, subject to $\forall l \in [L] : \|\mathbf{u}_l\|_2 = 1$

11 Robust PCA

• Idea: Approximate $\mathbf{X}$ with $\mathbf{L}+\mathbf{S}$, $\mathbf{L}$ is low-rank, $\mathbf{S}$ is sparse.
• $\min_{\mathbf{L},\mathbf{S}}\mathrm{rank}(\mathbf{L}) + \mu\|\mathbf{S}\|_0$, s. t. $\mathbf{L}+\mathbf{S} = \mathbf{X}$. As non-convex, change to $\min_{\mathbf{L},\mathbf{S}}\|\mathbf{L}\|_\star + \mu\|\mathbf{S}\|_1$ (*not* the same in general)
• Perfect recovery ($X = S_0 + L_0 = S^\star + L^\star$) is *not* possible if $\mathbf{S}$ is low-rank, $\mathbf{L}$ is sparse, or $\mathbf{X}$ is low-rank *and* sparse at the same time. Principal components of $\mathbf{L}$ must not be sparse.

11.1 Alternating Direction Method of Multipliers (ADMM)

$\min_{\mathbf{x}_1,\mathbf{x}_2} f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2)$ s. t. $\mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_2\mathbf{x}_2 = \mathbf{b}$, f_1, f_2 convex • Augmented Lagrangian: $L_\rho(\mathbf{x}_1,\mathbf{x}_2,\lambda) = f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2) + \lambda^\top(\mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_2\mathbf{x}_2 - \mathbf{b}) + \frac{\rho}{2}\|\mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_2\mathbf{x}_2 - \mathbf{b}\|_2^2$

• ADMM: $\mathbf{x}_1^{(t+1)} := \arg\min_{\mathbf{x}_1} L_p(\mathbf{x}_1,\mathbf{x}_2^{(t)},\lambda^{(t)})$, $\mathbf{x}_2^{(t+1)} := \arg\min_{\mathbf{x}_2} L_p(\mathbf{x}_1^{(t+1)},\mathbf{x}_2,\lambda^{(t)})$, $\lambda^{(t+1)} := \lambda^{(t)} + p(\mathbf{A}_1\mathbf{x}_1^{(t+1)} + \mathbf{A}_2\mathbf{x}_2^{(t+1)} - \mathbf{b})$ • ADDM for RPCA: $L_\rho(\mathbf{L},\mathbf{S},\lambda) = \|\mathbf{L}\|_\star + \mu\|\mathbf{S}\|_1 + \langle\lambda, vec(\mathbf{L}+\mathbf{S}-\mathbf{X})\rangle + \frac{\rho}{2}\|\mathbf{L}+\mathbf{S}-\mathbf{X}\|_F^2$
• $\mathbf{L}^{t+1} = argmin_{\mathbf{L}}(L_p(\mathbf{L},\mathbf{S}^t,\lambda^t))$ • $\mathbf{S}^{t+1} = argmin_{\mathbf{L}}(L_p(\mathbf{L}^{t+1},\mathbf{S},\lambda^t))$ • $\lambda^{t+1} = \lambda^t + \rho vec(\mathbf{L}^{t+1} + \mathbf{S}^{t+1} - \mathbf{X})$