# CIL Cheat Sheet 2015

Jan Wilken Dörrie

## I. Linear Algebra Primer

### A. Equivalent Conditions

For $\mathbf{A} \in \mathbb{R}^{M\times M}$ the following conditions are equivalent:

- $\mathbf{A}$ has an inverse $\mathbf{A}^{-1}$,
- $\operatorname{rank}(\mathbf{A}) = M$,
- $\operatorname{range}(\mathbf{A}) = \mathbb{R}^M$,
- $\operatorname{null}(\mathbf{A}) = \{\mathbf{0}\}$,
- $0$ is not an eigenvalue of $\mathbf{A}$,
- $0$ is not a singular value of $\mathbf{A}$

## II. Norms

### A. Vector norms

A *norm* is a function $\|\bullet\| : V \to \mathbb{R}$ 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: $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$ for $\mathbf{x}, \mathbf{y} \in V$.
3) Seperability: $\|\mathbf{x}\| = 0$ implies $\mathbf{x} = 0$.

- The most commonly used norms are the *p-norms*:

$$\|\mathbf{x}\|_p := \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}$$

- A special case is the *Euclidean norm*

$$\|\mathbf{x}\|_2 := \sqrt{\sum_{i=1}^{n} x_i^2}$$

- The "0-*norm*" (not really a norm) is $\|\mathbf{x}\|_0 := |\{x_i \mid x_i \neq 0\}|$

### B. Matrix norms

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

- *Frobenius norm*:

$$\|\mathbf{A}\|_F := \sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N} a_{ij}^2} = \sqrt{\sum_{i=1}^{K} \sigma_i^2}, \quad K = \min(M, N)$$

  Only depends on singular values of $\mathbf{A}$
- *p-norms for matrices*: $\|\mathbf{A}\|_p := \sup\{\|\mathbf{A}\mathbf{x}\|_p : \|\mathbf{x}\|_p = 1\}$
- *Euclidean or spectral norm*:

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

  the largest singular value.

## III. Dimension Reduction

### A. 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.

*1) Statistics of Projected Data:*

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

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

*2) 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 $\{\lambda_1, \ldots, \lambda_K\}$.

### B. 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 $d_k \geq d_{k+1}$)*

## IV. Clustering

### A. K-Means

*1) 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 $\subseteq \mathbb{R}^D$

*2) 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

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

*3) Objective Function for $K$-Means:*

- Useful notation: represent $\pi$ 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 $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_K] \in \mathbb{R}^{D \times K}$

*4) $K$-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

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

*5) $K$-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 $\mathbf{u}_k$

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

  - setting gradient to zero

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

*6) $K$-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 $\frac{1}{2} \|\mathbf{Z} - \mathbf{Z}'\|_0 = 1$ (differs in one assignment)
  - $J(\mathbf{U}^*(\mathbf{Z}'), \mathbf{Z}') \geq J(\mathbf{U}, \mathbf{Z})$
  - may gain by changing assignments of $\geq 2$ points
- $K$-means algorithm can be used to compress data
  - with information loss, if $K < N$
  - store only the centroids and the assignments

## B. Mixture Models

*1) Gaussian Mixture Model (GMM):*

$$p_\theta(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k),$$

where $\pi_k \geq 0$, $\sum_{k=1}^{K} \pi_k = 1$.

*2) 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) $p(\mathbf{x}, \mathbf{z}) = \prod_{k=1}^{K} [\pi_k \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]^{z_k}$

*3) Posterior Assignments: Posterior probabilities* for assignments

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

*4) 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)

*5) Mixture Model: Expectation Step:*

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

*6) Mixture Model: Maximization Step:*

$$\pi_k^* = \frac{1}{N} \sum_{n=1}^{N} q_{kn}$$

$$\boldsymbol{\mu}_k^* = \frac{\sum_{n=1}^{N} q_{kn} \mathbf{x}_n}{\sum_{n=1}^{N} q_{kn}}$$

$$\boldsymbol{\Sigma}_k^* = \frac{\sum_{n=1}^{N} q_{kn} \mathbf{x}_n \mathbf{x}_n^\top}{\sum_{n=1}^{N} q_{kn}}$$

*7) 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.

## V. Sparse Coding

## VI. Robust PCA