Various

Filtering (Stop word removal): Get grid of common terms
+ reduce vocabulary size (space), - language specific

Stemming: Stripping off word endings to reduce a word to its stem/core (e.g. PorterStemmer).
+ reduce vocabulary size (space)
+ unifies words with same meaning, but slight variation (foxes > fox)
− language specific (for each language different rules)
− 1 extra step (quite expensive)
− can result in non-dictionary words
− words with different meaning can be mapped to same word (automatic / automate > autom)

Lemmatization: Mapping words to its root form.
E.g. (walk, walked, walks, walking) > walk or better > good
+ get true dictionary form of a word, - hard to achieve in practice

Term normalization (general): Allows matching more terms
+ identify small variations of same term
− can lead to loss in precision

3 - Evaluation of Relevance

Precision (P): Fraction of retrieved documents that are relevant.

$$\mathrm{P} = \frac{\#\text{ relevant items retrieved}}{\#\text{ items retrieved}} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$$

Recall (R): Fraction of relevant documents that are retrieved.

$$\mathrm{R} = \frac{\#\text{ relevant items retrieved}}{\#\text{ relevant items in collection}} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$$

High Precision vs High Recall: It's a tradeoff!

- By returning more documents ⇒ recall increases monotonically
  E.g. return all document ⇒ recall of 1
  Comparison shopping ⇒ wants high recall (user wants all offers)
- By returning fewer documents ⇒ often precision increases
  E.g. return 1 document ⇒ precision of 1 if document is relevant
  Web search ⇒ wants high precision (user just looks at few result)

F-Measure (F): Something in between precision and recall

$$F_\beta = \frac{(\beta^2+1)\mathrm{PR}}{\beta^2\mathrm{P}+\mathrm{R}} \quad \text{where } \beta^2 = \frac{1-\alpha}{\alpha}$$

$$F_1 = \frac{2\,\mathrm{PR}}{\mathrm{P}+\mathrm{R}}$$

A/B tests: Can also run A/B tests with 2 systems

1. Sample queries to evaluate on 2 systems
2. For each query show both results to raters
3. Raters judge which system is better
4. Compute overall statistics

4 - Scoring: TF-IDF

Scoring by matching terms: We have the general expectation:

- If query term doesn't occur in document ⇒ score contribution should be 0!
- The more frequent the query term in the document ⇒ the higher the score contribution!
- The more informative the query term ⇒ the higher the score contribution! E.g. bomb
- Given same term frequency ⇒ shorter document should be preferred

Term Frequency: Absolute frequency of a word in a document

$$\text{tf(w; d)} = \#\text{ word w in document d}$$

$$\text{log-tf(w; d)} = \log_2\left(1+\frac{\text{tf(w; d)}}{\text{document\_length}}\right)$$

$$\text{atf(w; d)} = \frac{1}{2}+\frac{1}{2}\frac{\text{tf(w; d)}}{\max\{\text{w}' : \text{tf(w}'\text{; d)}\}}$$

$$\text{score(query; d)} = \sum_{\text{w}\in\text{query}} \text{log-tf(w; d)}$$

Note:

- Augmented-tf is very sensible to maximum (stop word pruning)
- Using raw term frequencies (tf) is discouraged in practice

Document Frequency: Quantifies importance of a query term

$$\text{df(w)} = \#\{\text{d} : \text{tf(w; d)} > 0\}$$
$$= \#\text{ documents in collection that contain w}$$

Note:

- low df ⇒ more informative/topical (e.g. bomb)
- high df ⇒ less informative/topical (e.g. the)

Inverse Document Frequency: Translates df into term weights

$$\text{idf(w)} = \log\left(\frac{n}{\text{df(w)}}\right) = \log(n) - \log(\text{df(w)}) \qquad n = \text{num documents}$$

Note:

- low idf ⇒ less informative/topical (e.g. the)
- high idf ⇒ more informative/topical (e.g. bomb)

Collection Frequency: Like term freq. but in whole collection

$$\text{cf(w)} = \sum_d \text{tf(w; d)}$$
$$= \#\text{word occurrences in whole collection}$$

$$\text{rcf(w)} = \frac{\text{cf(w)}}{\sum_{\text{w}'}\text{cf(w}')}$$

TF-IDF: Combine both tf and idf in one term weight

$$\text{tf-idf(w; d)} = \text{log-tf(w; d)}\cdot\text{idf(w)}$$
$$= \log\left(1+\frac{\text{tf(w; d)}}{\text{document\_length}}\right)\cdot\log\left(\frac{n}{\text{df(w)}}\right)$$

$$\text{score(query; d)} = \sum_{\text{w}\in\text{query}} \text{tf-idf(w; d)}$$

Note: We want that both tf(w) and idf(w) are large!

Vector Space Model: Represent both documents and queries in vector space (BoW) and rank documents according to their proximity to the query (e.g. use cosine distance).

Gaussian Mixture Model

K mixture components: $p(\mathbf{x}) = \sum_{k=1}^{K}\pi_k p(\mathbf{x}|\theta_k) = \sum_{k=1}^{K}\pi_k\mathcal{N}(\mathbf{x}|\mu_k,\mathbf{\Sigma}_k)$

Data-Likelihood: $p(\mathbf{X}|\pi,\mu,\mathbf{\Sigma}) \overset{iid}{=} \prod_{n=1}^{N} p(\mathbf{x}_n) = \prod_{n=1}^{N}\sum_{k=1}^{K}\pi_k\mathcal{N}(\mathbf{x}_n|\mu_k,\mathbf{\Sigma}_k)$

EM-Algorithm

Maximize log-likelihood:

$$(\hat{\pi},\hat{\mu},\hat{\mathbf{\Sigma}}) \in \underset{\pi,\mu,\mathbf{\Sigma}}{\arg\max}\sum_{n=1}^{N}\log\sum_{k=1}^{K}\pi_k\mathcal{N}(\mathbf{x}_n|\mu_k,\mathbf{\Sigma}_k)$$

1. *Introduce latent variable:* $z_k \in \{0,1\}$, $\sum_k z_k = 1$, $p(z_k=1)=\pi_k$ and initialize μ_k and π_k. Set $\mathbf{\Sigma}_k$ to the given covariances.
2. *E-step:* Compute expectation (responsibilities)
   $\gamma(z_{k,n}) = p(z_{k,n}=1|\mathbf{x}_n) = \frac{p(z_k=1)p(\mathbf{x}_n|z_k=1)}{\sum_{j=1}^{K}p(z_j=1)p(\mathbf{x}_n|z_j=1)} = \frac{\pi_k\mathcal{N}(\mathbf{x}_n|\mu_k,\mathbf{\Sigma}_k)}{\sum_{j=1}^{K}\pi_j\mathcal{N}(\mathbf{x}_n|\mu_j,\Sigma_j)}$
3. *M-step:* Re-estimate model parameters
   $\mu_k^{\text{new}} = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{k,n})\mathbf{x}_n$ and $\pi_k^{\text{new}} = \frac{N_k}{N}$ where $N_k = \sum_{n=1}^{N}\gamma(z_{k,n})$

Estimating K: $\kappa(\cdot)$ = dof. E.g in GMM: $\kappa(\mathbf{U},\mathbf{Z}) = KD + (K-1)$

- *AIC:* $-\ln p(\mathbf{X}|\theta) + \kappa(\mathbf{U},\mathbf{Z})$
- *BIC:* $-\ln p(\mathbf{X}|\theta) + \frac{1}{2}\kappa(\mathbf{U},\mathbf{Z})\ln N$

RBAC

Given user-permission matrix $\mathbf{X}\in\mathbb{R}^{D\times N}$, find roles
$\mathbf{U} = [\mathbf{u}_1,\ldots,\mathbf{u}_K]\in\mathbb{B}^{D\times K}$ and assignment
$\mathbf{Z} = [\mathbf{z}_1,\ldots,\mathbf{z}_N]\in\mathbb{B}^{K\times N}$ s.t:
$\mathbf{X} = \mathbf{U}\otimes\mathbf{Z} \iff x_{dn} = \bigvee_k[u_{dk}\wedge z_{kn}]$

Model with $\beta = (\beta_{dk})^{D\times K}$: probability that role k has **no** permission d

SAC: $p(\mathbf{X}|\beta,\mathbf{Z}) = \prod_{n,d} p(x_{dn}=1|\beta_{d\cdot},\mathbf{z}_{\cdot n})^{x_{dn}}\cdot p(x_{dn}=0|\beta_{d\cdot},\mathbf{z}_{\cdot n})^{1-x_{dn}}$
$= \prod_{n,d}(1-\beta_{d,k_n})^{x_{dn}}\,(\beta_{d,k_n})^{1-x_{dn}}$

MAC: $p(\mathbf{X}|\beta,\mathbf{Z}) = \prod_{n,d}(1-\prod_k\beta_{dk}^{z_{kn}})^{x_{dn}}\,(\prod_k\beta_{dk}^{z_{kn}})^{1-x_{dn}}$
$= \prod_{n,d}(1-\beta_{d,\mathcal{L}_n})^{x_{dn}}\,(\beta_{d,\mathcal{L}_n})^{1-x_{dn}}$

Final Model: using noise model: $x_{dn} = (1-\xi_{dn})(\mathbf{U}\otimes\mathbf{Z})_{dn} + \xi_{dn}\nu_{dn})$

$p(\mathbf{X}|\mathbf{Z},\beta,\varepsilon,r)$: $\prod_{n,d}(\varepsilon r+(1-\varepsilon)(1-\beta_{d,\mathcal{L}_n}))^{x_{dn}}\cdot(\varepsilon(1-r)+(1-\varepsilon)\beta_{d,\mathcal{L}_n})^{1-x_{dn}}$

ε: Noise probability, r probability of noisy bits to be 1

Evaluating a Matrix Reconstruction

Deviation: $\frac{1}{N \cdot D}||X-\hat{\mathbf{U}}\otimes\hat{\mathbf{Z}}||_1$ **Deviating 1:** $\frac{|\{(i,j)|\hat{x}_{i,j}=1,x_{i,j}=0\}|}{|\{(i,j)|x_{i,j}=1|}$

Coverage: $\frac{|\{(i,j)|\hat{x}_{i,j}=x_{i,j}=1\}|}{|\{(i,j)|x_{i,j}=1|}$ **Deviating 0:** $\frac{|\{(i,j)|\hat{x}_{i,j}=0,x_{i,j}=1\}|}{|\{(i,j)|x_{i,j}=0|}$

Non-Negative MF

Given Document-term matrix $\mathbf{X}\in\mathbb{R}_+^{D\times N}$. We want a NMF for which holds:

$$\mathbf{X}\approx\mathbf{UZ} \text{ with } \mathbf{U}\in\mathbb{R}_+^{D\times K} \text{ and } \mathbf{Z}\in\mathbb{R}_+^{K\times N}$$

Probabilistic LSI

Generate tuple $(document, word)$**:**

1. Sample document according to $P(document)$
2. Sample word according to $P(word|document)$

Assume a factorization: $P(word|doc) = \sum_{\text{topic}} P(word|topic)P(topic|doc)$

Therefore: $P(word, doc) = \sum_{\text{topic}} P(word|topic)P(topic, doc)$

Rewrite: $P(d\text{th } word,\ n\text{th } document) = x_{dn} = (\mathbf{UZ})_{dn}$

Quadratic NMF

Consider non-negative $\mathbf{X}$ and quadratic cost fnc. (like in K-Means):

$$\min_{\mathbf{U},\ \mathbf{Z}} J(\mathbf{U},\mathbf{Z})=\frac{1}{2}||\mathbf{X}-\mathbf{UZ}||_F^2 \quad \text{s.t.} \quad u_{dk},\ z_{kn}\in\mathbb{R}_0^+$$

Algorithm:

1. Init $\mathbf{U},\ \mathbf{Z}$ with pos. random values
2. loop:

   Update factors $\mathbf{U}$: $u_{dk}=u_{dk}\frac{(\mathbf{XZ}^\top)_{dk}}{(\mathbf{UZZ}^\top)_{dk}}$

   Update coefficients $\mathbf{Z}$: $z_{kn}=z_{kn}\frac{(\mathbf{U}^\top\mathbf{X})_{kn}}{(\mathbf{U}^\top\mathbf{UZ})_{kn}}$

This leads to $\mathbf{X}\approx\mathbf{UZ}$ when $K<N$

Derivation:

Lagrangian: $L(\mathbf{U},\mathbf{Z},\boldsymbol{\alpha},\boldsymbol{\beta})=J(\mathbf{U},\mathbf{Z})-tr(\boldsymbol{\alpha}\mathbf{U}^\top)-tr(\boldsymbol{\beta}\mathbf{Z}^\top)$

$J(\mathbf{U},\mathbf{Z})=\frac{1}{2}||\mathbf{X}-\mathbf{UZ}||_F^2=\frac{1}{2}tr\left((X-UZ)(X-UZ)^T\right)$

$=\frac{1}{2}(\mathrm{tr}(\mathbf{XX}^\top)-2\mathrm{tr}(\mathbf{XU}^\top\mathbf{Z}^\top)+\mathrm{tr}(\mathbf{UZZ}^\top\mathbf{U}^\top))$

Taking derivatives and setting to 0 leads to above update rules:

$\frac{\partial J}{\partial\mathbf{U}}=\mathbf{UZZ}^\top-\mathbf{XZ}^\top\overset{!}{=}0$ and $\frac{\partial J}{\partial\mathbf{Z}}=\mathbf{U}^\top\mathbf{UZ}-\mathbf{U}^\top\mathbf{X}\overset{!}{=}0$

Sparse Coding

Given signal $f=\mathbf{x}$ and orthonormal basis $\mathbf{U}=[\mathbf{u}_1,\ldots,\mathbf{u}_K]$:

Full reconstruction: $f=\sum_{k=1}^{K}\langle f,u_k\rangle u_k=\sum_{k=1}^{K}z_k u_k$

Approx. (compression): $\hat{f}=\sum_{k\in\sigma}z_k u_k$ where σ is a subset of size $\tilde{K}$

Reconstruction error: $||f-\hat{f}||_2^2=\langle f-\hat{f},f-\hat{f}\rangle=\ldots=\sum_{k\notin\sigma}z_k^2$

Fourier basis: Global support: + for sine-like sig, - for localized sig.

Wavelet basis: Local support: + for localizd sig, - for nonvanishing sig

Compressive Sensing

Assume $\mathbf{x}\in\mathbb{R}^{D\times 1}$ is sparse in some orthonormal basis $\mathbf{U}\in\mathbb{R}^{D\times D}$ with K large coefficients in $\mathbf{z}\in\mathbb{R}^{D\times 1}$: $\mathbf{x}=\mathbf{Uz}$

Idea: Instead of saving $\mathbf{x}$ we save $\mathbf{y}$ with dimension $M<<D$

Store $\mathbf{y}$*:* $\mathbf{y}=\mathbf{Wx}=\mathbf{WUz}=\boldsymbol{\Theta}\mathbf{z}$ with $\boldsymbol{\Theta}=\mathbf{WU}\in\mathbb{R}^{M\times D}$

Restore $\mathbf{x}$*:* $\mathbf{z}^\star=\arg\min_{\mathbf{z}}||\mathbf{z}||_0$ s.t. $\boldsymbol{\Theta}\mathbf{z}=\mathbf{y}$ (use MP)

$\mathbf{x}=\mathbf{Uz}^\star$

Overcomplete dictionaries

Assume $\mathbf{U}\in\mathbb{R}^{D\times L}$ is overcomplete

Objective: $\mathbf{z}^*\in\arg\min_z||\mathbf{z}||_0$ s.t. $\mathbf{x}=\mathbf{Uz}$ (NP hard problem)

Coherence

Increasing the overcompleteness factor $\frac{L}{D}$: Increases the sparsity of the coding, but also increases the linear dependency between atoms.

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

- $m(\mathbf{B})=0$ for an orthogonal basis $\mathbf{B}$
- $m([\mathbf{Bu}])\geq\frac{1}{\sqrt{D}}$ if atom $\mathbf{u}$ added to $\mathbf{B}$

Matching Pursuit (MP)

Greedy algo to approximate NP hard problem iteratively.

Objective: $\mathbf{z}^\star\in\arg\min_{\mathbf{z}}||\mathbf{x}-\mathbf{Uz}||_2$ s.t. $||z||_0\leq K$

Algo: At each iter., take a step in direction of the atom $\mathbf{u}_{d^\star}$ that minimizes at most the residual $||\mathbf{x}\text{-}\mathbf{Uz}||_2$ where $d^\star\in\arg\max_d|\langle\mathbf{r},\mathbf{u}_d\rangle|$

Note: minimizing $||\mathbf{r}||_2$ is equiv. as maxim. abs. correlation $|\langle\mathbf{r},\mathbf{u}_d\rangle|$

1. Start with zero vector $\mathbf{z}=\mathbf{0}$ and residual $\mathbf{r}=\mathbf{x}$
2. While $||\mathbf{z}||_0<K$:

   Criteria: $d^\star=\arg\max_d|\mathbf{u}_d^\top\mathbf{r}|$

   Update: $z_{d^\star}=z_{d^\star}+\mathbf{u}_{d^\star}^\top\mathbf{r}$

   $\mathbf{r}=\mathbf{r}-(\mathbf{u}_{d^\star}^\top\mathbf{r})\,\mathbf{u}_{d^\star}$

Exact recovery when $K<\frac{1}{2}\left(1+\frac{1}{m(\mathbf{U})}\right)$ (K: # non-zero elements)

Dictionary Learning

Factorize training set $\mathbf{X} \in \mathbb{R}^{D\times N}$ into a dictionary $\mathbf{U} \in \mathbb{R}^{D\times L}$ and sparse matrix $\mathbf{Z} \in \mathbb{R}^{L\times N}$ such that: $(\mathbf{U}^*, \mathbf{Z}^*) \in \arg\min_{\mathbf{U},\mathbf{Z}} ||\mathbf{X} - \mathbf{U}\mathbf{Z}||_F^2$

Algorithm: Iterative greedy minimization between 2 steps

1. **Coding step**: Fix $\mathbf{U}$ and find sparsest possible $\mathbf{Z}$
   Objective: $\mathbf{Z}^{t+1} \in \arg\min_{\mathbf{Z}} ||\mathbf{X} - \mathbf{U}^t\mathbf{Z}||_F^2$, subject to $\mathbf{Z}$ being sparse
   $$\Rightarrow \mathbf{z}_n^{t+1} \in \arg\min_{\mathbf{z}} ||\mathbf{z}||_0 \quad \text{s.t.} \quad ||\mathbf{x}_n - \mathbf{U}^t\mathbf{z}||_2 \le \sigma ||\mathbf{x}_n||_2$$
2. **Dictionary update step**: Fix $\mathbf{Z}$ and find best $\mathbf{U}$
   Objective: $\mathbf{U}^{t+1} \in \arg\min_{\mathbf{U}} ||\mathbf{X} - \mathbf{U}\mathbf{Z}^{t+1}||_F^2$, subject to $||\mathbf{u}_l||_2 = 1\ \forall l$
   Approximation: update one atom $\mathbf{u}_l$ at a time for all $l = 1, .., L$:
   (a) Set $\tilde{\mathbf{U}} = [\mathbf{u}_1^t \dots \mathbf{u}_l \dots \mathbf{u}_L^t]$ (fix all atoms except $\mathbf{u}_l$).
   (b) Isolate $\mathbf{R}_l^t$, the residual that is due to atom $\mathbf{u}_l$:
   $||\mathbf{X} - \tilde{\mathbf{U}} \cdot \mathbf{Z}^{t+1}||_F^2 = ||\mathbf{R}_l^t - \mathbf{u}_l(\mathbf{z}_l^{t+1})^\top||_F^2$
   where $\mathbf{R}_l^t = X - \sum_{i\ne l} \mathbf{u}_i(\mathbf{z}_i^{t+1})^\top$
   Note: sum represents $\mathbf{UZ}$ ohne $\mathbf{u}_l$
   (c) Find $\mathbf{u}_l^\star$ that minimizes $\mathbf{R}_l^t$, subject to $||\mathbf{u}_l^\star||_2 = 1$ (use SVD):
   $\mathbf{R}_l^t = \mathbf{U}\mathbf{D}\mathbf{V}^\top = \sum_i d_i \mathbf{u}_i \mathbf{v}_i^\top \quad \Rightarrow \mathbf{u}_l^\star =$ first column of U

Convex Optimization

Primal problem: $\min_{\mathbf{x}} f(\mathbf{x})$ subject to $g_i(\mathbf{x}) \le 0, i = 1, \dots, m$
$h_i(\mathbf{x}) = 0, i = 1, \dots, p$

Lagrangian: $L(\mathbf{x}, \lambda, \nu) = f(\mathbf{x}) + \sum_{i=1}^m \lambda_i g_i(\mathbf{x}) + \sum_{i=1}^p \nu_i h_i(\mathbf{x})$ where $\lambda \ge 0$

Dual function: $d(\lambda, \nu) = \inf_{\mathbf{x}} L(\mathbf{x}, \lambda, \nu)$

Dual problem: $\max_{\lambda,\nu} d(\lambda, \nu)$ subject to $\lambda \ge 0$

Recover optimal x: $\mathbf{x}^\star = \arg\min_{\mathbf{x}} L(\mathbf{x}, \lambda^\star, \nu^\star)$

Note: Dual function is a lower bound on optimal value $p^\star$ of primal!
Proof: $d(\lambda, \nu) = \inf_{\mathbf{x}} L(\mathbf{x}, \lambda, \nu) \le \inf_{\tilde{\mathbf{x}}} L(\tilde{\mathbf{x}}, \lambda, \nu) \le \min_{\tilde{\mathbf{x}}} f(\tilde{\mathbf{x}}) = p^\star$

Convex optimization with equality constraints

Primal problem: $\min_x f(x)$ subject to $Ax = b$
Lagrangian: $L(\mathbf{x}, \nu) = f(x) + \nu^\top(Ax - b)$
Dual function: $d(\nu) = \inf_x L(x, \nu)$
Dual problem: $\max_{\nu} d(\nu)$

Gradient Method for dual:
$x^{k+1} = \arg\min_x L(x, \nu^k)$
$\nu^{k+1} = \nu^k + \alpha^k \nabla d(\nu^k) = \nu^k + \alpha^k \frac{\partial}{\partial \nu} L(x^{k+1}, \nu^k) = \nu^k + \alpha^k(Ax^{k+1} - b)$

Dual decomposition: If $f(x)$ with $x \in \mathbb{R}^n$ is separable than $L(x, \nu)$ is separable and we can split the x-min step:
$f(x) = f_1(x_1) + ... + f_n(x_n) \Rightarrow L(x, \nu) = L_1(x_1, \nu) + ... + L_n(x_n, \nu) - \nu^\top b$
$x_i^{k+1} = \arg\min_{x_i} L_i(x_i, \nu^k) = \arg\min_{x_i} f_i(x_i) + \nu^\top A_i x_i \quad i = 1..n$
$\nu^{k+1} = \nu^k + \alpha^k \nabla d(\nu^k) = \nu^k + \alpha^k(\sum_{i=1}^n A_i x_i^{k+1} - b)$

Method of Multipliers: Augment Lagrangian to L_ρ with $\frac{\rho}{2}||\cdot||_2^2$
$L_\rho(x, \nu) = f(x) + \nu^T(Ax - b) + \frac{\rho}{2}||Ax - b||_2^2$
$x^{k+1} = \arg\min_x L_\rho(x, \nu^k)$
$\nu^{k+1} = \nu^k + \rho \nabla d(\nu^k) = \nu^k + \rho(Ax^{k+1} - b)$
Choose ρ as step size, since x^{k+1} minimizes $L_\rho(x, \nu^k)$:
$$0 = \nabla_x L\rho(x^{k+1}, \nu^k) = \nabla_x f(x^{k+1}) + \underbrace{A^T(\nu^k + \rho(Ax^{k+1} - b))}_{A^T\nu^{k+1}}$$

Alternating Direction Method of Multipliers:
Since aug. Lag. L_ρ not separable anymore, can't parallelize x-min!
Primal: $\min_{x,z} f(x) + p(z)$ subject to $Ax + Bz = c \quad f, p$ convex
$L(x, z, \nu) = f(x) + p(z) + \nu^T(Ax + Bz - c)$
$L_\rho(x, z, \nu) = f(x) + p(z) + \nu^T(Ax + Bz - c) + \frac{\rho}{2}||Ax + Bz - c||_2^2$
$x^{k+1} = \arg\min_x L_\rho(x, z^k, \nu^k)$
$z^{k+1} = \arg\min_z L_\rho(x^{k+1}, z, \nu^k)$
$\nu^{k+1} = \nu^k + \rho\nabla d(\nu^k) = \nu^k + \rho(Ax^{k+1} + Bz^{k+1} - c)$
Primal Feasibility condition: $Ax^\star + Bz^\star - c = 0$
Dual Feasibiliy conditions: $\nabla f(x^\star) + A^\top \nu^\star = 0$ and $\nabla p(z^\star) + B^\top \nu^\star = 0$

Robust PCA

Original: $\min_{L,S} \text{rank}(\mathbf{L}) + \lambda \text{card}(\mathbf{S})$ subject to $\mathbf{L} + \mathbf{S} = \mathbf{X}$
Convex relaxation: $\min_{L,S} ||\mathbf{L}||_* + \lambda||\mathbf{S}||_1$ subject to $\mathbf{L} + \mathbf{S} = \mathbf{X}$
Exact($\mathrm{L}^\star = \mathrm{L}_0$, $\mathrm{S}^\star = \mathrm{S}_0$) with prob. $1 - \mathcal{O}(n^{-10})$,PCP with $\lambda = \frac{1}{\sqrt{n}}$ for:
$\mathbf{L}_0$: $n \times n$, of $rank(\mathbf{L}_0) \le \rho_r n \mu^{-1}(\log n)^{-2}$
$\mathbf{S}_0$: $n \times n$, random sparsity pattern of cardinality $m \le \rho_s n^2$
RPCA for CF: $\min_{L,S} ||\mathbf{L}||_* + \lambda||\mathbf{S}||_1$ s.t. $\mathbf{L}_{ij} + \mathbf{S}_{ij} = \mathbf{X}_{ij}$