CIL Cheat Sheet 2015

Jan Wilken Dörrie

Norms

Vector norms

A *norm* is a function $\|\cdot\| : V \to \mathbb{R}$ quantifying the size of a vector. It must satisfy

1) Positive scalability: $\|a \cdot \mathbf{x}\| = |a| \cdot \|\mathbf{x}\|$ for $a \in \mathbb{R}$
2) Triangle inequality: $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$, $\mathbf{x}, \mathbf{y} \in V$.
3) Seperability: $\|\mathbf{x}\| = 0$ implies $\mathbf{x} = 0$.

- Most common are *p-norms*: $\|\mathbf{x}\|_p := \left(\sum_{i=1}^n |x_i|^p\right)^{1/p}$
- Special case is *Euclidean norm*: $\|\mathbf{x}\|_2 := \sqrt{\sum_{i=1}^n x_i^2}$
- The "0-*norm*" is $\|\mathbf{x}\|_0 := |\{x_i \mid x_i \neq 0\}|$

Matrix norms

We can also define norms on matrices, satisfying the properties described above. $\mathbf{A} \in \mathbb{R}^{M \times N}$:

- *Frobenius*: $\|\mathbf{A}\|_F := \sqrt{\sum_{ij} a_{ij}^2} = \sqrt{\sum_{i=1}^{\min(M,N)} \sigma_i^2}$
- *p-norms for matrices*: $\|\mathbf{A}\|_p := \sup\{\|\mathbf{A}\mathbf{x}\|_p \,/\, \|\mathbf{x}\|_p\}$
- *Euclidean*: $\|\mathbf{A}\|_2 := \sup\{\|\mathbf{A}\mathbf{x}\|_2 \,/\, \|\mathbf{x}\|_2\} = \sigma_{\max}$
- *Nuclear norm*: $\|\mathbf{A}\|_* := \sum_{i=1}^{\min(M,N)} \sigma_i$

Statistics

Kullback-Leibler Divergence

- Divergence between discrete probability distributions P and Q: $D_{\mathrm{KL}}(P\|Q) = \sum_{\omega \in \Omega} P(\omega) \log\left(\frac{P(\omega)}{Q(\omega)}\right)$.
- Properties of the Kullback-Leibler Divergence
  - $D_{\mathrm{KL}}(P\|Q) \geq 0$.
  - $D_{\mathrm{KL}}(P\|Q) = 0$ if and only if P and Q are identical.
  - $D_{\mathrm{KL}}(P\|Q) \neq D_{\mathrm{KL}}(Q\|P)$!
  - caution: the KL-Divergence is not symmetric, therefore it is not a metric/distance!

Dimension Reduction

Principal Component Analysis (PCA)

Orthogonal linear projection of high dimensional data onto low dimensional subspace. Objectives:

1) Minimize error $\|\mathbf{x} - \tilde{\mathbf{x}}\|_2$ of point $\mathbf{x}$ and its approximation $\tilde{\mathbf{x}}$.
2) Preserve information: maximize variance.

Both objectives are shown to be formally equivalent.

Statistics of Projected Data:

- Mean of the data: sample mean $\bar{\mathbf{x}}$
- Covariance of the data:

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

Solution: Eigenvalue Decomposition: The eigenvalue decomposition of the covariance matrix $\mathbf{\Sigma} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ contains all relevant information.

- For $K \leq D$ dimensional projection space: Choose K eigenvectors $\{\mathbf{u}_1, \ldots, \mathbf{u}_K\}$ with largest associated eigenvalues $\{\lambda_1, \ldots, \lambda_K\}$.

Singular Value Decomposition

Theorem (Eckart-Young). *Let $\mathbf{A}$ be a matrix of rank R, if we wish to approximate $\mathbf{A}$ using a matrix of a lower rank K then, $\tilde{\mathbf{A}} = \sum_{k=1}^{K} d_k \mathbf{u}_k \mathbf{v}_k^\top$ is the closest matrix in the Frobenius norm. (Assumes ordering of singular values $d_k \geq d_{k+1}$)*

Clustering

K-Means

Motivation:

- Given: set of data points $\mathbf{x}_1, \ldots, \mathbf{x}_N \in \mathbb{R}^D$
- Goal: find *meaningful partition* of the data
  - i.e. a labeling of each data point with a unique label

    $$\pi : \{1, \ldots, N\} \to \{1, \ldots, K\} \text{ or } \pi : \mathbb{R}^D \to \{1, \ldots, K\}$$
  - note: numbering of clusters is arbitrary
  - k-th cluster recovered by $\pi^{-1}(k) \subseteq \{1, \ldots, N\}$ or $\subseteq \mathbb{R}^D$

Vector Quantization:

- Partition of the space $\mathbb{R}^D$
- Clusters represented by *centroids* $\mathbf{u}_k \in \mathbb{R}^D$
- Mapping induced via nearest centroid rule

$$\pi(\mathbf{x}) = \underset{k=1,\ldots,K}{\operatorname{argmin}} \|\mathbf{u}_k - \mathbf{x}\|_2$$

Objective Function for K-Means:

- Useful notation: represent π via indicator matrix $\mathbf{Z}$:

$$z_{kn} := [\pi(\mathbf{x}_n) = k]$$

- K-means Objective function

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

where $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N] \in \mathbb{R}^{D \times N}$ and $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_K] \in \mathbb{R}^{D \times K}$

K-means Algorithm: Optimal Assignment:

- Compute optimal assignment $\mathbf{Z}$, given centroids $\mathbf{U}$
  - minimize each column of $\mathbf{Z}$ separately

    $$\mathbf{z}_{\bullet n}^* = \underset{z_{1n}, \ldots, z_{Kn}}{\operatorname{argmin}} \sum_{k=1}^{K} z_{kn} \|\mathbf{x}_n - \mathbf{u}_k\|_2^2$$
  - optimum is attained by mapping to the closest centroid

    $$z_{kn}^*(\mathbf{U}) = \left[k = \underset{l}{\operatorname{argmin}} \|\mathbf{x} - \mathbf{u}_l\|_2\right]$$

K-means Algorithm: Optimal Assignment:

- Compute optimal choice of $\mathbf{U}$, given assignments $\mathbf{Z}$
  - continuous variables: compute gradient and set to zero (necessary optimality condition)
  - look at (partial) gradient for every centroid $\mathbf{u}_k$

    $$\nabla_{\mathbf{u}_k} J(\mathbf{U}, \mathbf{Z}) = \sum_{n=1}^{N} z_{kn} \nabla_{\mathbf{u}_k} \|\mathbf{x}_n - \mathbf{u}_k\|_2^2 = -2 \sum_{n=1}^{N} z_{kn}($$
  - setting gradient to zero

    $$\nabla_{\mathbf{U}} J(\mathbf{U}, \mathbf{Z}) \stackrel{!}{=} 0 \implies \mathbf{u}_k^*(\mathbf{Z}) = \frac{\sum_{n=1}^{N} z_{kn} \mathbf{x}_n}{\sum_{n=1}^{N} z_{kn}}$$

K-means Algorithm: Analysis:

- Computational cost of each iteration is $O(KND)$
- K-means convergence is guaranteed
- K-means optimizes a non-convex objective. Hence we are not guaranteed to find the global optimum.
- Finds a local optimum $(\mathbf{U}, \mathbf{Z})$ in the following sense
  - for each $\mathbf{Z}'$ with $\frac{1}{2}\left\|\mathbf{Z} - \mathbf{Z}'\right\|_0 = 1$ (differs in one assignment)
  - $J(\mathbf{U}^*(\mathbf{Z}'), \mathbf{Z}') \geq J(\mathbf{U}, \mathbf{Z})$
  - may gain by changing assignments of ≥ 2 points
- K-means algorithm can be used to compress data
  - with information loss, if $K < N$
  - store only the centroids and the assignments

Mixture Models

GMM: $p_\theta(\mathbf{x}) = \sum_k \pi_k \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, $\boldsymbol{\pi} \geq \mathbf{0}$, $\|\boldsymbol{\pi}\|_1 = 1$.

Complete Data Distribution:

- Explicitly introduce latent variables in the generative model
- Assignment variable (for a generic data point) $z_k \in \{0,1\}$, $\sum_{k=1}^K z_k = 1$
- We have that $\Pr(z_k = 1) = \pi_k$ or $p(\mathbf{z}) = \prod_{k=1}^K \pi_k^{z_k}$
- Joint distribution over $(\mathbf{x}, \mathbf{z})$ (*complete data* distribution) $p(\mathbf{x}, \mathbf{z}) = \prod_{k=1}^K \left[\pi_k \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right]^{z_k}$

Posterior Assignments: Posterior probabilities for assignments

$$\Pr(z_k = 1 \mid \mathbf{x}) = \frac{\pi_k \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{l=1}^K \pi_l \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}$$

Lower Bounding the Log-Likelihood:

- Expectation Maximization
  - maximize a lower bound on the log-likelihood
  - systematic way of deriving a family of bounds
  - based on complete data distribution
- Specifically:

$$\begin{aligned}
\ln p_\theta(\mathbf{x}) &= \ln \sum_{\mathbf{z}} p_\theta(\mathbf{x}, \mathbf{z}) = \ln \sum_{k=1}^K p(\mathbf{x}, \theta_k)\pi_k \\
&= \ln \sum_{k=1}^K q_k \frac{p(\mathbf{x}; \theta_k)\pi_k}{q_k} \\
&\geq \sum_{k=1}^K q_k \left[\ln p(\mathbf{x}, \theta_k) + \ln \pi_k - \ln q_k\right]
\end{aligned}$$

  - follows from Jensen's inequality (concavity of logarithm)
  - can be done for the contribution of each data point (additive)

Mixture Model: Expectation Step:

$$q_k = \frac{\pi_k\, p(\mathbf{x}; \theta_k)}{\sum_{l=1}^K \pi_l\, p(\mathbf{x}, \theta_l)} = \Pr(z_k = 1 \mid \mathbf{x})$$

Mixture Model: Maximization Step: $\pi_k^* = \frac{1}{N}\sum_{n=1}^N q_{kn}$, $\boldsymbol{\mu}_k^* = \frac{\sum_{n=1}^N q_{kn}\mathbf{x}_n}{\sum_{n=1}^N q_{kn}}$ and $\boldsymbol{\Sigma}_k^* = \frac{\sum_{n=1}^N q_{kn}(\mathbf{x}_n - \boldsymbol{\mu}_k^*)(\mathbf{x}_n - \boldsymbol{\mu}_k^*)^\top}{\sum_{n=1}^N q_{kn}}$

AIC and BIC:

- Trade-off: achieve balance between data fit — measured by likelihood $p(\mathbf{X} \mid \theta)$ — and complexity. Complexity can be measured by the number of free parameters $\kappa(\cdot)$.
- Different Heuristics for choosing K
  - *Akaike Information Criterion* (AIC)

$$\mathrm{AIC}(\theta \mid \mathbf{X}) = -\ln p_\theta(\mathbf{X}) + \kappa(\theta)$$

  - *Bayesian Information Criterion* (BIC)

$$\mathrm{BIC}(\theta \mid \mathbf{X}) = -\ln p_\theta(\mathbf{X}) + \frac{1}{2}\kappa(\theta)\ln N$$

- Generally speaking, the BIC criterion penalizes complexity more than the AIC criterion.

Non-Negative Matrix Factorization

Non-Negative Matrix Factorization:

- *Document-term matrix* $\mathbf{X} \in \mathbb{R}_{\geq 0}^{D \times N}$ storing the word counts for each document:

$$\mathbf{X} = \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$$

  x_{dn}: Frequency of the d-th word in the n-th document.
- *Non-negative matrix factorization* (NMF) of $\mathbf{X}$:

$$\mathbf{X} \approx \mathbf{UZ}$$

  - with $\mathbf{U} \in \mathbb{R}_{\geq 0}^{D \times K}$ and $\mathbf{Z} \in \mathbb{R}_{\geq 0}^{K \times N}$
    * N: number of documents
    * D: vocabulary size
    * K: number of dimensions (design choice)
    * data reduction: $(D + N)K \ll DN$

pLSI — Generative Model:

- For a given *document* sample $\mathrm{len}(\mathrm{document})$ words by a two-stage procedure:
  - sample a topic according to $P(\mathrm{topic} \mid \mathrm{document})$
  - sample a word according to $P(\mathrm{word} \mid \mathrm{topic})$
- Key assumption: *conditional independence* of word and document given topic
- Conditional distribution of a word, given a document:

$$P(\mathrm{word} \mid \mathrm{document}) = \sum_{k=1}^K P(\mathrm{word} \mid \mathrm{topic}_k) P(\mathrm{topic}_k \mid \mathrm{do}$$

- Side note: how to sample a "new" document? Can use fully generative model of LDA.

pLSI — Matrix Factorization View:

- *Normalize* the elements of $\mathbf{X}$ so that they correspond to relative frequencies:

$$T := \sum_{d=1}^D \sum_{n=1}^N x_{dn}, \qquad x_{dn} \leftarrow \frac{x_{dn}}{T}$$

- *Matrix Factorization*
  - pLSI can be understood as a matrix factorization of the form $\mathbf{X} \approx \mathbf{UZ}$, with $\mathbf{U} \in \mathbb{R}_{\geq 0}^{D \times K}$, and $\mathbf{Z} \in \mathbb{R}_{\geq 0}^{K \times N}$
  - where additionally we have the constraints:
    * $\sum_{d=1}^D u_{dk} = 1 (\forall k)$, identify $u_{dk} \equiv P(\mathrm{word}_d \mid \mathrm{topic}_k)$
    * $\sum_{k,n} z_{kn} = 1$, identify $z_{kn} \equiv P(\mathrm{topic}_k \mid \mathrm{document}_n) P(\mathrm{document}_n)$

pLSI — Parameter Estimation:

- Goal: maximize the likelihood of the data under the model
- Data: the relative frequencies $\mathbf{X}$
- Probabilistic model: $P(\text{word}_d, \text{document}_n) = \sum_{k=1}^{K} P(\text{word}_d \mid \text{topic}_k) P(\text{topic}_k \mid \text{document}_n) = (\mathbf{UZ})_{dn}$
- *Log likelihood*: $\log \mathcal{L}(\mathbf{U}, \mathbf{Z}; \mathbf{X}) = \log P(\mathbf{X}; \mathbf{U}, \mathbf{Z}) = \sum_{d=1}^{D} \sum_{n=1}^{N} x_{dn} \log \sum_{k=1}^{K} u_{dk} z_{kn}$

EM for pLSI — Variational Likelihood:

- Follow similar recipe as for Gaussian Mixture Model
- Reindex the observations in a per token manner with $t = 1, \ldots, T$
  - pairs of word/documents indexes (d_t, n_t)
  - note that $\sum_{t=1}^{T} f(d_t, n_t) = \sum_{d=1}^{D} \sum_{n=1}^{N} x_{dn} f(d, n)$ for arbitrary functions f
- *Variational Likelihood*

$$\begin{aligned}
&\log P(\mathbf{X}; \mathbf{U}, \mathbf{Z}) \\
&= \sum_{t=1}^{T} \log (\mathbf{UZ})_{d_t n_t} = \sum_{t=1}^{T} \log \left[\sum_{k=1}^{K} u_{d_t k} z_{k n_t} \right] \\
&\geq \sum_{t=1}^{T} \max_{q \in \mathcal{S}_K} \sum_{k=1}^{K} q_k [\log u_{d_t k} + \log z_{k n_t} - \log q_k]
\end{aligned}$$

  - $\mathcal{S}_K := \left\{ x \in \mathbb{R}^K \mid x \geq 0, \sum_{k=1}^{K} x_k = 1 \right\}$ (probability simplex)

EM for pLSI — Derivation of E-step:

- Compute the argmin in the variational bound

$$q_t^* = \operatorname*{argmax}_{q \in \mathcal{S}_K} \sum_{k=1}^{K} q_k [\log u_{d_t k} + \log z_{k n_t} + \log q_k]$$

- Form Lagrangian and differentiate

$$\frac{\partial}{\partial q_k} \{ q_k [\log u_{d_t k} + \log z_{k n_t} - \log q_k - \lambda_t^*] \} \stackrel{!}{=} 0$$

$$\Longrightarrow q_{tk}^* \propto u_{d_t k} z_{k n_t}, \text{ i.e. } q_{tk}^* = \frac{u_{d_t k} z_{k n_t}}{\sum_{l=1}^{K} u_{d_t l} z_{l n_t}}$$

- q_{tk}^* = posterior probability that t-th token (i.e. word with index d_t in document with index n_t) has been generated from topic k

EM for pLSI — Derivation of M-step:

- Differentiate lower bound with plugged in optimal choices for q_t^* $(t = 1, \ldots, T)$
- M-step solution for $\mathbf{U}$ and $\mathbf{Z}$

$$u_{dk}^* = \frac{\sum_{t: d_t = d} q_{tk}^*}{\sum_{t=1}^{T} q_{tk}^*} \qquad z_{kn}^* = \frac{\sum_{t: n_t = n} q_{tk}^*}{T}$$

SPARSE CODING

Optimization

Coordinate Descent: Idea: Update one coordinate at a time, while keeping others fixed.

- Algorithm:
  - initialize $\mathbf{x}^{(0)} \in \mathbb{R}^D$
  - for $t = 0, \ldots, \text{maxIter}$
    - $d \leftarrow \mathcal{U}\{1, D\}$
    - $u^* \leftarrow \operatorname{argmin}_{u \in \mathbb{R}} f\left(x_1^{(t)}, \ldots, x_{d-1}^{(t)}, u, x_{d+1}^{(t)}, \ldots, x_D^{(t)}\right)$
    - $x_d^{(t+1)} \leftarrow u^*, \quad x_{d'}^{(t+1)} \leftarrow x_{d'}^{(t)}$ for $d' \neq d$

Gradient Descent Method:

- Algorithm:
  - initialize $\mathbf{x}^{(0)} \in \mathbb{R}^D$
  - for $t = 0, \ldots, \text{maxIter}$
    - $\mathbf{x}^{(t+1)} \leftarrow \mathbf{x}^{(t)} - \gamma \nabla f\left(\mathbf{x}^{(t)}\right)$
- simple to implement
- good scalability and robustness
- *stepsize* γ usually decreasing with $\gamma \approx \frac{1}{t}$

Stochastic Gradient Descent:

- Optimization Problem Structure: minimize $f(\mathbf{x}) = \frac{1}{N} \sum_{n=1}^{N} f_n(\mathbf{x})$ with $\mathbf{x} \in \mathbb{R}^D$
- Algorithm:
  - initialize $\mathbf{x}^{(0)} \in \mathbb{R}^D$
  - for $t = 0, \ldots, \text{maxIter}$
    - $n \leftarrow \mathcal{U}\{1, N\}$
    - $\mathbf{x}^{(t+1)} \leftarrow \mathbf{x}^{(t)} - \gamma \nabla f_n\left(\mathbf{x}^{(t)}\right)$

Duality for Constrained Optimization:

- Constrained Problem Formulation (Standard Form): minimize $f(\mathbf{x})$ subject to $g_i(\mathbf{x}) \leq 0$, $i = 1, \ldots, m$, $h_i(\mathbf{x}) = 0$, $i = 1, \ldots, p$
- Unconstrained Problem: minimze $f(\mathbf{x}) + \sum_{i=1}^{m} I_-(g_i(\mathbf{x})) + \sum_{i=1}^{p} I_0(h_i(\mathbf{x}))$. I_- and I_0 are "brickwall" indicator functions.

Dual Problem:

- Lagrangian: $L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) := f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}) + \sum_{i=1}^{p} \nu_i h_i(\mathbf{x})$
- Lagrange dual function: $d(\boldsymbol{\lambda}, \boldsymbol{\nu}) := \inf_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu})$
- Lagrange *dual problem*: maximize $d(\boldsymbol{\lambda}, \boldsymbol{\nu})$ subject to $\boldsymbol{\lambda} \geq \mathbf{0}$.
  It is always a lower bound on the primal value $f(\mathbf{x})$ of any feasible $\mathbf{x}$ and thus a lower bound on the unknown solution value $f(\mathbf{x}^*)$ of the primal problem.
  Strong Duality: If the primal optimization problem is convex and under some additional conditions, the solution value of the dual problem is *equal* to the solution value $f(\mathbf{x}^*)$ of the primal problem.

Convexity:

- Convex Set: A set $\mathcal{Q}$ is convex if for any $\mathbf{x}, \mathbf{y} \in \mathcal{Q}$ and any $\theta \in [0, 1]$, we have $\theta \mathbf{x} + (1 - \theta) \mathbf{y} \in \mathcal{Q}$.
- Convex Function: A function $f\colon \mathbb{R}^D \to \mathbb{R}$ is convex if $\operatorname{dom} f$ is a convex set and if for all $\mathbf{x}, \mathbf{y} \in \operatorname{dom} f$, and $\theta \in [0, 1]$ we have $f(\theta \mathbf{x} + (1 - \theta) \mathbf{y}) \leq \theta f(\mathbf{x}) + (1 - \theta) f(\mathbf{y})$.
- Convex Optimization: Convex Optimization Problems are of the form $\min f(\mathbf{x})$ s.t. $\mathbf{x} \in \mathcal{Q}$ where both f is a convex function and $\mathcal{Q}$ is a convex set (note: $\mathbb{R}^D$ is convex). In Convex Optimization Problems every local minimum is a *global minimum*.

Sparse Coding

Properties of Haar Wavelets:

- Mother wavelet: $\psi(t) = \left[t \in [0, \tfrac{1}{2})\right] - \left[t \in [\tfrac{1}{2}, 1)\right]$
- Haar function: $\psi_{n,k}(t) = 2^{n/2} \psi(2^n t - k)$, $n, k \in \mathbb{Z}$
- $\psi_{n,k}(t)$ non-zero on $I_{n,k} = [k 2^{-n}, (k+1) 2^{-n})$
- Integral 0: $\int_{\mathbb{R}} \psi_{n,k}(t) \, \mathrm{d}t = 0$
- Norm 1: $\|\psi_{n,k}\|_{L^2(\mathbb{R})}^2 = \int_{\mathbb{R}} \psi_{n,k}(t)^2 \, \mathrm{d}t = 1$

- Orthogonal: $\int_{\mathbb{R}} \psi_{n_1,k_1}(t)\psi_{n_2,k_2}(t)\,\mathrm{d}t = \delta_{n_1,n_2}\delta_{k_1,k_2}$
- $\implies$ Haar system is orthonormal basis in $L^2(\mathbb{R})$

Any continuous real function on $[0, 1]$ can be approximated uniformly on $[0, 1]$ by linear combinations of the constant function $\mathbf{1}, \psi(t), \psi(2t), \psi(4t), \ldots, \psi(2^n t), \ldots$ and their shifted functions.

Discrete cosine transform (DCT):

- 1D DCT: $z_k = \sum_{n=0}^{N-1} x_n \cos[\frac{\pi}{N}(n + \frac{1}{2})k]$
- 2D DCT: $z_{k_1,k_2} = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x_{n_1,n_2} \cos[\frac{\pi}{N_1}(n_1+\frac{1}{2})k_1] \cos[\frac{\pi}{N_2}(n_2+\frac{1}{2})k_2]$

Compressive Sensing: Main idea: acquire the set $\mathbf{y}$ of M linear combinations of the initial signal instead of the signal itself and then reconstruct the initial signal from these measurements. $\mathbf{y} = \mathbf{W}\mathbf{x} = \mathbf{W}\mathbf{U}\mathbf{z} =: \mathbf{\Theta}\mathbf{z}$, with $\mathbf{\Theta} = \mathbf{W}\mathbf{U} \in \mathbb{R}^{M\times D}$. Surprisingly given any orthonormal basis $\mathbf{U}$ we can obtain a stable reconstruction for any K-sparse, compressible signal. Two conditions: $w_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \frac{1}{D})$ and $M \geq cK\log(\frac{D}{K})$, $c \in \mathbb{R}$. For $M \ll D$ ill-posed, hence $\mathbf{z}^* \in \operatorname{argmin}_{\mathbf{z}} \|\mathbf{z}\|_0$, s.t. $\mathbf{y} = \mathbf{\Theta}\mathbf{z}$. NP-hard, approximate with *Matching Pursuit* or do convex relaxation with $\|\mathbf{z}\|_1$.

Coding via orthogonal transforms: Given orig. signal $\mathbf{x}$ and orthogonal matrix $\mathbf{U}$ compute change of basis $\mathbf{z} = \mathbf{U}^\top\mathbf{x}$. Truncate "small" values, giving $\hat{\mathbf{z}}$. Compute inverse transform $\hat{\mathbf{x}} = \mathbf{U}\hat{\mathbf{z}}$.

Measure performance: error $\|\mathbf{x} - \hat{\mathbf{x}}\|$ and sparsity $\|\mathbf{z}\|_0$.

Dictionary choice: Fourier Dictionary is good for "sine like" signals, Wavelet Dictionary is good for localized signals.

Overcomplete Dictionaries

In contract to othogonal bases there are more atoms than dimensions ($L > D$). Coding algorithm chooses best representation (subset of atoms), but this is mathematically involved due to non-orthogonality (no closed form reconstruction).

Coherence: Increasing the Overcompleteness factor $\frac{L}{D}$ potentially increases the sparsity of the coding, but also increases the linear dependency between atoms. Coherence is a measurement for this: $m(\mathbf{U}) = \max_{i,j:i\neq j} |\mathbf{u}_i^\top \mathbf{u}_j|$. $m(\mathbf{U}) = 0$ for an orthogonal basis $\mathbf{B}$, $m([\mathbf{B}\mathbf{u}]) \geq \frac{1}{\sqrt{D}}$ if atom $\mathbf{u}$ is added to orthogonal $\mathbf{B}$.

Signal Coding: $\mathbf{U} \in \mathbb{R}^{D\times L}$ is overcomplete, so finding $\mathbf{z}$ such that $\mathbf{x} = \mathbf{U}\mathbf{z}$ is ill-posed, more unknowns than equations. Need to add sparsity constraint: $\mathbf{z}^* \in \operatorname{argmin}_{\mathbf{z}} \|\mathbf{z}\|_0$ s.t. $\mathbf{x} = \mathbf{U}\mathbf{z}$. Problem is NP-hard, can be brute-forced for small instances, needs Matching Pursuit else.

Noisy Observations: Signal might be corrupted, $\mathbf{x} = \mathbf{U}\mathbf{z} + \mathbf{n}$ with $n_d \sim \mathcal{N}(0, \sigma^2)$. Solve either $\mathbf{z}^* \in \operatorname{argmin}_{\mathbf{z}} \|\mathbf{z}\|_0$ s.t. $\|\mathbf{x} - \mathbf{U}\mathbf{z}\|_2^2 < D\sigma^2$ or $\mathbf{z}^* \in \operatorname{argmin}_{\mathbf{z}} \|\mathbf{x} - \mathbf{U}\mathbf{z}\|_2$ s.t. $\|\mathbf{z}\|_0 \leq K$.

Matching Pursuit (MP) Algorithm: Greedy algorithm that starts with zero vector $\mathbf{z} = \mathbf{0}$ and residual $\mathbf{r}^0 = \mathbf{x}$. At each iteration t selects atom with maximal absolute correlation to residual $d^* \leftarrow \operatorname{argmax}_d |\mathbf{u}_d^\top \mathbf{r}^{(t)}|$ and updates vectors $z_{d^*} \leftarrow z_{d^*} + \mathbf{u}_{d^*}^\top \mathbf{r}^{(t)}$, $\mathbf{r}^{(t+1)} \leftarrow \mathbf{r}^{(t)} - (\mathbf{u}_{d^*}^\top \mathbf{r}^{(t)})\mathbf{u}_{d^*}$. Stops when $\|\mathbf{z}\|_0 = K$. MP is an approximation, but recovers exact coding when $K < \frac{1}{2}(1 + \frac{1}{m(\mathbf{U})})$.

Sparse Coding for Inpainting: Define diagonal masking matrix $\mathbf{M}$, $m_{d,d} = [\text{pixel } d \text{ is known}]$, sparse coding of known parts in overcomplete dictionary $\mathbf{U}$: $\mathbf{z}^* \in \operatorname{argmin}_{\mathbf{z}} \|\mathbf{z}\|_0$ s.t. $\|\mathbf{M}(\mathbf{x} - \mathbf{U}\mathbf{z})\|_2 < \sigma$. Image reconstruction using mask: $\hat{\mathbf{x}} = \mathbf{M}\mathbf{x} + (\mathbf{I} - \mathbf{M})\mathbf{U}\mathbf{z}^*$.

Dictionary Learning

When learning the dictionary we adapt a dictionary to signal characteristics in the data, for which we have to solve a matrix factorization problem $\mathbf{X} = \mathbf{U}\mathbf{Z}$ with sparsity constraint on $\mathbf{Z}$ and atom norm constraint on $\mathbf{U}$. $(\mathbf{U}^*, \mathbf{Z}^*) \in \operatorname{argmin}_{\mathbf{U},\mathbf{Z}} \|\mathbf{X} - \mathbf{U}\mathbf{Z}\|_F^2$, objective not jointly convex over $\mathbf{U}$ and $\mathbf{Z}$ but convex in either of them when the other one is fixed.

Iterative greedy minimization:

1) Coding step: $\mathbf{Z}^{(t+1)} \in \operatorname{argmin}_{\mathbf{Z}} \left\|\mathbf{X} - \mathbf{U}^{(t)}\mathbf{Z}\right\|_F^2$, subject to $\mathbf{Z}$ being sparse and $\mathbf{U}$ being fixed.
2) Dict. update: $\mathbf{U}^{(t+1)} \in \operatorname{argmin}_{\mathbf{U}} \left\|\mathbf{X} - \mathbf{U}\mathbf{Z}^{(t+1)}\right\|_F^2$, subject to $\|\mathbf{u}_l\|_2 = 1$ for all l and $\mathbf{Z}$ being fixed.

Coding step can be done column-wise via *Matching Pursuit* and dictionary update via *K-SVD* algorithm involving a power iteration to approximate SVD solution. Dictionary learning can also be used for Speech Enhancement by learning the Speech and Noise dictionaries and then setting the noise coefficients to zero.

ROBUST PCA

Goal: Find a low rank representation of a matrix $\mathbf{X}$, which is corrupted by a sparse perturbation or sparse structured noise. Additive decomposition problem: $\min_{\mathbf{L},\mathbf{S}} \operatorname{rank}(\mathbf{L}) + \lambda\|\mathbf{S}\|_0$ s.t. $\mathbf{L} + \mathbf{S} = \mathbf{X}$. This problem is non-convex and thus hard to solve, convex relaxation: $\min_{\mathbf{L},\mathbf{S}} \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1$ s.t. $\mathbf{L} + \mathbf{S} = \mathbf{X}$. This is *not* the same problem, but achieves the same solution under broad conditions.

Alternating Direction Method of Multiplies (ADMM): $\min_{\mathbf{x}_1,\mathbf{x}_2} f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2)$ s.t. $\mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_2\mathbf{x}_2 = \mathbf{b}$ with f_1, f_2 convex. *Augmented Lagrangian*: $L_\rho(\mathbf{x}_1, \mathbf{x}_2, \boldsymbol{\nu}) = f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2) + \boldsymbol{\nu}^\top(\mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_2\mathbf{x}_2 - \mathbf{b}) + \frac{\rho}{2}\|\mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_2\mathbf{x}_2 - \mathbf{b}\|_2^2$, punishes violations of the constraints even more. Update steps: $\mathbf{x}_1^{(t+1)} := \operatorname{argmin}_{\mathbf{x}_1} L_\rho(\mathbf{x}_1, \mathbf{x}_2^{(t)}, \boldsymbol{\nu}^{(t)})$, $\mathbf{x}_2^{(t+1)} := \operatorname{argmin}_{\mathbf{x}_2} L_\rho(\mathbf{x}_1^{(t+1)}, \mathbf{x}_2, \boldsymbol{\nu}^{(t)})$, $\boldsymbol{\nu}^{(t+1)} := \boldsymbol{\nu}^{(t)} + \rho(\mathbf{A}_1\mathbf{x}_1^{(t+1)} + \mathbf{A}_2\mathbf{x}_2^{(t+1)} - \mathbf{b})$.

ADMM for RPCA: Here $f_1(\mathbf{x}_1) = \|\mathbf{L}\|_*$ and $f_2(\mathbf{x}_2) = \lambda\|\mathbf{S}\|_1$, hence $L_\rho(\mathbf{L}, \mathbf{S}, \mathbf{N}) = \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1 + \langle\mathbf{N}, (\mathbf{L} + \mathbf{S} - \mathbf{X})\rangle + \frac{\rho}{2}\|\mathbf{L} + \mathbf{S} - \mathbf{X}\|_F^2$. Update sequence: $\mathbf{L}^{(t+1)} := \operatorname{argmin}_{\mathbf{L}} L_\rho(\mathbf{L}, \mathbf{S}^{(t)}, \mathbf{N}^{(t)})$, $\mathbf{S}^{(t+1)} := \operatorname{argmin}_{\mathbf{S}} L_\rho(\mathbf{L}^{(t+1)}, \mathbf{S}, \mathbf{N}^{(t)})$, $\mathbf{N}^{(t+1)} := \rho(\mathbf{L}^{(t+1)} + \mathbf{S}^{(t+1)} - \mathbf{X})$. Solving explicitly: $\operatorname{argmin}_{\mathbf{L}} L_\rho(\mathbf{L}, \mathbf{S}, \mathbf{N}) = \mathcal{D}_{\rho^{-1}}(\mathbf{X} - \mathbf{S} - \rho^{-1}\mathbf{N})$, $\operatorname{argmin}_{\mathbf{S}} L_\rho(\mathbf{L}, \mathbf{S}, \mathbf{N}) = \mathcal{S}_{\rho^{-1}}(\mathbf{X} - \mathbf{L} - \rho^{-1}\mathbf{N})$, where $\mathcal{S}_\tau(x) = \operatorname{sgn}(x)\max(|x| - \tau, 0)$, $\mathcal{S}_\tau(\mathbf{X})$ applies $\mathcal{S}_\tau$ to all x_{ij}, $\mathcal{D}_\tau(\mathbf{X}) = \mathbf{U}\mathcal{S}_\tau(\mathbf{\Sigma})\mathbf{V}^\top$, where $\operatorname{SVD}(\mathbf{X}) = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$.

Identifiability: The solutions to the convex relaxation solve the original problem when the Coherence condition is fulfilled (principal components must not be sparse (spiky)) and both the rank of $\mathbf{L}_0$ and the number of non-zero entries of $\mathbf{Z}_0$ is not too large.

RPCA can be used for collaborative filtering when relaxing the constraints to only consider known observations.