Basics

General

Cauchy-Schwartz Inequality: $|u \cdot v| \leq \|u\|\|v\|$
Gauss distribution:
$f_{\mathbf{X}}(x_1, \ldots, x_k) = \frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)}{\sqrt{(2\pi)^k|\boldsymbol{\Sigma}|}}$

Probability Basics

$\Pr[a \leq X \leq b] = \int_a^b f_X(x)\,dx$
Cumulative distribution function: $F_X(x) = \mathrm{P}(X \leq x) = \int_{-\infty}^{x} f_X(t)\,dt$
Bayes: $p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)} = p(\text{model}|\text{data}) = \frac{p(\text{data}|\text{model})p(\text{model})}{p(\text{data})}$
Posterior probability $\propto$ Likelihood $\times$ Prior probability
Markov inequality $\mathbb{P}(g(X) \geq r) \leq \frac{\mathbb{E}[g(X)]}{r}$
Chebyshev inequality $\mathbb{P}(|X-\mu| \geq \epsilon) \leq \frac{\sigma^2}{\epsilon^2}$
Jensen inequality $\varphi(\mathrm{E}[X]) \leq \mathrm{E}[\varphi(X)]$ where φ is a *convex* function. E.g. $log(x)$ if $x > 0$
Law of large numbers $\mathbb{P}(|\frac{S_n}{n} - \mu| \geq \epsilon) \leq \frac{\sigma^2}{n \cdot \epsilon^2} \to 0$ if $n \to 0$
Bound on disjunction: $\max \mathbb{P}(A_1), \ldots, \mathbb{P}(A_n) \leq \mathbb{P}(E) \leq \min 1, \mathbb{P}(A_1) + \cdots + \mathbb{P}(A_n)$
Glivenko-Catelli theorem: $\mathbb{P}(\sup_x |F_n(x) - F(x)| \to 0) = 1$ if $n \to 0$.
Completing the square: If we want to go from $ax^2 + bx + c = 0$ to $a(x+d)^2 + e = 0$, we get d by $d = \frac{b}{2a}$ and $e = c - \frac{b^2}{4a}$

Derivatives

$\frac{\partial}{\partial \mathbf{x}}(\mathbf{b}^\top\mathbf{x}) = \frac{\partial}{\partial \mathbf{x}}(\mathbf{x}^\top\mathbf{b}) = \mathbf{b} \quad \frac{\partial}{\partial \mathbf{x}}(\mathbf{x}^\top\mathbf{x}) = 2\mathbf{x}$
$\frac{\partial}{\partial \mathbf{x}}(\mathbf{x}^\top\mathbf{A}\mathbf{x}) = (\mathbf{A}^\top + \mathbf{A})\mathbf{x} \quad \frac{\partial}{\partial \mathbf{x}}(\mathbf{b}^\top\mathbf{A}\mathbf{x}) = \mathbf{A}^\top\mathbf{b}$
$\frac{\partial}{\partial \mathbf{X}}(\mathbf{c}^\top\mathbf{X}\mathbf{b}) = \mathbf{c}\mathbf{b}^\top \quad \frac{\partial}{\partial \mathbf{X}}(\mathbf{c}^\top\mathbf{X}^\top\mathbf{b}) = \mathbf{b}\mathbf{c}^\top$
$\frac{\partial}{\partial \mathbf{x}}(\|\mathbf{x}-\mathbf{b}\|_2) = \frac{\mathbf{x}-\mathbf{b}}{\|\mathbf{x}-\mathbf{b}\|_2} \quad \frac{\partial}{\partial \mathbf{x}}(\|\mathbf{x}\|_2^2) = \frac{\partial}{\partial \mathbf{x}}(\mathbf{x}^\top\mathbf{x}) = 2\mathbf{x}$
$\frac{\partial}{\partial \mathbf{X}}(\|\mathbf{X}\|_F^2) = 2\mathbf{X} \quad \frac{\partial}{\partial \mathbf{X}}(\|\mathbf{X}\|_2) = \frac{x}{\|\mathbf{X}\|_2}$

$\frac{\partial}{\partial \mathbf{x}}(\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2) = \mathbf{2}(\mathbf{A}^\top\mathbf{A}\mathbf{x} - \mathbf{A}^\top\mathbf{b}) \quad \frac{\partial}{\partial x}\frac{g(x)}{h(x)} = \frac{g'(x)h(x) - g(x)h'(x)}{h(x)^2}$
$\frac{\partial}{\partial (w_i^T x)} softmax(j) = softmax(j)(\delta_{ij} - softmax(i))$
For functionals, we have the Taylor approximation of: $F[x + \epsilon\eta] = F[X] + \epsilon \int \frac{\partial F}{\partial x}(z)\eta(z)dz + O(\epsilon^2)$
$cosh' = sinh \; sinh' = cosh$

Maximization Basics

Lagrange multipliers
$\mathcal{L}(x, y, \lambda) = f(x, y) - \lambda \cdot g(x, y)$
solve $\nabla_{x,y,\lambda}\mathcal{L}(x, y, \lambda) = 0$

Calculus of Variations
Goal: find the maximum of a functional.
$J(x + \delta x) - J(x) = 0$
Given: functional of form $J[f] = \int_a^b L[x, f(x), f'(x)]dx$
change $\delta J = \int_a^b \frac{\delta J}{\delta f(x)}\delta f(x)dx$

functional derivative: $\frac{\delta J}{\delta f(x)} = \frac{\partial L}{\partial f} - \frac{d}{dx}\frac{\partial L}{\partial f'}$
complete data likelihood: $p(x, k|\mu, \eta)$
incomplete data likelihood: $p(x|\mu, \eta)$

EM-Algorithm
given: $p(\mathbf{X}, \mathbf{Z}|\theta)$ with observed variables $\mathbf{X}$ and latent variables $\mathbf{Z}$, parameters θ *goal:* maximize likelihood function $p(\mathbf{X}|\theta)$ wrt. θ.

1. choose initial setting for $\boldsymbol{\theta}^{old}$
2. **E-step**: evaluate $Q_i(z^{(i)}) := p(z^{(i)}|x^{(i)}; \theta)$
3. **M-step**: $\theta := argmax_\theta \sum_i \sum_{z^{(i)}} Q_i(z^{(i)}) log \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})}$
4. check if converged.

Gaussian Mixture Models (GMM)

For GMM let $\boldsymbol{\theta}_k = (\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$; $p_{\theta_k}(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k)$
Mixture Models: $p_\theta(\mathbf{x}) = \sum_{k=1}^K \pi_k p_{\theta_k}(\mathbf{x})$
Assignment variable (generative model): $z_{ij} \in \{0, 1\}$, $\sum_{j=1}^k z_{ij} = 1$
$\Pr(z_k = 1) = \pi_k \Leftrightarrow p(\mathbf{z}) = \prod_{k=1}^K \pi_k^{z_k}$
Complete data distribution:
$p_\theta(\mathbf{x}, \mathbf{z}) = \prod_{k=1}^K (\boldsymbol{\pi}_k p_{\theta_k}(\mathbf{x}))^{z_k}$
Posterior Probabilities:
$\Pr(z_k = 1|\mathbf{x}) = \frac{\Pr(z_k=1)p(\mathbf{x}|z_k=1)}{\sum_{l=1}^K \Pr(z_l=1)p(\mathbf{x}|z_l=1)} = \frac{\boldsymbol{\pi}_k p_{\theta_k}(\mathbf{x})}{\sum_{l=1}^K \boldsymbol{\pi}_l p_{\theta_l}(\mathbf{x})}$ **Likelihood of observed data X**:
$p_\theta(\mathbf{X}) = \prod_{n=1}^N p_\theta(\mathbf{x}_n) = \prod_{n=1}^N \left(\sum_{k=1}^K \pi_k p_{\theta_k}(\mathbf{x}_n)\right)$
Max. Likelihood Estimation (MLE):
$\text{argmax}_\theta \sum_{n=1}^N \log\left(\sum_{k=1}^K \pi_k p_{\theta_k}(\mathbf{x}_n)\right)$
$\geq \sum_{n=1}^N \sum_{k=1}^K q_k[\log p_{\theta_k}(\mathbf{x}_n) + \log \pi_k - \log q_k]$
with $\sum_{k=1}^K q_k = 1$ by Jensen Inequality.

Generative Model

1. sample cluster index $j \sim Categorical(\pi)$
2. given j, sample data $x \sim \text{Normal}(\mu_j, \Sigma_j)$

Expectation-Maximization (EM) for GMM

E-Step:
$\Pr[z_{k,n} = 1|\mathbf{x}_n] = q_{k,n} = \frac{\boldsymbol{\pi}_k^{(t-1)}\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k^{(t-1)}, \boldsymbol{\Sigma}_k^{(t-1)})}{\sum_{j=1}^K \boldsymbol{\pi}_j^{(t-1)}\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j^{(t-1)}, \boldsymbol{\Sigma}_j^{(t-1)})}$
M-Step: $\boldsymbol{\mu}_k^{(t)} := \frac{\sum_{n=1}^N q_{k,n}\mathbf{x}_n}{\sum_{n=1}^N q_{k,n}}$, $\boldsymbol{\pi}_k^{(t)} := \frac{1}{N}\sum_{n=1}^N q_{k,n}$
$\Sigma_k^{(t)} = \frac{\sum_{n=1}^N q_{k,n}(\mathbf{x}_n - \boldsymbol{\mu}_k^{(t)})(\mathbf{x}_n - \boldsymbol{\mu}_k^{(t)})^\top}{\sum_{n=1}^N q_{k,n}}$

Gaussian distribution

Std. deviation σ, mean μ: $f(x) = \frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}}$
Cov. matrix Σ: $f(x) = \frac{1}{(2\pi)^{\frac{D}{2}}\sqrt{|\Sigma|}}e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}$

Information Theory

Mutual information:
measures the amount of information that can be obtained about one random variable by observing another
$I(X;Y) = \mathbb{E}_{X,Y}[SI(x,y)] = \sum_{x,y} p(x,y)\log\frac{p(x,y)}{p(x)\,p(y)}$
$I(X;Y) = H(X) - H(X|Y)$.
$I(X;Y) = I(Y;X) = H(X) + H(Y) - H(X,Y)$.
$I(X,Y;Z) = I(X;Z) + I(Y;Z|X)$
chain rule for information: $I(X;Y,Z) = I(X;Z) + I(X;Y|Z)$
$I(X;Y) = \mathbb{E}_{p(y)}[D_{\mathrm{KL}}(p(X|Y=y)\|p(X))]$.
Information Processing Inequality: Let $X \to Y \to Z$ be a Markov Chain, then $I(x,z) \leq I(x,y)$

Kullback-Leibler divergence

- quantifies coding cost describing data with probability distribution q when true distribution is p.
- KL divergence is positive semidefinite. $D_{\mathrm{KL}} \geq 0$. Use Jensen's inequality to prove it.

way of comparing two distributions: a "true"probability distribution $p(X)$, and an arbitrary probability distribution $q(X)$.
or: ünnecessary surprise". **not** symmetric!
q(X) is the distribution underlying some data, when, in reality, p(X) is the correct distribution, the Kullback–Leibler divergence is the number of average additional bits per datum necessary for compression.
$D_{\mathrm{KL}}(p(X)\|q(X)) = \sum_{x\in X} -p(x)\log q(x) - \sum_{x\in X} -p(x)\log p(x) = \sum_{x\in X} p(x)\log\frac{p(x)}{q(x)}$
continuous case: $D_{\mathrm{KL}}(P\|Q) = \int_{-\infty}^{\infty} p(x)\log\frac{p(x)}{q(x)}dx$

Dimensionality Reduction

Principal Component Analysis (PCA)

Idea
dimensionality reduction with orthogonal projection of D-dimensional data onto lower M-dimensional linear space (*principal subspace*). Variance is maximized.

Procedure
1. calculate mean $\bar{x} = \frac{1}{N}\sum_{n=1}^N x_n$ and covariance matrix **S** of data set
2. find the M eigenvectors corresponding to the M largest eigenvalues.
3. data projected on these eigenvectors have largest variance.

Probabilistic Principal Component Analysis (PPCA)

Idea
$\mathbf{x} = \mathbf{W}\mathbf{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon}$, with $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I})$ and $\epsilon\ \mathcal{N}(0, \sigma^2\mathbf{I})$
$p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}|\mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \sigma^2\mathbf{I})$

Procedure
1. calculate mean $\bar{x} = \frac{1}{N}\sum_{n=1}^N x_n$ and covariance matrix **S** of data set
2. find the M eigenvectors corresponding to the M largest eigenvalues.
3. data projected on these eigenvectors have largest variance.

Advantages
- computationally efficient $\to$ no need to evaluate covariance matrix as intermediate step
- allows us to deal with missing data in data set

Locally Linear Embedding

N real valued vector $\vec{X}_i$ of dimension D.
Assumption: data lies close on locally linear patch of manifold.
Construct each data point form its K nearest (e.g. euclidean distance) neighbors.
Reconstruction error: $\mathcal{E}(W) = \sum_i |\vec{X}_i - \sum_j W_{ij}\vec{X}_j|$
where W_{ij} summarize the contribution of the jth data point to the ith reconstruction. Invariant under uniform translation of points x_i

LLE Algorithm:

1. Compute the neighbors of each data point, $\vec{X}_i$.
2. Compute the weights W_{ij} that best reconstruct each data point $\vec{X}_i$ from its neighbors, minimizing the cost $\mathcal{E}(W)$ by constrained linear fits. We have $w_{ij} = \frac{\sum_k C_{jk}^{(-i)}}{\sum_{lk} C_{lk}^{(-i)}}$, where $C_{jk}^{(i)} = (x_i - x_j)^T(x_i - x_k)$
3. Compute the vectors $\vec{Y}_i$ best reconstructed by the weights W_{ij}, minimizing the quadratic form in $\Phi(Y) = \sum_i |\vec{X}_i - \sum_j W_{ij}\vec{Y}_j|^2$ by its bottom nonzero eigenvectors. The embedding vectors are given by the eigenvectors corresponding to the d+1 smallest eigenvalues of $M = (1-W)^T(1-W)$

Maximum Entropy Inference

Maximizing entropy yields least biased inference method $\to$ maximally non-committal wrt. missing data.

Maximum entropy distribution

Idea
Outcomes $\omega_i \in \Omega$, determine probabilities $p_i = \mathbb{P}(\omega_i)$. Given constraints/moments $\mu_i = \mathbb{E}[r_j] = \sum_{i=1}^n r_j(\omega_i)p_i$, where $r_j : \Omega \to \mathbb{R} j = 1, 2, ..., n$ are different functions defined over Ω.

Gibbs distribution
minimal sesitivity to changes in constraint moments μ_i
$p(x) = \frac{1}{Z}\exp(-\sum_{j=1}^m \lambda_j r_j(x))$ where m is the number of constraints.

Examples:

Extremality of Gauß & Laplace Distribution

Question: Which distribution with $\mathbb{E}\{x\} = 0,\ \mathbb{E}\{x^2\} = \sigma^2$ maximizes the entropy?

Conditions:
$$p(x) \geq 0 \qquad \int_{-\infty}^{\infty} xp(x)dx = 0$$
$$\int_{-\infty}^{\infty} p(x)dx = 1 \qquad \int_{-\infty}^{\infty} x^2p(x)dx = \sigma^2$$

Method of Lagrange Multipliers:

$$\mathcal{L} = \int_{\mathbb{R}} \left(-\ln p(x) + \lambda_1 + \lambda_2 x - \lambda_3 x^2\right) p(x)dx$$
$$\frac{\delta\mathcal{L}}{\delta p} = -\ln p - 1 + \lambda_1 + \lambda_2 x - \lambda_3 x^2 = 0 \quad \Rightarrow$$
$$p(x) = \exp\left(-1 + \lambda_1 + \lambda_2 x - \lambda_3 x^2\right)$$

Inserting the constraints yields:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma}}\exp\left(-\frac{x^2}{2\sigma^2}\right)$$

Extremality of Laplace Distribution

Let the random variable $X \sim \mathcal{N}(0, \sigma^2)$ be normal distributed. The conditional distribution $\mathcal{N}(x|\sigma^2)$ assumes a different variance in each experiment. The variance is distributed according to $\sigma^2 \sim \mathbf{P}(\sigma^2)$.

$$\mathbf{P}\{x\} = \int_0^\infty \mathcal{N}(x|\sigma^2)\mathbf{P}(\sigma^2)d\sigma^2$$

Extreme Experimental Conditions

The variance is maximally indetermined, i.e.,

$$\mathbf{P}(\sigma^2) \geq 0, \quad \int \mathbf{P}(\sigma^2)d\sigma^2 = 1, \quad \int \sigma^2\mathbf{P}(\sigma^2)d\sigma^2 = 2\Delta^2$$

Entropy Maximization: $\mathbf{P}(\sigma^2) = \frac{1}{2\Delta^2}\exp\left(-\frac{\sigma^2}{2\Delta^2}\right)$

Laplace distribution maximizes entropy with expected variance

Insert likelihood and prior

$$\begin{aligned}\mathbf{P}(X) &= \frac{1}{\sqrt{2\pi}2\Delta^2}\int_0^\infty \frac{1}{\sigma}\exp\left(-\frac{x^2}{2\sigma^2}\right)\exp\left(-\frac{\sigma^2}{2\Delta^2}\right)d\sigma^2 \\ &= \frac{1}{\Delta}\exp\left(-\frac{|x|}{\Delta}\right)\underbrace{\int_0^\infty \frac{1}{\sqrt{2\pi}\Delta}\exp\left(-\frac{1}{2}\left(\frac{\sigma}{\Delta} - \frac{|x|}{\sigma}\right)^2\right)d\sigma}_{1/2} \\ &= \frac{1}{2\Delta}\exp\left(-\frac{|x|}{\Delta}\right)\end{aligned}$$

where we have used the formula $\int_0^\infty f((\frac{x}{a} - \frac{b}{x})^2)dx = a\int_0^\infty f(y^2)dy$.

$Z = \int_x \exp(-\sum_{j=1}^m \lambda_j r_j(x))dx$

Gibbs Free Energy
$G(p) = \sum_x p(x)E(x) + \frac{1}{\beta}\sum_x p(x)logp(x) = \frac{1}{\beta}D_{KL}(p||p_\beta) + F(\beta)$ Minimising $G(p)$ minimises the expected cost and maximises the entropy of $p(x)$.

Derive least sensitive distributions
1. determine all r_j from constraints.
2. calculate Z and $p(x)$ from Gibbs distribution
3. calculate all moments and set equal to constraints.

For a general cost function:
$$\mathbf{P}(\mathbf{c}) = \frac{e^{-\beta\mathcal{R}(c)}}{\sum_{c'\in\mathcal{C}e^{-\beta\mathcal{R}(c')}}} = \frac{e^{-\beta\mathcal{R}(c)}}{Z} = \exp(-\beta(\mathcal{R}(c) - \mathcal{F}))$$
where $\mathcal{F}(\beta) = -\frac{1}{\beta}\log(Z)$ is known as „free energy".
Free energy for any distribution: $\mathcal{F} = -\frac{1}{\beta}S(\mathbf{P}) + \mathbb{E}_{\mathbf{P}}\mathcal{R}$

Entropy

H_X of a discrete random variable X is a measure of the amount of *uncertainty* associated with the value of X when only its distribution is known
$H(X) = \mathbb{E}_X[I(x)] = -\sum_{x\in\mathbb{X}} p(x)\log p(x)$
$H(Y|X) = \sum_{x,y} p(x,y)log(p(y|x)$
Chain rule: $H(X,Y) = H(X) + H(Y|X)$

$H(X,Y|Z) = H(X|Z) + H(Y|X,Z)$

Maximum Entropy Clustering

Idea
Determine probabilistic centroids y_α and probabilistic assignments $P_{i\alpha}$ of i-th object to cluster α. Data $\mathcal{X}$ and labelings c are r.v.

Procedure
1. Define posterior probability distribution $P(c|\mathcal{X}, \mathcal{Y})$, $c \in \mathcal{C}$, constraint $\mathbb{E}_{P(c|\mathcal{X},\mathcal{Y})}\mathcal{R}(c, \mathcal{X}, \mathcal{Y}) = \mu$ where μ is a constant.
2. find centroid conditions and assignments $P_{i\alpha}$ by maximizing entropy wrt. $\mathcal{Y} \to \frac{\partial}{\partial \mathbf{y}_\alpha}H(P(c|\mathcal{X}, \mathcal{Y})) = \mathbf{0}$

Most agnostic $P(c|\mathcal{X}, \mathcal{Y})$ is Gibbs distribution.

$$P(c|\mathcal{X},\mathcal{Y}) = \frac{\exp(-\mathcal{R}(c,\mathcal{X},\mathcal{Y})/T)}{\sum_{c'\in\mathcal{C}}\exp(-\mathcal{R}(c',\mathcal{X},\mathcal{Y})/T)}$$

Remark: If the cost function is linear in individual costs, posterior can be factorized!

Clustering distributional data

Idea

„Close" objects have similar feature distributions. → cluster via similarity.
Distributional data encoded in dyads (pair $(x_i, y_i) \in \mathcal{X}\times\mathcal{Y}$): n different objects $\{x_i\} \in \mathcal{X}$, m possible feature values $\{y_i\} \in \mathcal{Y}$. Data is set of l observations $\mathcal{Z} = \{(x_{i(r)}, y_{j(r)})\}_{r=1}^{l}$.
Likelihood of data $\mathcal{Z}$ $\mathcal{L}(\mathcal{Z}) = \Pi_{i\le n}\Pi_{j\le m}P((i,j)|c(i),Q)^{l\hat{P}(i)}$
where $l\hat{P}(i)$ = # object i exhibits value j and $\hat{P}(i)$ = empirical probability finding object i with feature value j.

Parametric distributional clustering

Density estimation is assumed to be Gaussian $g_a(y)$.
$p(y|\nu) = \sum_{a\le s} g_a(y)p(a|\nu)$ where $p(a|\nu)$ mixture coefficients, s number of mixture components.
Remark: Often feature values are restricted to specific domains, mixture components → rectified. Complete data likelihood is given by: $p(x,M|\theta) = \prod_{i=1}^{n}\sum_{r=1}^{k} M_{ir}p_r p(x_i|r)$, where $M_{ir}\in\{0,1\}$ and $\sum_r M_{ir} = 1$
Assuming that the observations x_{ij} are taken from domain I, which is equipartitioned into I_p for $p = 1..m$, we have $p(x_{ij}\in i_p|\alpha) =: G_\alpha(p)$. Thus: $p(x,M|\theta) = \prod_{i=1}^{n}\prod_{r=1}^{k}[p_r\prod_{p=1}^{m}\sum_{\alpha=1}^{l}p(\alpha|r)G_\alpha(p)^{n_{ip}}]^{M_{ir}}$, where n_{ip} counts how many observations x_{ij} from object i fell into bin I_p
The connection to the Info Bottleneck idea is given by:
X: decision to pick an object i uniformly.
$\hat{X}$: encoding of object i by cluster r.
Y: Relative frequency of observing bin p.
Seeing that $I(X,\hat{X}) = H(X\hat{X}) = -\sum_{r=1}p_r log p_r$, i.e. $I(X,\hat{X}) = log k$ for $p_r = \frac{1}{k}$
We also have $H(Y|\hat{X}) = -\sum_{i=1}^{n}\sum_{r=1}^{k}\sum_{p=1}^{m}M_{ir}\frac{n_{ip}}{n_i}\frac{1}{n}log\sum_\alpha p(\alpha|r)G_\alpha(p)$
Putting it all together we get that $I(X,\hat{X}) - \lambda I(\hat{X},Y) = -\sum_r(\frac{1}{n}\sum_i M_{ir})log p_r - \lambda\sum_{i,r}\frac{1}{n}M_{ir}\sum_p\frac{n_{ip}}{n_i}log\sum_\alpha p(\alpha|r)G_\alpha(p) - \lambda H(Y) = -\frac{1}{n}\sum_{i,r}M_{ir}[log p_r + \frac{\lambda}{n_i}\sum_p n_{ip} log\sum_\alpha p(\alpha|r)G_\alpha(p)] - \lambda H(Y)$
By choosing the right constants and then dropping stuff that doesn't affect the minimum. That this is equal to the negative log likelihood.

Markov Chain Monte Carlo (MCMC)

Idea: sampling from a probability distribution by constructing a Markov chain that has the desired distribution as its equilibrium distribution.

Markov Chain

$P_N(x_1,...,x_N) = p_1(x1)\prod_{t=1}^{N-1}w(x_t\to x_{t+1})$
where $\{p_1(x)\}_{x\in\mathcal{X}}$ is the initial state and $\{w(x\to y)\}_{x,y\in\mathcal{X}}$ are the transition probabilities
transition probabilities must be non-negative and normalized: $\sum_{y\ in\mathcal{X}} w(x\to y) = 1$, for any $y\in\mathcal{X}$ *Eigenvalues:*
First eigenvalue is always equal to 1. The smaller λ_2 the faster the MCMC converges.
irreducible: It is possible to get to any state from any state
Detailed balance: $\pi_i P_{ij} = \pi_j P_{ji}$ or $p(x)(x\to y) = p(y)p(y\to x)$, implies stationarity!
Stationarity: $\pi\mathbf{P} = \pi$
Aperiodic: Chain should not get trapped in cycles, i.e. for all states x,y there's an integer $n(x,y)$, s.t. $\forall n > n(x,y)$, there's a path of lenght n between x and y

Metropolis-Hastings

Initialization: Choose an arbitrary point x_0 as first sample, choose arbitrary probability density $g(x|y)$ that suggests a candidate for the next sample value x. For Metropolis algorithm, g must be symmetric → $g(x|y) = g(y|x)$ e.g. a Gaussian distribution centered at y, → points closer to y more likely to be visited next → sequence of samples → random walk. function g → proposal density or jumping distribution.
Algorithm:

1. Generate a new sample x' from the old x according to a proposal PDF: $x' \sim q(\,|\,x)$
2. Calculate acceptance probability $A(x',x)$ according to quotient $Q(x',x) = \frac{p(x')q(x|x')}{p(x)q(x'|x)}$ $A(x',x) = min(1, Q(x',x))$
3. Accept x' as new sample with with probability $A(x',x)$, else keep x.

Disadvantages:

- samples are correlated. For independent samples one would have to take only every *n*-th sample.
- initial samples may follow a very different distribution. Solution: throw away first 1000 samples.
- for high dimensions: finding the right jumping distribution can be difficult, as the different individual dimensions behave in very different ways, and the jumping width must be "just right"for all dimensions at once to avoid excessively slow mixing → Gibbs Sampling.

Detailed balance requires

$$\mathbf{P}(c_2)\mathbf{P}(c_2\to c_1) = \mathbf{P}(c_1)\mathbf{P}(c_1\to c_2)$$

$$\frac{\exp(-R(c_2)/T)}{Z}1 = \frac{\exp(-R(c_1)/T)}{Z}\exp(-(R(c_2)-R(c_1))/T)$$

Pseudocode:

Metropolis Sampler for Clustering

Input: n objects, cost function $\mathcal{R}(c,\mathbf{Y},\mathbf{X})$
Output: partition c : {objects} → {clusters}

MCMC Algorithm

initialize $c(i)\in\{1,\dots,k\}$ randomly;
repeat
  draw $c'\sim Q(c)$; // $Q(c)$: proposal distribution
  $p\leftarrow\min\{\exp(-(\mathcal{R}(c',\mathbf{Y},\mathbf{X})-\mathcal{R}(c,\mathbf{Y},\mathbf{X}))/T),1\}$;
  draw $b\sim$ Bernoulli(p);
  if $b = 1$ **then**
    $c\leftarrow c'$;
  end if
  $t\leftarrow t+1$;
until converged
stop

Variants: Gibbs, importance sampling, (competitive!?)

Gibbs Sampling

given a multivariate distribution it is simpler to sample from a conditional distribution than to marginalize by integrating over a joint distribution.
Applicable if:

- conditional distribution of each variable is known and is easy (or at least, easier than joint distribution) to sample from
-

For Gibbs sampling in the Ising model we use the transition probability to a new state where only σ_i is flipped: $p_i(\sigma) = exp(-\beta max[0,\Delta E_i(\sigma)])$ simulated annealing is often used to reduce the "random walk"behavior in the early part of the sampling process

Input: n objects, cost function $\mathcal{R}(c,\mathbf{X})$
Output: partition c : objects → clusters

Gibbs Sampler Algorithm

initialize $c(i)\in\{1,\dots,k\}$ randomly;
repeat
  draw $i\in\{1,\dots,n\}$ randomly; // site selection
  $c(i)\sim\mathbf{P}(c(i)|c(1),\dots,c(i-1),c(i+1),\dots,c(n))$;
  $t\leftarrow t+1$;
until joint distribution $\mathbf{P}(c(1),\dots,c(n))$ is converged
stop

Heat-Bath Algorithm

Same as metropolis but with acceptance probability:
$p(\sigma_{new}) = exp(-\beta E(\sigma_{new}))/(exp(-\beta E(\sigma_{new}) + exp(-\beta E(\sigma_{old})))$

Simulated Annealing
Idea

- Markov process samples solution from solution space Ω
- cost function $\mathcal{H}:\Omega\to\mathbb{R}, \omega\in\Omega$ denotes admissible solution.
- solutions accepted/rejected according to Metropolis algorithm: decreased cost → accepted, increased cost → accepted with probability $exp(-\Delta\mathcal{H}/T)$ where $\Delta\mathcal{H} = \mathcal{H}(\omega_{new}) - \mathcal{H}(\omega_{old})$. T is the computational temperature.
- reduce gradually the temperature during search proces → force system into solution with low costs. → random walk in solution space.
- annealing schedule, i.e. the βs, could be linear, quadratic of logarithmic for instance.

Markov process with converges to equilibrium:
$\mathbf{P^{Gb}}(\omega) = \exp(-(\mathcal{H}(\omega)-\mathcal{F}(\mathcal{H}))/T)$
where $\mathcal{F}(\mathcal{H})) = -Tlog\sum_{\omega'\in\Omega}\exp(-\mathcal{H}(\omega')/T)$

Remarks

- Temperature formally ≈ Lagrange parameter → constraint on expected costs. $\langle H\rangle = \sum_{\omega\in\Omega}P^{Gb}(\omega)\mathcal{H}(\omega)$
- Gibbs free energy related to expected cost via entropy $\mathcal{S}(P^{Gb}) = -\sum_{\omega\in\Omega}P^{Gb}(\omega)\log(P^{Gb}(\omega)) = \frac{1}{T}\langle H\rangle - \frac{1}{T}\mathcal{F}(\mathcal{H})$

Parallel Tempering

Two Monte Carlo simulation were to run in parallel, one at high and one at a low temperature. At certain steps we swap the configurations of the two runs. This way both runs have access to low and high temperature configuration improving convergence to a global optimum. We swap the configurations after each epoch.

Deterministic Annealing

Idea

deterministic variant of *Simulated annealing*.
The critical temperature, at which a cluster will split is: $T_c = 2\lambda_{max}$ where λ_{max} is the largest eigenvalue of $C_{x|y_0} = \sum_x p(x|y)(x-y)(x-y)^t$

Deterministic Annealing of k-Means Clustering

INITIALIZE: assign $\forall\alpha$, $y_\alpha = \frac{1}{n}\sum_{i\le n}x_i$ and $\mathbf{P}_{i\alpha}\in(0,1] : \sum_{\nu\le k}\mathbf{P}_{i\nu} = 1$.
WHILE $T\ge T_0$ **DO**
  $\forall\alpha\neq\beta$ **IF** $\|y_\alpha - y_\beta\| < \epsilon$ **THEN** $y_\beta\leftarrow y_\beta + \xi_\beta$
  WHILE ¬ converged **DO**
    ESTIMATE $\mathbf{P}_{i\alpha} = \mathbf{P}\{c(i)=\alpha\}$ for fixed centroids
    MAXIMIZE entropy w.r.t. $\{y_\nu\}$ for fixed $\mathbf{P}_{i\alpha}$
  ENDDO
  $T\leftarrow T^{\text{new}}$
ENDDO
RETURN $\{\mathbf{P}_{i\alpha}, y_\alpha\}$
- $\xi_\beta\sim(\mathcal{N}(0,\epsilon^2))^d$

Information Bottleneck Method

Idea

input signal X should be efficiently encoded by e.g. cluster variable $\tilde{X}$ preserving relevant information about context variable Y as good as possible.
$I(X;\tilde{X}) - \lambda I(\tilde{X},Y)$

Parametric Distributional Clustering

Idea

cluster set of n objects $\mathbf{o}_i$ in k groups. Assignments encoded in $M_{i,\nu}$, $M_{i,\nu} = 1$ if $\mathbf{o}_i$ assigned to cluster ν. Enforce $\sum_{\nu\le k}M_{i,\nu} = 1$.

Graph based clustering

relations among objects not necessarily metric.

Idea

$\mathcal{O}$ is the set of vertices $\mathcal{V}$, set of edges is inferred, set of (di)similarity measures $\mathcal{D} = \{D_{ij}\}$ or $\mathcal{S} = \{S_{ij}\}$ are the weights.
Cluster defined as: $\mathcal{G}_\alpha\{\mathbf{o}\in\mathcal{O} : c(\mathbf{O}) = \alpha\}$

Correlation Clustering

Idea

agreement within cluster, disagreement between clusters maximized.
Script p. 30

Pairwise Data Clustering

cost function:

$$\mathcal{R}^{pc}(c,\mathcal{D}) = \frac{1}{2}\sum_{\nu\le k}\left(|\mathcal{G}_\nu|\sum_{(i,j)}\in\epsilon_{\nu\nu}\frac{D_{ij}}{|\epsilon_{\nu\nu}|}\right)$$

Remarks:

- dissimilarity matrix $\mathcal{D}_{ij}$ only zero for self-dissimilarity entries, $|\mathcal{G}_\nu|, |\epsilon_{\nu\nu}|$ cardinality of cluster/edges.
- cost function invariant under symmetrization and constant/additive shifts of non-diagonal elements
- metric data can always be transformed to PC data but not always vice versa

⇒ embed given nonmetric proximity data problem in vector space without changing underlying properties. → see CSE.

Constant Shift Embedding (CSE)

Embed pairwise clustering problem into vector space.

CSE algorithm

1. given dissimilarity matrix D, calculate centralized version D^c (3.13)
2. calculate matrix $S^c = -1/2D^c$ (Lemma 3)
3. calculate eigenvalues of S^c. If S^c isn't positive semidefinite, $\tilde{S} = S^c - \lambda_n\cdot I_n$ (subtract smallest eigenvalue from diagonal elements)
4. $\tilde{D}_{ij} = \tilde{S}_{ii} + \tilde{S}_{jj} + \tilde{S}_{ij}$
5. Next steps only if you want embedding vectors
6. Get $\tilde{S}^c = \frac{1}{2}\tilde{D}^c$
7. Get eigendecomposition of $\tilde{S}^c = V\Lambda V^T$ where column i of V is the eigenvector i, and Λ is the matrix with the corresponding eigenvalue on the diagonal.
8. The rows of $X_p = V_p(\Lambda_p)^{1/2}$ are the p-dimensional embedding vectors of the data. V_p are the first p eigenvectors.

Remarks:

- embedding validity shows equivalence to k-means clustering, ideas of centroids and cluster representatives can be used
- pairwise data can be denoised when transformed into vectorial data. (e.g. in preprocessing)
- minimization processes pairwise cost function or k-means problem is $\mathcal{NP}$-hard → algorithms like deterministic annealing and mean field approximation needed. For $\mathcal{R}^{km}$ it's exact, for $\mathcal{R}^{pc}$ remains an approximation.

Reconstructing the embedded vectors

1. calculate centralized dot product matrix $\tilde{S}^c = -\frac{1}{2}Q\tilde{D}Q$ from matrix of squared Eucledian distances $\tilde{D}$.
2. Express in its eigenbasis $\tilde{S}^c = V\Lambda V^T$, $V = (v_1,...,v_n)$ contains eigenvectors v_i, $\Lambda = diag(\lambda_1,...,\lambda_n)$ diagonal matrix of eigenvalues and $\lambda_1\ge...\ge\lambda_p > \lambda_{p+1} = ... = \lambda_n = 0$, at least one eigenvalue = 0, $p\le(n-1)$.
3. calculate $n\times p$ map matrix $X_p = V_p(\Lambda_p)^{1/2}$ with $V = (v_1,...,v_p)$, $\Lambda_p = diag(\lambda_1,...,\lambda_p)$ rows of X_p contain vectors x_i in p dimensional space, mutual distances are given my $\tilde{D}$

Scheme:

$D \xrightarrow{\text{CSE}} \tilde{D} \xrightarrow[\tilde{D}_{ij}=\tilde{S}_{ii}+\tilde{S}_{jj}-2\tilde{S}_{ij}]{\text{decomposition}} \tilde{S} \xrightarrow{\text{centering}} \tilde{S}^c = -\frac{1}{2}\tilde{D}^c$ step 1+2

loss-free ↓ reconstruction

$X_p = V_p(\Lambda_p)^{1/2}$ step 3

approximation ↓ & denoising

$X_t = V_t(\Lambda_t)^{1/2},\quad t < p$ step 4

Predicting Cluster Membership of New Data

$$D^{\text{new}} \xrightarrow[D^{\text{new}}_{ij}=S^{\text{new}}_{ii}+\tilde{S}^{c}_{jj}-2S^{\text{new}}_{ij}]{\text{Decomposition}} S^{\text{new}} \xrightarrow{\text{Centering}} (S^{\text{new}})^c \quad \boxed{\text{step 1}}$$

$$\Big\Vert$$

$$\underbrace{-\frac{1}{2}\left[D^{\text{new}}(I_n-\frac{1}{n}e_ne_n^\top)-\frac{1}{n}e_me_n^\top\tilde{D}(I_n-\frac{1}{n}e_ne_n^\top)\right]}$$

$$\Big\downarrow$$

$$X^{\text{new}}_p=\overbrace{(S^{\text{new}})^c}V_p\Lambda_p^{-1/2} \quad \boxed{\text{step 2}}$$

$$\text{assignment to } \Big\downarrow \text{ closest centroid in } \{y_1,\dots y_k\}$$

$$\hat{\nu}(x^{\text{new}}_p)_i=\arg\min_\nu \|(x^{\text{new}}_p)_i-y_\nu\|^2 \quad \boxed{\text{step 3}}$$

Cut

Partitioning a graph $G(V,E)$ with nodes V and edges E into disjoint sets $A,B \rightarrow$ removing edges connecting parts. Degree of dissimilarity is computed as total weight of removed edges:
$cut(A,B)=\sum_{u\in A,v\in B}w(u,v)$
Minimum cut: optimal bipartitioning of graph that minimizes cut value.
Note: Minimum cut favors cutting small sets of isolated nodes in graph. $\rightarrow$ use normalized cut.

Normalized Cut

$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)}+\frac{cut(A,B)}{assoc(B,V)}$ where $assoc(A,V)=\sum_{u\in A,t\in V}w(u,t)$ *Ncut(A,B)* can be seen as the disassociation between two groups.
measure for total normalized association within groups for a given partition: $Nassoc(A,B)=\frac{assoc(A,A)}{assoc(A,V)}+\frac{assoc(B,B)}{assoc(B,V)}$ where $assoc(A,A)$ are the total weights of edges connecting nodes within A, B respectively. Measures how tightly on average nodes are connected within group.
$Ncut(A,B)=2-Nassoc(A,B)$

Computing Optimal Partition
Grouping algorithm

1. Given image or image sequence, set up weighted graph $\mathbf{G}=(\mathbf{V},\mathbf{E})$, set weight on edge connecting two nodes to measure of similarity
2. Solve $(\mathbf{D}-\mathbf{W})\mathbf{x}=\lambda\mathbf{D}\mathbf{x}$ for eigenvectors with smallest eigenvalues.
3. use eigenvector with second smallest eigenvalue to bipartition graph by finding splitting point s.t. $Ncut$ is minimized.
4. Decide if current partition should be subdivided by checking stability of cut, make sure $Ncut$ is below prespecified value.
5. recursively repartition if necessary.

Computational complexity

- Solving eigenvalue problem for all eigenvectors takes $\mathcal{O}(n^3)$ operations
- Graphs often only locally connected, only top few eigenvectors needed, precision requirements low $\rightarrow \mathcal{O}(nm)+\mathcal{O}(mM(n))$ where $M(n)$ is the cost of matrix-vector computation $\mathbf{Ax}$ ($A=\mathbf{D}^{1/2}(\mathbf{D}-\mathbf{W})\mathbf{D}^{1/2}$).

Mean Field Approximation

Mean Field Theory

Approximate Gibbs distribution by neglecting correlations between stochastic variables $\rightarrow$ determine „most similar" factorized distribution. Only consider factorized distributions: $q(\mathbf{Z})=\prod_{i=1}^M q_i(\mathbf{Z}_i)$
$\rightarrow$ no correlations between assignments of different objects. Then minimize the Kullback-Leibler divergence to obtain best factorial approximation.
Mean field $h_{u\alpha}$ is the expected cost $\mathcal{R}(c)$ that object u assigned to cluster α
Mean field upper bound: $F(\beta)\leq F_0(\beta)+\langle E\rangle_0-\langle E_0\rangle_0=KL(p_0||p)/\beta+F(\beta)=\sum_\sigma p_0(\sigma)E(\sigma)+\frac{1}{\beta}\sum_\sigma p_0(\sigma)log_{p_0}(\sigma)$

Determine mean field

1. split cost function $\mathcal{R}$ into terms that contain the object u and other terms. Term has a form like $\mathcal{R}(c)=f(u)+\mathcal{R}(c|u)$
2. take the expected value $\mathbb{E}_{\mathbf{Q}_{u\ toα}}$

Pseudocode:

1: initialize $\forall i, q_i(1),\dots,q_i(k)$ randomly s.t. $\sum_\nu q_i(\nu)=1$;
2: **repeat**
3: draw $u\in\{1,\dots,n\}$ randomly; // site selection
4: calculate $h_{u,\alpha}=\mathbb{E}_{\mathbf{Q}_{u\rightarrow\alpha}}[\mathcal{R}(c)]$ for given $q_u(1),\dots,q_u(k)$;
5: estimate $\forall\alpha,\ q_u(\alpha)=\frac{\exp(-\beta h_{u,\alpha})}{\sum_{\nu\leq k}\exp(-\beta h_{u,\nu})}$ given $h_{u,1},\dots,h_{u,k}$;
6: $t\leftarrow t+1$;
7: **until** $\forall\alpha$ probabilities $q_u(\alpha)$ and meanfields $h_{u,\alpha}$ are converged
8: stop

Approximation set coding

Find robust, informative pattern (hypothesis $c\in\mathcal{C}$ hypothesis space) in data

Approximation sets

data space $\mathcal{X}$, cost function $\mathcal{R}(c)$, hypothesis $c\in\mathcal{C}$
solution $c^\perp(X)=\text{argmin}_{c\in\mathcal{C}}\mathcal{R}(c)$ is random since X is random.
$\rightarrow$ if second set X' is used $\rightarrow c^\perp(X)\neq c^\perp(X')$ $\rightarrow$ do not commit to a single answer, use set where each hypothesis c has weight.

Weights

weights $w$rank different hypothesis according to how well they solve problem:
$\forall c,c'\in\mathcal{C}\,\mathcal{R}(c,X)\leq\mathcal{R}(c,X')\Leftrightarrow w_\beta(c,X)\geq w_\beta(c,X')$
where approximation precision β controls resolution of solutions.
$\rightarrow$ set of hypothesis that approximately solve the problem:
$C_\beta(X)=\{c\in\mathcal{C}:\ \mathcal{R}(c,X)-\mathcal{R}(c^\perp(X),X)\leq 1/\beta\}$
i.e. lead to solutions that are $1/\beta$ close to minimum.
assume every $c\in\mathcal{C}_\beta$ equally good $\rightarrow$ posterior $P_\beta(c|X)=\text{unif}(C_\beta(X))$
$\beta\rightarrow 0$: every hypothesis good solution $\rightarrow$ not informative.
$\beta\rightarrow\infty$: each set has only one element $\rightarrow$ not robust.
$\Rightarrow$ β controls tradeoff between informativeness and robustness/how much posterior distributions agree for different values of β.

Model Selection

neg. log-likelihood usually decreases with increasing model complexity. More parameters support a better fit! Correct this tendency to select complex models with a complexity penalty
Minimum Description Length: $-logp(x|\theta_k)-logp(\theta_k)$

Bayesian Information Criterion: $-2log(\hat{p}|\hat{\theta}_k,\mathcal{M}_k)+klogn$, where $\mathcal{M}_k$ is one of the possible model classes.
Gap Statistic: $gap_n(k):=E^*_n[log(W_k)]-log(W_k)$, where $W_k=\sum_\eta\frac{1}{2n_\eta}\sum_{i,j\in\epsilon_{\eta\eta}}D_{ij}$ and $D_{ij}=||x_i-x_j||^2$ The optimal k can then be chosen as: $k^*=min[k|gap_n(k)\geq gap_n(k+1)-\sigma_{k+1}]$
MDL and BIC formally equivalent. *Gap Statistic:*

1. Train a predictor $\phi_{Z'}$ on the data $Z':=(X',\alpha(X'))$
2. Predict labels on X using $\phi_{Z'}$
3. Compare clustering solutions $(\phi_{Z'}(X_i))_{i\in[n]}$ and $\alpha(X)$ on X

Choose classifier that matches chosen clustering model. Fails if data too noisy. The gap statistic assumes compact or spherically distributed clusters.
Stability:
Solutions on two data sets from the same source should be similar.
Two Sample Scenario:
General procedure
1. Draw two data sets from the same source. 2. Cluster both data sets. 3. Compute disagreement
(In-)Stability := expected (dis-)agreement of the solutions. Critics:
The stability method does not take the complexity of a solution appropriately into account. The tradeoff between informativeness and stability of solutions should be quantitatively evaluated.

Coding Exercises

Deterministic Annealing Clustering

Deterministic Annealing Clustering is an approach to the unsupervised learning problem of clustering. At its core the algorithm is about minimizing the Lagrangian:

$$F=D-TH$$

, where D is a measure of the distortion, the "distanceöf the datapoints from their respective cluster centers, H is the Shannon Entropy of the system and T is the temperature. The algorithm is grounded in both information theory and statistical physics. The core idea is that, at an initially high temperature the entropy is maximized. Then as we gradually lower the temperature T, the distortion is minimized.
The *preferred implementation* of the algorithm is as follows:

1. Initialize the number of codevectors, the minimal temperature, the initial temperature, the initial number of clusters, the initial cluster center and it's probabiltiy.
2. For i in [1,K]
$$y_i=\frac{\sum_x xp(x)(y_i|x)}{p(y_i)}$$
where
$$p(y_i|x)=\frac{p(y_i)e^{(x-y_i)^2/T}}{\sum_{j=1}^K p(y_i)e^{(x-y_j)^2/T}}$$
and
$$p(y_i)=\sum_x p(x)p(y_i|x)$$
3. If not converged, repeat (2)
4. If $T<T_{min}$, stop
5. Cool down: T = α T, $\alpha<1$
6. If $K<K_{max}$ check if any of the clusters has reached critical temperature. If so, add a new codevector $yK+1=y_j+\delta, p(y_{K+1}=p(y_j)/2, p(y_j)=p(y_j)/2$, where j is the cluster that has reached critical temperature. Increase K by one.
7. Repeat from (3)

The *critical temperature* is the temperature for which a given cluster splits into two. Assuming Euclidean distance as our measure of distortion, it can be explcitly computed as

$$T_{c,y_i}=2\lambda_{max}$$

, where λ_{max} is the largest eigenvalue of $C_{x|y_i}$, which in turn is the Covariance matrix of the posterior distribution of $p(x|y)$

Histogram Clustering

MAP updates: $\hat{P}(y|c)=\sum_{x\in\hat{c}}\frac{n(x)}{\sum_{x'\in c}n(x')}\hat{P}(y|x)$ (1)
$\hat{c}(x)=argmin_a\{-\sum_{y\in Y}\hat{P}(y|x)log\hat{P}(y|a)\}$
For DA clustering (1) becomes: $\hat{P}(y|c)=\sum_{x\in X}\hat{P}(x|c)\hat{P}(y|x)$ with $\hat{P}(c)=\sum_{x\in X}\hat{P}(c|x)\hat{P}(x)$
$\hat{P}(x|c)=\frac{\hat{P}(c|x)\hat{P}(x)}{\sum_{x\in X}\hat{P}(c|x)\hat{P}(x)}$
and, with
$\hat{P}(c|x)=\frac{exp(-D_{KL}[\hat{P}(.|x)||\hat{P}(.|a)]/T)}{\sum_{b=1}^K exp(-D_{KL}[\hat{P}(.|x)||\hat{P}(.|a)]/T)}$
replacing equation (6)
Limitations histogram clustering:

Lack of Topology: Histograms represent categorical information; permutations of the bin index will not change the KL-divergence $D^{\text{KL}}(\hat{p}(\cdot|i)\|q(\cdot|c(i)))$.
Most feature spaces are equipped with a natural topology, e.g., color circle, frequency ordering, edge directions, ..., feature similarity in general!
Heuristic histogram smoothing is advisable in many applications, e.g., distribute a small percentage of a count to neighboring bins.
Systematic approach: **Model feature distribution with a parametric mixture** $\Rightarrow$ parametric distributional clustering.

DA for pairwise clustering

Given a dataset of size N and an $N\times K$ dissimilarity matrix $\mathbf{D}=(\mathcal{D}_{ik})$, the problem of pairwise clustering tries to cluster the data into K clusters, represented by an $N\times K$ assignment matrix $\mathbf{M}=(M_{i\nu})\in\{0,1\}^{N\times K}$ that minimizes the cost function

$$\mathcal{H}^{pc}(\mathbf{M})=\frac{1}{2}\sum_{i,k=1}^N\frac{\mathcal{D}_{ik}}{N}\left(\sum_{\nu=1}^K\frac{M_{i\nu}M_{k\nu}}{p_\nu}-1\right)$$

where $p_\nu=\frac{1}{N}\sum_{i=1}^N M_{i\nu}$ is the weight of cluster ν. Algorithm II approximates the Gibbs distribution given by $\mathcal{H}^{pc}$ with the Gibbs distribution given by the factorial cost function $\mathcal{H}^0(\mathbf{M},\mathcal{E})=\sum_{i=1}^N\sum_{\nu=1}^K M_{i\nu}\mathcal{E}_{i\nu}$, parameterized by external mean fields $\mathcal{E}=(\mathcal{E}_{i\nu})$. Minimizing the Kullback-Leibler divergence $D^{KL}(\mathbf{P}^{Gb}(\mathcal{H}^0)\|\mathbf{P}^{Gb}(\mathcal{H}^{pc}))$, where $\mathbf{P}^{Gb}(\mathcal{H})$ is the Gibbs distribution for the cost function $\mathcal{H}(\mathbf{M})$ gives the optimal $\mathcal{E}$ as

$$\mathcal{E}^*_{i\nu}=\left\langle\frac{1}{1+\sum_{j\neq i}M_{j\nu}}\left[\frac{1}{2}\mathcal{D}_{ii}+\sum_{k\neq i}M_{k\nu}\left(\mathcal{D}_{ik}-\frac{1}{2}\frac{\sum_{j\neq i}M_{j\nu}\mathcal{D}_{jk}}{\sum_{j\neq i}M_{j\nu}}\right)\right]\right\rangle,$$

where $\langle\cdot\rangle$ corresponds to taking expectations with respect to $\mathbf{P}^{Gb}(\mathcal{H}^0(\mathbf{M}))$. In the limit of large N, this optimum can be approximated as

$$\mathcal{E}^*_{i\nu}=\frac{1}{1+\sum_{j\neq i}\langle M_{j\nu}\rangle}\left[\frac{1}{2}\mathcal{D}_{ii}+\sum_{k\neq i}\langle M_{k\nu}\rangle\left(\mathcal{D}_{ik}-\frac{1}{2}\frac{\sum_{j\neq i}\langle M_{j\nu}\rangle\mathcal{D}_{jk}}{\sum_{j\neq i}\langle M_{j\nu}\rangle}\right)\right].$$

The expectations $\langle M_{j\alpha}\rangle$ under the Gibbs distribution $\mathbf{P}^{Gb}(\mathcal{H}^0(\mathbf{M}))$ with temperature T are then given by

$$\langle M_{j\alpha}\rangle=\frac{\exp(-\mathcal{E}^*_{i\alpha}/T)}{\sum_{\nu=1}^K\exp(-\mathcal{E}^*_{i\nu}/T)}$$

Algorithm II then uses deterministic annealing to approximate the Gibbs distribution for varying temperatures, by starting at high temperature, iteratively applying the two above equations for $\langle M_{j\alpha}\rangle$ and $\mathcal{E}^*_{i\nu}$ in an EM scheme for fixed T, and decreasing T exponentially until the final temperature is reached.
Algorithm III further constrains the variational distribution to be in the form of a Gibbs distribution for the k-means cost function

$$\mathcal{H}^{cc}(\mathbf{M})=\sum_{i=1}^N\sum_{\nu=1}^K M_{i\nu}\|\mathbf{x}_i-\mathbf{y}_\nu\|^2,$$

where $(\mathbf{x}_i)_{i=1}^N$ are embeddings of the data to be determined, and $(\mathbf{y}_\nu)_{\nu=1}^K$ are centroids calculated from $(\mathbf{x}_i)_{i=1}^N$; in this case the mean fields are constrained to take the form $\mathcal{E}_{i\nu}=\|\mathbf{x}_i-\mathbf{y}_\nu\|^2$. Minimizing the KL-divergence $D^{KL}(\mathbf{P}^{Gb}(\mathcal{H}^{cc})\|\mathbf{P}^{Gb}(\mathcal{H}^{pc}))$ with respect to $(\mathbf{x}_i)_{i=1}^N$ and $(\mathbf{y}_\nu)_{\nu=1}^K$ now gives

$$\mathbf{K}_i\mathbf{x}_i\approx\frac{1}{2}\sum_{\nu=1}^K\langle M_{i\nu}\rangle\left(\|\mathbf{y}_\nu\|^2-\mathcal{E}^*_{i\nu}\right)\left(\mathbf{y}_\nu-\langle\mathbf{y}\rangle_i\right)$$

where $\langle\mathbf{y}\rangle_i=\sum_{\nu=1}^K\langle M_{i\nu}\rangle\mathbf{y}_\nu$ is the mean of $\mathbf{y}$ under the conditional distribution $p(\nu|i)=\langle M_{i\nu}\rangle$ and $\mathbf{K}_i=\left\langle\mathbf{y}\mathbf{y}^\top\right\rangle_i-\langle\mathbf{y}\rangle_i\langle\mathbf{y}\rangle_i^\top$ the corresponding covariance matrix, and $\mathcal{E}^*_{i\nu}$ is as calculated in Algorithm II. Algorithm III again uses deterministic annealing, calculating $\langle M_{i\nu}\rangle^{(t+1)}$ in the E-step from the mean fields $\mathcal{E}^{(t)}_{i\nu}=\|\mathbf{x}^{(t)}_i-\mathbf{y}^{(t)}_\nu\|^2$, and in the M-step updating $\mathbf{x}^{(t+1)}_i$ and $\mathbf{y}^{(t+1)}_\nu$ iteratively from fixed $\langle M_{i\nu}\rangle^{(t+1)}$ until the variables converge.

MFA for Image Denoising in Ising Model

In the image denoising setting, we are looking for an image σ to minimize:

$$E(\sigma)=-\lambda\sum_{i=1}^N h_i\sigma-\sum_{i,j=1}^N J_{i,j}\sigma_i\sigma_j$$

where h_i is the value of pixel i in the noisy observation.
We thus seek to minimize $KL(p_0||p)+F(\beta)$, which is equivalent to minimizing the Gibbs free energy $G(p)$, where $F(\beta)$ is the free energy, p_0 the proposal distribution and p the true distribution.
It can be shown that:

$$G(p)=-\lambda\sum_{i=1}^N m_ih_i-\sum_{i=1}^{N-1}\sum_{j=i+1}^N m_iJ_{ij}m_j$$

$$+1/\beta\sum_{i=1}^N(\frac{1+m_i}{2}log\frac{1+m_i}{2}+\frac{1-m_i}{2}log\frac{1-m_i}{2})$$

where m_i is the mean field for pixel i. Deriving the stationary equations, we find that $G(p)$ is minimized by:

$$m_k=tanh(\beta(\sum_{i\neq k}J_{ki}m_i+\lambda h_k)), k\in 1,N$$

Which we can solve by iteratively updating the m_k.

Model Validation

//Some of this stuff is in lecture notes so we might want to cut it. In order to compare the stability of cost functions $R(c,\mathbf{X})\in\mathcal{R}$ under noise in the data $\mathbf{X}\sim\mathbb{P}(\mathbf{X})$, approximation set coding considers a communication scenario including a sender, a problem generator and a receiver. Each entity is initially given access to a sample $\mathbf{X}^{(1)}\sim\mathbb{P}(\mathbf{X})$ and an optimal solution $c^\perp(\mathbf{X}^{(1)})\in\arg\max_{c\in\mathcal{C}}R(c,\mathbf{X}^{(1)})$. The sender then chooses a transformation $\tau_s\in\mathbb{T}$, $|\mathbb{T}|=2^{n\rho}$ which acts on the solution space (by permuting elements of the dataset in the case of clustering) and sends it to the problem generator, which generates another sample $\mathbf{X}^{(2)}\sim\mathbb{P}(\mathbf{X})$ and sends $\tau_s\circ\mathbf{X}^{(2)}$ to the receiver. The receiver then estimates τ_s by maximizing the posterior agreement between $\mathbb{P}(c|\beta,\mathbf{X}^{(1)})\propto\exp(-\beta R(c,\mathbf{X}^{(1)}))$ and $\mathbb{P}(c|\beta,\tau\circ\mathbf{X}^{(2)})\propto\exp(-\beta R(c,\tau\circ\mathbf{X}^{(2)}))$, returning $\hat{\tau}\in\arg\max_{\tau\sum\mathbb{T}}\sum_{c\in\mathcal{C}}\exp(-\beta(R(c,\mathbf{X}^{(1)}+R(c,\tau\circ\mathbf{X}^{(2)}))$. Here β represents the degree to which the data is trusted; the larger the noise in the data, the lower β has to be chosen to prevent decoding errors. The decoding error $\mathbb{P}(\hat{\tau}\neq\tau_s|\tau_s)$ can be shown to vanish asymptotically if the rate of transmission ρ falls below the mutual information

$$\mathcal{I}_\beta(\tau_s,\hat{\tau})=\frac{1}{n}\log\frac{|\{\tau_s\}|\mathcal{Z}_{12}}{\mathcal{Z}_1\mathcal{Z}_2},$$

where $\mathcal{Z}_1 = \sum_{c\in\mathcal{C}} \exp(-\beta R(c,\mathbf{X}^{(1)}))$, $\mathcal{Z}_2 = \sum_{c\in\mathcal{C}} \exp(-\beta R(c,\mathbf{X}^{(2)}))$, $\mathcal{Z}_{12} = \sum_{c\in\mathcal{C}} \exp(-\beta(R(c,\mathbf{X}^{(1)}) + R(c,\mathbf{X}^{(2)})))$ and $|\{\tau_s\}|$ is the number of possible realizations of $c^{\perp}(\tau_s \circ \mathbf{X}^{(2)})$.
The approximation capacity is then defined as the maximum of $\mathcal{I}_\beta$ over all inverse temperatures β. Then to apply model selection on $\mathcal{R}$, we split a given dataset $\mathbf{X}$ randomly into two datasets $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$, compute the mutual information $\mathcal{I}_\beta$ for each $R \in \mathcal{R}$, maximize it with respect to β, and choose the cost function with the largest approximation capacity.
For k-means clustering, we have $R(c,\mathbf{X};\mathbf{Y}) = \sum_{i=1}^{n} \epsilon_{i,c(i)}$ where $\epsilon_{ik} = \epsilon_{ik}(\mathbf{X},\mathbf{Y}) = \|x_i - y_k\|^2$ and $\mathbf{Y}$ are the cluster centroids inferred by maximizing the entropy of $\mathbb{P}(c|\beta,\mathbf{X})$. For $\epsilon_{ik}^{(i)} = \epsilon_{ik}(\mathbf{X}^{(i)},\mathbf{Y})$ the mutual information is calculated as

$$\mathcal{I}_\beta = \frac{1}{n}\log|\{\tau_s\}| + \frac{1}{n}\sum_{i=1}^{n}\log\frac{\sum_{k=1}^{K} e^{-\beta(\epsilon_{ik}^{(1)}+\epsilon_{ik}^{(2)})}}{\sum_{k=1}^{K} e^{-\beta\epsilon_{ik}^{(1)}}\sum_{k=1}^{K} e^{-\beta\epsilon_{ik}^{(2)}}}$$

where $|\{\tau_s\}|$ is the number of distinct clusterings on $\mathbf{X}^{(1)}$.
Then to evaluate the approximation capacity of the k-means cost function, we use deterministic annealing to compute the optimal centroids and costs $R(c,\mathbf{X}^{(i)})$ at different temperatures $T = \beta^{-1}$, allowing for $2K$ possible clusters to enable overfitting, and choose as stopping temperature the one with the highest mutual information, balancing informativeness and robustness.
Examples of transformations: permutations in graph based clustering, translation for mean estimation, rotation for SVD, scalling for linear regression, permutation and scaling for sparse linear regression.

Exercise Sheets

Optimal β for Expected Log-Posterior Agreement

We consider the Gibbs posterior over hypotheses: $p_\beta(c|X) = \frac{1}{Z(\beta,X)} exp(-\beta R(c,X))$. For two data sets X', X', the posterior agreement is defined as: $\hat{k}_\beta(X',X'') = \sum_{c\in\mathcal{C}} p_\beta(c|X')p_\beta(c|X'')$ We want to find β^* for which the posterior agreement is maximal. Finding the right β can be seen as a trade off between stability and informativeness of the distribution. In a sense, the optimal β is supposed to give us the most robust solution.

Quantization errors are caused by assignments $x_i \to y_{c(i)}$

$$\sum_{i=1}^{n} D(x_i, y_{c(i)})$$

Code vector transmission error

The sender decides for assignment $c(i)$; but due to channel noise, the receiver decodes code vector index α with probability $\mathbf{T}_{c(i),\alpha}$.

$\mathbf{T}_{c(i),\alpha}$ can be interpreted as a conditional probability to receive α given that $c(i)$ has been sent, i.e., $\sum_{\nu=1}^{k} \mathbf{T}_{c(i),\nu} = 1$ holds.

The channel noise breaks the permutation symmetry of the cluster labels. Whether two cluster indices are confused, depends on the closeness in representation space, i.e., how likely the code for the indices are confused.

Topological costs and re-estimation equations

Topological costs

In addition to quantization errors $D(x_i,y_\nu) = \|x_i - y_\nu\|^2$ we have to consider distortions due to "erroneous assignments" of prototypes.

$$\mathcal{R}^{\text{som}}(c,\mathbf{Y},\mathbf{X}) = \sum_{1\le i\le n}\sum_{1\le\nu\le k} \mathbf{T}_{c(i),\nu}\, D(x_i,y_\nu)$$

Centroid equation for topological clustering

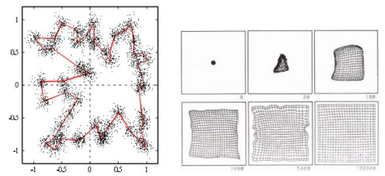
They are derived by the condition $\frac{\partial}{\partial y_\alpha}\mathcal{R}^{\text{som}}(c,\mathbf{Y},\mathbf{X}) = 0$

$$0 = \frac{1}{n}\sum_{i=1}^{n}\sum_{\gamma=1}^{k}\mathbf{P}_{i\gamma}\mathbf{T}_{\gamma\alpha}\frac{\partial}{\partial y_\alpha}D(x_i,y_\alpha), \qquad p_\alpha = \frac{1}{n}\sum_{i=1}^{n}\mathbf{P}_{i\alpha}$$

Topological clustering and Kohonen maps

Fuzzy assignment variables ($\mathcal{R}_{i\alpha} := \sum_{1\le\nu\le k}\mathbf{T}_{\alpha,\nu}\, D(x_i,y_\nu)$)

$$\mathbf{P}_{i\alpha} = \frac{\exp(-\beta\mathcal{R}_{i\alpha})}{\sum_{\mu=1}^{k}\exp(-\beta\mathcal{R}_{i\mu})}.$$



Complexity Constraint Clustering

Complexity constraint costs

The number of clusters is not set a priori, but it is determined by jointly minimizing distortion and complexity costs.

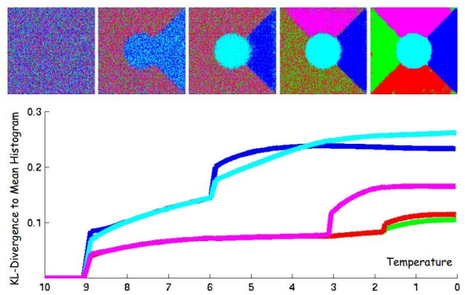
Complexity limited costs

(cluster size $p_\alpha = \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}_{\{c(i)=\alpha\}}$)

$$\begin{aligned}\mathcal{R}^{\text{cc}}(c,\mathbf{Y},\mathbf{X}) &= \sum_{1\le i\le n}\left(D(x_i,y_{c(i)}) + \lambda C(p_{c(i)})\right),\\ &= \frac{1}{n}\sum_{1\le i\le n} D(x_i,y_{c(i)}) + \lambda\sum_{1\le\nu\le k} p_\nu C(p_\nu),\end{aligned}$$

$C^{\text{ent}}(p_\nu) = -\log p_\nu$ entropy constraint
$C^{\text{lb}}(p_\nu) = (p_\nu)^{-s}, \quad s \ge 1$ load balancing ($s = 1$ k-means)

Phase Transitions in Segmentation



EM Update Scheme for PDC

E-step $h_{i\nu} = -\log p_\nu - \sum_j \frac{l}{n}\hat{p}(y_j|x_i)\log\left(\sum_\alpha p(\alpha\,|\,\nu)\tilde{G}_\alpha(j)\right)$
$q_{i\nu} = \mathbb{E}\left[\mathbb{I}_{\{c(i)=\nu\}}\right] \propto \exp\left(-\frac{h_{i\nu}}{T}\right)$

M-step $p_\nu = \frac{1}{n}\sum_i q_{i\nu}$
numerical solution for $p(\alpha\,|\,\nu)$

No closed formula for $p(\alpha\,|\,\nu)$, nor for μ_α is known ! Therefore, we iteratively optimize pairs $p(\alpha_1\,|\,\nu)$, $p(\alpha_2\,|\,\nu)$ until convergence such that $\sum_\rho p(\rho|\nu) = 1$. Interval bisection is used to optimize Gaussian means μ_α.

Maximum Likelihood Approach for Histogram Clustering

Prior of assignment function c:

$$p\,(c) = \prod_{i=1}^{n} p_{c(i)}$$

Data likelihood for given c:

$$p\,(\mathcal{Y}\,|\,c,\theta) = \prod_{i=1}^{n}\left[\prod_{j=1}^{m}\left(\sum_{\alpha=1}^{s} p(\alpha\,|\,c(i))\tilde{G}_\alpha(j)\right)^{\hat{p}(ij)}\right]$$

The parameter vector θ summarizes all the parameters of the mixture model, e.g., mixture coefficients, centroids and covariances of the Gaussians, binning parameters.

Cost Function for PDC

Cost function = negative log-likelihood:

$$\begin{aligned}\mathcal{R}(c,p_{\cdot|c}) &= -\sum_i\Bigg[\log p_{c(i)}\\ &\quad + \frac{l}{n}\sum_j\hat{p}(y_j|x_i)\log\left(\sum_\alpha p(\alpha\,|\,c(i))\tilde{G}_\alpha(j)\right)\Bigg]\end{aligned}$$

Interpretation as **two-part coding scheme**:
Expected codelength when encoding the cluster memberships and, based on that information, encoding the individual feature values.

Transformation of dissimilarities into similarities

Exponential scaling

$$S_{ij} = \exp(-D_{ij}/\Delta)$$

This transformation "seems" to be psychologically motivated, e.g., it is often used in psychology. From a mathematical point it maps dissimilarities $D_{ij} \ge \Delta$ into the interval $[0, e^{-1}]$.

Why is this beneficial?

Because large distances are hard to measure accurately in many applications like protein homology, face comparison & image matching, speaker independent speech analysis, ...

Alternative conversion

Linear mapping between similarities and dissimilarities:

$$S_{ij} = \max_{ij} D_{ij} - D_{ij}$$

Shifted Correlation Clustering

Count only similarities relative to a threshold value u!

The motivation for correlation clustering, namely to group together highly similar objects and to separate highly dissimilar objects, is modified by choosing continuous similarities ($S_{ij} \in [-1,+1]$) and to sum them up relative to a threshold u that support this design principle.

$$\begin{aligned}\mathcal{R}^{\text{cc}}(c;\mathcal{D}) &= -\frac{1}{2}\sum_{\nu\le k}\sum_{(i,j)\in\mathcal{E}_{\nu\nu}}(|S_{ij}+u| + S_{ij} + u)\\ &\quad + \frac{1}{2}\sum_{\nu\le k}\sum_{\substack{\mu\le k\\ \mu\ne\nu}}\sum_{(i,j)\in\mathcal{E}_{\nu\mu}}(|S_{ij}+u| - S_{ij} - u)\end{aligned}$$

The function $\frac{1}{2}(|X| \pm X) = \max\{0, \pm X\}$ selects only positive/negative weights X dependent on the sign $\pm$.

Alternative Cut Objectives

Average Cut: (originally formulated for similarities)

$$\mathcal{R}^{\text{ac}}(c;\mathcal{S}) = \sum_{\nu\le k}\left(\frac{\text{cut}(\mathcal{G}_\nu, \mathcal{V}\setminus\mathcal{G}_\nu)}{|\mathcal{G}_\nu|}\right)$$

Min-Max Cut:

$$\mathcal{R}^{\text{mmc}}(c;\mathcal{S}) = \sum_{\nu\le k}\left(\frac{\text{cut}(\mathcal{G}_\nu, \mathcal{V}\setminus\mathcal{G}_\nu)}{\text{assoc}(\mathcal{G}_\nu, \mathcal{G}_\nu)}\right)$$

Note: Min-Max Cut introduces a severe bias towards equipartitions.

Graph theoretic definitions: W is the weight matrix of graph $(\mathcal{V},\mathcal{E})$

$$\text{cut}(A,B) = \sum_{i\in A, j\in B} W_{i,j}, \quad \text{assoc}(A,\mathcal{V}) = \sum_{i\in A, j\in\mathcal{V}} W_{i,j}$$

$\text{assoc}(A,\mathcal{V})$ measures the total connection strengths from nodes in A to all nodes in the graph.

Posterior agreement kernel ...

$$k(X',X'') = \sum_{c\in\mathcal{C}} P_\theta(c|X')P_\theta(c|X'')$$

measures the similarity of X' and X'' that is induced by the posterior distribution of width θ.

Essentially, the posterior $P_\theta(c|X)$ specifies a sampling procedure how to choose hypotheses c that are highly likely given data X.
The posterior agreement kernel $k(X',X'')$ measures if a hypothesis c yields high posterior values both for given X' and X'', i.e., if c is a highly likely hypothesis under both data sets.
$k(X',X'')$ quantifies the information in X',X'' that is accounted for by selecting hypotheses $c \sim P_\theta(c|X)$. The posterior induces the similarity by its robustness to fluctuations in the data.

In the tutorial we discussed the relationship between the information bottleneck method and Parametric Distributional Clustering. The bottleneck functional is given by $L = I(X,\tilde{X}) - \beta I(\tilde{X},Y)$. Show that with the definitions

X: Decision to pick object i uniformly, i.e. $p(X=i) = \frac{1}{n}$, $i = 1\cdots n$
$\tilde{X}$: Encoding of object i by cluster ν: $p(\tilde{X}=\nu|X=i) = M_{i\nu} \in \{0,1\}$, $\nu = 1\cdots k$
Y: Distribution of observations over $b = 1\cdots m$ bins: $p(Y = p_i(b))$

the Information Bottleneck functional L is equivalent to the negative log of the PDC likelihood

$$p(X,M|\theta) = \prod_{i=1}^{n}\prod_{\nu=1}^{k}[p_\nu\prod_{b=1}^{m}\{\sum_{\alpha=1}^{l}p(\alpha|\nu)G_\alpha(p)\}^{n_{ib}}]^{M_{i\nu}}$$

Solution: To solve this problem, it is recommended to look up the basic ideas in the article. The cost function is

$$I(X;\tilde{X}) - \beta I(\tilde{X};Y) = H(\tilde{X}) - H(\tilde{X}|X) - \beta H(Y) - \beta H(Y|\tilde{X})$$

There are two key ingredients for the proof:
1. $H(\tilde{X}|X) = 0$, as it is equal to

$$\sum_{i=1}^{n} p_i\sum_{\nu=1}^{k} M_{i\nu}\log(M_{i\nu}) = 0,$$

where $\lim_{a\to 0} a\log(a) = 0$.

2. The following Markov property holds

$$p(\tilde{X}|X,Y) = p(\tilde{X}|X) = M_{i\nu} \qquad \text{or} \qquad \tilde{X} \to X \to Y$$

therefore the cost function

$$H(\tilde{X}) - \beta H(Y) - \beta H(Y|\tilde{X}) = H(\tilde{X}) - \beta H(Y) - \beta\sum_X\sum_{\tilde{X}}\sum_Y\frac{1}{n}p(Y|X)p(\tilde{X}|X)\log(p(Y|\tilde{X}))$$

Eliminating $H(Y)$ leads to

$$\sum_{\nu=1}^{k}\sum_{i=1}^{n}\frac{1}{n}M_{i\nu}\log(\frac{1}{p_\nu}) - \beta\sum_X\sum_{\tilde{X}}\sum_Y\frac{1}{n}p(Y|X)p(\tilde{X}|X)\log(p(Y|\tilde{X}))$$

where $p(Y|X)$ is the frequency given by the histogram (n_i is the number of samples and n_{ib} is the number of samples from object i that have feature b), whereas $p(Y|\tilde{X})$ is given by the gaussian mixture prediction. Gathering some terms together and scaling by a factor n leads to

$$\begin{aligned}&-\sum_{\nu=1}^{k}\sum_{i=1}^{n}\frac{1}{n}M_{i\nu}\log(p_\nu) - \beta\sum_{\nu=1}^{k}\sum_{i=1}^{n}\sum_{b=1}^{m}\frac{1}{n}\frac{n_{ib}}{n_i}M_{i\nu}\log(\sum_{\alpha=1}^{l}p(\alpha|\nu)G_\alpha(b))\\ &= -\sum_{\nu=1}^{k}\sum_{i=1}^{n}M_{i\nu}\left[\log(p_\nu) + \frac{\beta}{n_i}\sum_{b=1}^{m}n_{ib}\log\left(\sum_{\alpha=1}^{l}p(\alpha|\nu)G_\alpha(b)\right)\right]\end{aligned}$$

From the tutorial the likelihood of the observation given the parameters

$$p(X,M|\theta) = \prod_i\prod_\nu\left[p_\nu\prod_{b=1}^{m}\left(\sum_{\alpha=1}^{l}p(\alpha|\nu)G_\alpha(b)\right)\right]$$

Now taking $-\log$ leads to

$$-\sum_{\nu=1}^{k}\sum_{i=1}^{n}M_{i\nu}\left[\log(p_\nu) + \sum_{b=1}^{m}n_{ib}\log\left(\sum_{\alpha=1}^{l}p(\alpha|\nu)G_\alpha(b)\right)\right]$$

which is basically the Information Bottleneck functional, if n_i is constant for all objects and $\beta = n_i$.

Meanfield

2) Using as a proxy cost function $E_0 = -\sum_i\sigma_i h_i^{eff}$ we get the proxy partition sum that factorizes as:

$$Z_0 = \sum_{\sigma_1}\cdots\sum_{\sigma_n}e^{\beta\sigma_1 h_1^{eff}}\cdots e^{\beta\sigma_n h_n^{eff}} = \left(\sum_{\sigma_1}e^{\beta\sigma_1 h_1^{eff}}\right)\cdots\left(\sum_{\sigma_n}e^{\beta\sigma_n h_n^{eff}}\right). \tag{2}$$

Hence the proxy pdf factorizes too and reads:

$$p_0(\boldsymbol{\sigma}) = 1/Z_0\prod_i e^{\beta\sigma_i h_i^{eff}} = \prod_i\frac{e^{\beta\sigma_i h_i^{eff}}}{Z_0^i}. \tag{3}$$

and so $q_i(\sigma_i) = \frac{e^{\beta\sigma_i h_i^{eff}}}{Z_0^i}$ and $Z_0^i = e^{\beta h_i^{eff}} + e^{-\beta h_i^{eff}}$. Now we can rewrite $q_i(\sigma_i)$ using the indicator functions $\mathbb{1}\{\sigma_i = 1\} = \frac{1+\sigma_i}{2}$ and $\mathbb{1}\{\sigma_i = -1\} = \frac{1-\sigma_i}{2}$.

$$q_i(\sigma_i) = \frac{1+\sigma_i}{2}\frac{e^{\beta h_i^{eff}}}{e^{\beta h_i^{eff}} + e^{-\beta h_i^{eff}}} + \frac{1-\sigma_i}{2}\frac{e^{-\beta h_i^{eff}}}{e^{\beta h_i^{eff}} + e^{-\beta h_i^{eff}}} = \frac{1+\sigma_i\tanh(\beta h^{eff})}{2}. \tag{4}$$

This last equation proves both 2) and 3).