1 Varia

Generalisation/Prediction Error: Expected #mistakes on unkown data

$GeneralisationError = Bias^2 + Variance + Noise$

2 Probability

Independence $E[XY] = E[X]E[Y]$

Covariance: $Cov(X_1, X_2) = E[X_1X_2] - E[X_1]E[X_2] = E[(X_1 - E[X_1])(X_2 - E[X_2])]$

Variance: $Var[X] = E[X^2] - E[X]^2$

Chain rule: $P(X,Y) = \frac{P(X,Y)P(Y)}{P(Y)} = P(X|Y)P(Y)$

Gaussian Dist. $p_{1D}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$

$p_{dD}(x) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$

Multiple of Gaussians are gaussian: $X \sim \mathcal{N}(\mu, \Sigma)$ $, Y = MX \Rightarrow Y \sim \mathcal{N}(M\mu, M\Sigma M^T)$

Sums of Gaussians are gaussian: $X_1 \sim \mathcal{N}(\mu_1, \Sigma_1)$, $X_2 \sim \mathcal{N}(\mu_2, \Sigma_2)$, $Y = X_1 + X_2 \Rightarrow Y \sim \mathcal{N}(\mu_1 + \mu_2; \Sigma_1 + \Sigma_2)$

3 Ridge Regression

Problem: $w^* = arg \min_w \sum_{i=1}^n \left(y_i - w^T x_i\right)^2 + \lambda||w||_2^2$

Closed Form: $w^* = \left(X^TX + \lambda I\right)^{-1} X^T y$

4 Sparse Regression: Lasso

Problem: $w^* = arg \min_w \sum_{i=1}^n \left(y_i - w^T x_i\right)^2 + \lambda||w||_1$

5 Kernelized Linear Regression

$$\min_{\alpha_1,\dots,\alpha_n} \sum_{i=1}^n \left(\sum_{j=1}^n \alpha_j k(x_i, x_j) - y_i\right)^2 + \lambda\alpha^T K\alpha$$

6 Linear Classification

Problem: $w^* = arg \min_w \sum_{i=1}^n l\left(w; x_i, y_i\right)$

6.1 Some Loss-Functions

Square-Loss $l_2 = (y_i - w^T x_i)^2$

0/1 Loss: $l_{0/1} = y_i \neq sign(w^T x_i)$

Perceptron Loss: $l_p\left(w; y_i, x_i\right) = max(0, -y_i w^T x_i)$

Hinge Loss (SVM): $l_h = max(0, 1 - y_i w^T x_i)$

6.1.1 Stochastic gradient Descent

pick random x' and y' if $l(w_t; x', y') \neq 0$

$w_{t+1} = w_t - \eta\nabla l(w_t; x', y')$ learning rate η

7 SVMs

Hard margin SVM problem: $\min_w w^T w, s.t.\ y_i W^T x_i >= 1$

Confidence $\eta = yw^T x$

Margin to w-plane $\gamma = \min_{x' \in L} ||x - x'||_2$

We are looking for a plane such that every sample has minimum distance 1(not actually 1).

7.1 Soft margin SVM

Unconstrained: $\min_w w^T w + C\sum_i^n max\left(0, 1 - y_i w^T x_i\right)$

Constrained: $\min_{w,\xi\geq 0} \frac{1}{2}||w||_2^2 + C\sum_{i=1}^n \xi_i$ s.t. $y_i\left(\langle w, x_i\rangle + w_0\right) \geq 1 - \xi_i\ \forall i = 1, ..., n$

Dual Form: $\max_\alpha \sum_{i=1}^m \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i\alpha_j y_i y_j x_i^T x_j$ s.t. $\sum_i \alpha_i y_i = 0$ and $0 \leq \alpha_i \leq C$

Optimal solution is a linear combination of the data $w^* = \sum_{i=1}^n \left(\alpha_i y_i\right) x_i$

Classify: $y_{new} = sign(\sum_{i=1}^n \alpha_i y_i k(x_i, x_{new})) = sign(w^T x)$

Multiclass: $\min_{w_1,\dots,w_c,\xi\geq 0} \sum_{y=1}^c w_y^T w_y + C\sum_i \xi_i$ s.t. $w_{y_i}^T x_i \geq w_y^T x_i + 1 - \xi_i \forall i \in \{1, ..., n\}, y \in \{1, ..., c\}$

8 Kernels $k(x, y)$

$k: \mathcal{X} \times \mathcal{X} \to \mathbb{R} \Leftrightarrow \forall x, y \in \mathcal{X}: k\left(x, y\right) = k\left(y, x\right)$

$\forall x, y$ Gram Matrix K: $(K)_{i,j} = k(i, j)$ is p.s.d

$k_1(x, y) = \langle\phi_1(x), \phi_1(y)\rangle$

8.1 Representer Theorem

Problem: $\min_{f\in\mathcal{H}} \sum_{i=1}^n l(f(x_i); x_i, y_i) + \lambda||f||_{\mathcal{H}}^2$

sol: $\min_{\alpha_1,\dots,\alpha_n} \sum_{i=1}^n l\left(\sum_{j=1}^n \alpha_j k(x_i, x_j); x_i, y_i\right) + \lambda\alpha^T K\alpha$

Best $f(x) = \sum_{j=1}^n \alpha_j k(x_j, x)$ is a sum of weighted kernel evaluations.

1. Gaussian Kernel: $exp\left(-\frac{||x-y||^2}{2\sigma^2}\right)$
2. Sigmoid Kernel: $tanh(\kappa x^T y + b)$
3. Polynomial Kernel: $\left(x^T y + c\right)^d, c \geq 0$

Closure Properties

1. $k(x, y) = ak_1(x, y) + bk_2(x, y)$, $a, b \geq 0$
2. $k(x, y) = k_1(x, y)k_2(x, y)$, k_1, k_2 are kernels
3. $k(x, y) = k_3(\phi(x), \phi(y))$, k_3 is a kernel
4. $k(x, y) = f(x)f(y)$
5. $k(x, y) = exp(k_1(x, y))$

9 Max. a posteriori estimation MAP

Pick most probable model w. Maximize likelihood of model parameter $w^* = arg \max_w P(w|x_1, ...x_n, y_1, ..., y_n)$

map estimate $P(w|x_1, .., x_n, y_1, ..., y_n) = \frac{P(w)P(y_1,\dots,y_n|x_1,\dots x_n,w)}{P(y_1,\dots,y_n|x_1,\dots,x_n)}$

10 Bayesian Learning

Key Idea: find $P(y|x, \theta)$

10.1 Prior Assumption

Laplace Prior $p(x; \mu, b) = \frac{1}{2b} exp\left(-\frac{|x-\mu|}{b}\right)$ corresponds to L1-Regularizer

Gauss Prior $p(x; \mu, \sigma)$ corresponds to L2-Regularizer

10.2 Logistic Regression

Classification method which replaces assumption about gaussian noise by iid bernoulli noise.

$P(y|x, w) = Ber(y; \sigma(w^T x))$

Link Func: $P(Y = +1|x, w) = \sigma(w^T x) = \frac{1}{1+exp(-w^T x)}$

Learn $w^* = arg \min_w \sum_{i=1}^n log\left(1 + exp(-y_i w^T x_i)\right)$

Classifiy $P(y|x_{new}, w^*) = \frac{1}{1+exp(-yw^{*T} x)}$

10.3 Bayesian Decision Theory

Best action a^* is determined: $a^* = arg\min_{a\in\mathcal{A}} E_y[C(y,a)|x] = arg\min_{a\in\mathcal{A}} \int_y C(y,a)p(y|x)dy$ where $C(y,a)$ is a cost function.

10.4 Bayesian model Averaging BMA

$P(y_{new}, x_{new}, D) = \int P(y_{new}|x_{new}, w)P(w|D)dw$

with $P(w|D) = \frac{P(w)P(D|w)}{P(D)}$ where $P(w)$ is the prior.

10.5 Neural Networks

learn $z = \sigma(x) = [\sigma_1(x), ..., \sigma_m(x)]$ and mapping $y = f(z)$

11 Gaussian Processes

$GP(f;\mu,k)$ with $\mu(x)$ as mean function and $k(x,x')$ as cov. function. If $y_i \sim \mathcal{N}(f(x_i),\sigma^2)$ with $A = \{x_1,...,x_n\}$ then Posterior is a GP $P(f|x_1,...,x_n1,y_1,...,y_n) = GP(f;\mu',k')$ with $\mu'(x) = \mu(x)+K_{x,A}(K_{AA}+\sigma^2 I)^{-1}(y_A-\mu_A)$ and $k'(x,x') = k(x,x') - K_{x,A}(K_{AA}+\sigma^2 I)^{-1}K_{A,x'}$

11.1 Bayesian linear regression

Prior $w \sim \mathcal{N}(0,\beta^2 I)$, prior distribution $y \sim \mathcal{N}(0,\beta^2 x^T x + \sigma^2)$ where error $e \sim \mathcal{N}(0,\sigma^2)$

Predictive dist $P(y|x,y_A) = \mathcal{N}(y; Y\mu_{y|A}, \sigma^2_{y|A})$ where $\mu_{y|A} = \Sigma_{x,A}\Sigma_{AA}^{-1}y_A$ and $\sigma^2_{y|A} = \Sigma_{xx} - \Sigma_{x,A}\Sigma_{AA}^{-1}\Sigma_{A,x}$

12 Ensemble Methods

Stumps $h(x) = sign(ax_i - t),\ a \in \{-1,+1\}$

Decision Trees are hierarchical ordered stumps.

Random Forest bagging with random ensemble of decision trees.

12.1 Bagging

Train each weak learner on a random subset of the data points. Classify new points by majority vote. Each iterations learns on a 'new' subset.

12.2 Boosting

AdaBoost, with weight w_i per datapoint greedily optimizes for exp loss.

- for $i = 1:m$
  - $h_i \leftarrow arg\min_h \sum_{j=1}^n w_j^{(i)}[h(x_j) \neq y_j]$
  - $err_i = \frac{\sum_{j=1}^n w_j^{(i)}[y_j \neq h_i(x_j)]}{\sum_{j=1}^n w_j^{(i)}}$, $\beta_i = log\frac{1-err_i}{err_i}$
  - $w_j^{(i+1)} = w_j^{(i)} exp(\beta_i[h_i(x_j) \neq y_j])$
- output: $f(x) = \sum_{i=1}^n \beta_i h_i(x)$

13 Generative Models

Aim to estimate joint dist. $P(y,x)$ instead of $P(y|x)$.

- Estimate prior on labels $P(y)$
- Estimate cond. dist. $P(x|y_i)\ \ \forall i \in I$
- Predict $P(y|x) = \frac{1}{Z}P(Y)P(x|y)$
  where $Z = \sum_{y'} P(x|y')$

13.1 Conjugate priors

Pair of Prior assumption $P(y)$ about data and likelihood function is called conjucate if posterior distribution remains in the same family as the prior.

13.2 Gaussian Naive Bayes Classifier

- MLE for class prior $P(Y=y) = \frac{Count(Y=y)}{n}$
- MLE feature dist. $P(x_i|y) = \mathcal{N}(x_i;\mu_{y,i},\sigma^2_{y,i})$ with $\mu_{y,i} = mean$ and $\sigma^2 = \frac{1}{Count(Y=y)}\sum(x-\mu)^2$

13.3 Fisher's LDA

2 gaussian dist. with fixed $p = 0.5$ and equal covariance Σ. Predict $y = sign(f(x)) = sign(w^T x + w_0)$ where $w = \Sigma^{-1}(\mu_+ - \mu_-)$ and $w_0 = \frac{1}{2}(\mu_-^T\Sigma^{-1}\mu_- - \mu_+^T\Sigma^{-1}\mu_+)$.

Where discriminant function $f(x) = log\frac{P(Y=1|x)}{P(Y=-1|x)}$

14 K-Means (Lloyd's Algo)

$z_i \leftarrow arg\min_{j\in\{1,...,k\}} ||x_i - \mu_j^{(t-1)}||_2^2$

$\mu_j^{(t)} \leftarrow \frac{1}{n_j}\sum_{i:z_i=j} x_i$

15 Gaussian Mixture Model (GMM)

$P(x|\theta) = P(x|\mu,\Sigma,w) = \sum_i w_i\mathcal{N}(x;\mu_i,\Sigma_i),\quad \sum_i w_i = 1$

Latent Variable: $\gamma_j(x_i) = P(z_i = j|x,\Sigma,\mu) = \frac{P(x|z_i=j)P(z_i=j)}{P(x)} = \frac{P(x|z_i=j)P(z_i=j)}{\sum_{q=1}^k P(x|z_i=q)P(z_i=q)} = \frac{w_j P(x|\Sigma_j,\mu_j)}{\sum_j w_j P(x|\Sigma_l,\mu_l)}$

$\mu_j^* = \frac{\sum_{i=1}^n \gamma_j(x_i)x_i}{\sum_{i=1}^n \gamma_j(x_i)}$

$\Sigma_j^* = \frac{\sum_{i=1}^n \gamma_j(x_i)(x_i-\mu_j^*)(x_i-\mu_j^*)^T}{\sum_{i=1}^n \gamma_j(x_i)} + \nu^2 I$

$w_j^* = \frac{1}{n}\sum_{i=1}^n \gamma_j(x_i)$

16 LinAlg

Positiv semi-definit $K \in \mathbb{R}$ is psd

$\iff x^T K x \geq 0\ \forall x \in \mathbb{R}$

$\iff$ all eigenvalues of K are ≥ 0

Norms: $||x||_p = (\sum_{i=1}^n |x_i|^p)^{\frac{1}{p}}$, $||x||_1 = \sum_{i=1}^n |x_i|$, $||x||_0 =$ nr of nonzero entries

17 Differentials

$f(g(x))\frac{d}{dx} = f'(g(x))\cdot g'(x)$, $\frac{d}{dx}log(x) = \frac{1}{x}$

17.1 Vector/Matrix differentiation

$\frac{d}{dx}f(x) = \left[\frac{\partial f}{\partial x_1},...,\frac{\partial f}{\partial x_n}\right]$, $\frac{d}{dx}(b^T x) = \frac{d}{dx}(x^T b) = b, \frac{d}{dx}(x^T x) = \frac{d}{dx}(x^T x) = 2x, \frac{d}{dx}(x^T A x) = (A^T + A)x$

18 Convex optimisation

minimize $f(x)$ subject to

$g_i(x) \leq 0, i = 1,...,m$ inequality constr.

$h_i(x) = 0, i = 1,...,p$ equality constr.

Create the Lagrangian

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

Lagrange dual function: $d(\lambda,\nu) = \inf_x L(x,\lambda,\nu)$

Lagrange dual problem: max. $d(\lambda,\nu)$ subj. to $\lambda \geq 0$

19 PCA - Principal Component Analysis

$Z = U_k^T \cdot X$ where Z is dim reduced. Project x to $\tilde{x}$ and minimize error $||x_n - \ x_n||_2$, variance of projected data is maximized.

- Covariance $\Sigma = \frac{1}{N}\cdot(X-M)(X-M)^T$
  - $Cov(X_i,X_i) = Var(X_i)$
  - Symmetric: $Cov(X_i,X_j) = Cov(X_j,X_i)$
- $Eig(\Sigma) = U\cdot\Lambda\cdot U^T$

Deduction Var. of proj. data Z is maximal if cov. $\Sigma_Z = A^T\Sigma_X A = \frac{1}{n}(A^T X - \bar{X})(A^T X - \bar{X})^T$.

By choosing $A = U$ where $\Sigma_X = U\Lambda U^T$ the covariance Σ_Z becomes diagonal.

ML - Summary 2013

January 31, 2014

20 PCA - Principal Component Analysis

High dimensional data is projected onto a low dimensional subspace while maximizing variance.

- project x to $\tilde{x}$ and minimize error $||x_n - \ x_n||_2$
- variance of projected data is maximized

20.0.1 Algorithm

- Covariance $\Sigma = \frac{1}{N} \cdot (X - M)(X - M)^T$
  - $Cov(X_i, X_i) = Var(X_i)$
  - Symmetric: $Cov(X_i, X_j) = Cov(X_j, X_i)$
- $Eig(\Sigma) = U \cdot \Lambda \cdot U^T$
- $Z = U_k^T \cdot X$ where Z is dim reduced.

20.0.2 Deduction

Var. of proj. data Z is maximal if cov.
$\Sigma_Z = A^T \Sigma_X A = \frac{1}{n}(A^T X - \bar{X})(A^T X - \bar{X})^T$.
By choosing $A = U$ where $\Sigma_X = U \Lambda U^T$ the covariance Σ_Z becomes diagonal.

21 SVD $M = UDV^T$

- Rank of M: Number of singular values
- Null space: right columns of V where σ_i are 0
- Range of M: left columns of U where σ_i are $\neq 0$
- Pseudo-Inverse: $M^+ = UD^+V$, where $D^+ = D$ with inverted singular values

21.0.3 SVD as a sum

$M_k = \sum_{i=1}^{k} U_i \cdot \Sigma_i \cdot V_i^T$
Minimize L2 Norm: SVD solves $||M - B||_2 = ||M - M_k||_2$ for euclidean matrix norms

21.1 Important

Eigenvectors of MM^T and $M^T M$:
$MM^T = UDV^T VDU^T = UD(V^T V)DU^T = UD^2U^T$
$M^T M = ... = VD^2V^T$
If $M = M^T$ (symmetric and real) then $S = U \cdot D \cdot U^T$
Where U has columns of Eigenvectors

22 Linear Algebra

22.1 Vector Norms

are positive scalable, full-fill the triangular inequality, norm of 0 is 0

22.1.1 p-Norm

$||x||_p = (\sum_{i=1}^{n} |x_i|^p)^{\frac{1}{p}}$

22.1.2 Euclidean Norm

p-Norm where $p = 2$

22.1.3 1-Norm

Manhattan-Norm $||x||_1 = \sum_{i=1}^{n} |x_i|$

22.1.4 Zero-Norm

counts the number of non-zero entries.

22.2 Matrix Norms

22.2.1 Nuclear Norm

$||.||_*$ sum of singular values

22.2.2 Frobenious-Norm

$sqrt(sum(sum(A.^2)))$

22.2.3 Spectral Norm

Largest singular value if square
$||A||_2 = \sigma_{max}(A)$ Is equals to the 2-Norm

22.2.4 Induced Matrix Norms

$||A|| = max\left(\frac{||Ax||}{||x||}\right)$

22.3 Orthogonality

22.3.1 Vectors

inner (scalar) product $\langle\ .\ ,\ .\ \rangle = 0$

22.3.2 Matrices

quadratic, values are in $\mathbb{R}$, $Q^T = Q^{-1}$

22.3.3 Functions

$f(x)$ orth. to $g(x)$ if $0 = \int f(x)g(x)dx$

22.3.4 Coherence

$m(U) = max_{i,j:i \neq j}|u_i^T u_j|$

22.3.5 Convexity

$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y)$

23 Differentials

Chain rule: $f(g(x))\frac{d}{dx} = f'(g(x)) \cdot g'(x)$

23.1 Vector/Matrix differentiation

$\frac{d}{dx}f(x) = \left[\frac{\partial f}{\partial x_1}, ..., \frac{\partial f}{\partial x_n}\right]$, $\frac{d}{dx}(b^T x) = \frac{d}{dx}(x^T b) = b$, $\frac{d}{dx}(x^T x) = \frac{d}{dx}(x^T x) = 2x$, $\frac{d}{dx}(x^T A x) = (A^T + A)x$

24 Probability

24.1 Notation

$Pr\{...\}$ Probability of an event

$P(x)$ Probability mass function (Verteilungsfunktion)

$p(x)$ Probability density function (Dichtefunktion)

$P(X,Y) = P(X|Y) \cdot P(Y) = P(Y|X) \cdot P(X)$

Bayes: $P(X|Y) = \frac{P(Y|X) \cdot P(X)}{P(Y)}$

25 Collaborative Filtering with SVD

Init/Set values to predict in M to be the avg value.

$M = U \cdot D \cdot V^T$

U = Row-to-Concept affinity

V = Column-to-Concept affinity

D = expressiveness of each concept in the data

25.1 Add new row (User Bob)

$M_{Bob} = U_{Bob} \cdot D \cdot V^T => M_{Bob} \cdot V \cdot D^-1 = U_{Bob}$

26 K-Means $X = U \cdot Z$

26.1 Hard Assignment

Minimize cost function: $J(U,Z) = ||X - UZ||_f^2 = \sum_{n=1}^N \sum_{k=1}^K z_{k,n} ||x_n - u_k||_2^2$

Equ. holds only iff $z_{k,n}$ is boolean and $sum(Z_n) = 1$

26.1.1 Algorithm

Step 1: Cluster Assignment Hard-Assign to Cluster where $||X_n - U_i||_2^2$ is minimal

$k^*(x_n) = argmin\{||x_n - u_1||_2^2, ..., ||x_n - u_k||_2^2, ..., ||x_n - u_K||_2^2\}$

Step 2: Centroid update u_k: Sum up data points associated to k-th centroid and average.

$$u_k = \frac{\sum_{n=1}^N z_{k,n} \cdot x_n}{\sum_{n=1}^N z_{k,n}}$$

26.1.2 Convergence K-Means

Step 1 minimizes J because it sets z_k, where $||X_n - U_i||_2^2$ is minimal

Step 2 minimizes J because $u_k = \dfrac{\sum_{n=1}^N z_{k,n} \cdot x_n}{\sum_{n=1}^N z_{k,n}}$ is the derivative of J with respect to u_k:

$\dfrac{\partial J}{\partial u_k} = \dfrac{\partial \sum_{n=1}^N z_{k,n} ||x_n - u_k||_2^2}{\partial u_k} = \sum_{n=1}^N z_{k,n} \left[\dfrac{\partial (x_{1,n} - u_{1,k})^2}{\partial u_{1,k}}, ..., \dfrac{\partial (x_{d,n} - u_{d,k})^2}{\partial u_{d,k}}\right]^T = -2\sum_{n=1}^N z_{k,n}(x_n - u_k)$ => solve for u_k , $\dfrac{\partial J^2}{u_k^2} > 0 => J$ does not increase after centroid update.

26.2 Estimate K - $\kappa(.)$ num. free param.

26.2.1 AIC

$= -lnp(X|.) + \kappa(U,Z)$

26.2.2 BIC

$= -lnp(X|.) + \frac{1}{2}\kappa(U,Z)ln(N)$

26.3 EM with GMM Gaussian Mixture Model

- $p(x) = \sum_{k=1}^K \pi_k \ \ p(x|\theta_k)$
- $\sum_{k=1}^K \pi_k = 1$ Each column sums up to 1
- Gaussian Distr.: μ: Expectation, $\sigma^2 = variance$, $\sigma = stddev$
- introduce latent variable γ in the E-Step and marginalize away in the M-Step
- $\gamma(z_{k,n})$ is the prob. of x_n beeing ass. to cluster k

26.3.1 E-Step

Evaluate Responsibilities

$\gamma(z_{k,n}) := \mathbb{E}[z_{k,n}] = \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_j \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)}$

26.3.2 M-Step

Re-Estimate model parameters

$N_k = \sum_{n=1}^N \gamma(z_{k,n})$

$u_k^{new} = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{k,n}) x_n$, $\pi_k^{new} = \frac{N_k}{N}$

27 Non-negative matrix factorisation

$X \in \mathbb{R}+^D$

Similar to k-means:

1. Init U, Z (random positive values)
2. Iterate
3. Update U: $\tilde{X} = UZ$ $X = UZ => XZ' = UZZ'$ XZ'/UZZ' is a coefficient matrix in $\mathbb{R}^+$

   $u_d k_{new} = u_d k \cdot ((XZ')/(UZZ'))$
4. Update Z: $X = UZ$; $U'X = U'UZ$; $z_d k_{new} < -z_d k * ((UX)/(U'UZ))$

27.1 Deduction

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

Lagrangian $L(U,Z,\alpha,\beta) = J(U,Z) - tr(\alpha U^T) - tr(\beta Z^T)$

where tr(.) is the trace of a matrix

27.2 Kullback-Leibler Divergence

$D_{KL}(P||Q) = \sum_{x \in X} P(x) \log\left(\dfrac{P(x)}{Q(x)}\right)$ KL-divergence of X and UZ for pLSI: $\min_{U,Z} \sum_{d=1}^D \sum_{n=1}^N x_{dn} \log\left(\dfrac{x_{dn}}{(UZ)_{dn}}\right)$ s.t. $\sum_{d=1}^D u_{dk} = 1 \forall k, \sum_{d,n} z_{kn} = 1, u_{dk} \geq 0, z_n \geq 0$

28 Role Based Access control - RBAC

Model with $\beta = (p\{u_{dk} = 0\})^{D\times K}$

SAC: $p(X|\beta, Z) = \prod_{n,d}(1-\beta_{dk_n})^{x_{dn}}(\beta_{dk_n})^{(1-x_{dn})}$

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

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

Deviating Ones:$d1 := \frac{|\{(i,j)|\hat{x}_{i,j} = x_{i,j} = 1, x_{i,j} = 0\}|}{|\{(i,j)|x_{i,j} = 1\}|}$

Deviating Zeros:$d0 := \frac{|\{(i,j)|\hat{x}_{i,j} = x_{i,j} = 0, x_{i,j} = 1\}|}{|\{(i,j)|x_{i,j} = 0\}|}$

29 Compressive Sensing

- x is a D-Dimensional measurement
- x is sparse in some orthonormal basis U, $x = U \cdot z$
- instead of saving x we save y with dim. $M << D$
- define any orthonormal basis U $(D \times D)$
- define W $(M \times D)$
- $y = Wx = WUz := \Theta z$
- $\Theta = W \cdot U$
- Store y: $Wx => y$
- Restore x: $y = \Theta \cdot z$, find most sparse matrix z
  - arg min z : $||z||_0$ s.t. $\Theta z = y$ (matching persuit)
  - $x = U \cdot z$

30 Sparse Coding

30.0.1 Matching Pursuit

Exact Recovery Conditions

$K < \frac{1}{2}\left(1 + \frac{1}{m(U)}\right)$

where Coherence: $m(U) = \underset{i,j:i\neq j}{max} \; |u_i^T u_j|$

30.0.2 Overcomplete Dicts.

- increasing overcompleteness
- increases (potentially) to a certain point sparse coding (gets sparser)
- increases linear dependence between atoms
- Solve: $argmin||z||_0 s.t. x = Uz$

31 Dictionary Learning

$X = U \cdot Z$ alternate betw. Coding and Dict. update step

- Update Z to be as sparse as possible (with MP)

Dictionary Update Step

- $U_{new} = argmin \; U||X - UZ||_F^2$
- Update one dictionary item U_l at a time
  - write $U \cdot Z$ as sum omit index l: $\sum_{i\neq l} U_i \cdot Z_i^T$
  - Residual $R_l = X - \left(\sum_{i\neq l} U_i \cdot Z_i^T\right)$
  - $=> R_l = U_l \cdot Z_l^T$ (where R_l, Z_l^T fix)
  - $R_l = UDV^T$ update U_l with first column of U

(hint: write SVD as SUM and you will see)

32 Robust PCA R-PCA

$X = L + S$ (L is low rank, S is sparse)

relax the problem to:

minimize $||L||_* + \lambda \cdot ||S||_1$ subject to $L + S = X$

32.1 Convex optimisation

minimize $f(x)$ subject to

$g_i(x) \leq 0, i = 1, ..., m$ inequality constr.

$h_i(x) = 0, i = 1, ..., p$ equality constr.

Create the Lagrangian

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

Lagrange dual function: $d(\lambda, \nu) = \inf_x L(x, \lambda, \nu)$

Lagrange dual problem: max. $d(\lambda, \nu)$ subj. to $\lambda \geq 0$

32.2 ADMM - Alternating Direction Method of Multipliers

32.2.1 Alternate Direction

Lagrangian: $L(x, \nu)$

Dual Function: $d(\nu) = \inf_x L(x, \nu)$

Dual Problem: maximize $d(\nu)$

Recover optimal x: $x^* \in argmin_x L(x, \nu^*)$

Gradient Method: $\nu^{k+1} = \nu^k + \alpha^k \nabla d(\nu^k)$

$\nabla d(\nu^k) = f(\tilde{x})$, where $\tilde{x} = argmin_x L(x, \nu^k)$

32.2.2 Dual decomposition

1. if $f(x)$ is separable into $f_1(x_1) + f_2(x_2) + ... + f_n(x_n)$ then $L(x, v)$ is separable so we can split the x-minimisation step
2. Method of multipliers
3. create augmented lagrange by adding a penalty function $\frac{\rho}{2}||.||_2^2$
4. add more penalty for violating constraints, leads to convergence under far more general condition

32.2.3 ADMM in short

minimize $f(x) + p(z)$ s.t. $Ax + Bz = c$ Augm. Lagrange: $L_p(x, z, \nu) = f(x) + p(z) + \nu^T(Ax + Bz - c) + \frac{\rho}{2}||Ax + Bz - c||_2^2$

ADMM:

$x^{k+1} := argmin_x L_\rho(x, z^k, \nu^k)$

$z^{k+1} := argmin_z L_\rho(x^{k+1}, z, \nu^k)$

$\nu^{k+1} := \nu^k + \rho(Ax^{k+1} + Bz^{k+1} - c)$

32.2.4 PCP Recovery Condition

Probability. $1 - \mathcal{O}(n^{-10})$ with $\lambda = \frac{1}{sqrt(n)}$