Maximum Entropy Clustering $O(nk^N) \rightarrow O(kN)$

Sum-product trick: $Z = (f_{11}f_{21}\dots f_{n1}) + (f_{12}f_{21}\dots f_{n1}) + \dots + (f_{1k}f_{2k}\dots f_{nk}) = (f_{11}+f_{12}+\dots+f_{1k})\dots(f_{n1}+f_{n2}+\dots+f_{nk})$

DB: $\exp(-R(c_2)/T)/Z = [\exp(-R(c_1)/T)/Z]\exp(-(R(c_2)-R(c_1))/T)$

$P(c_2)P(c_2 \rightarrow c_1) = P(c_1)P(c_1 \rightarrow c_2)$, $R(c_1) \leq R(c_2)$

Gibbs: draw i, $c(i) \sim p(c(i) \mid c(1),\dots,c(i-1),c(i+1),\dots,c(n))$

- Deterministic setting: $\nabla_\theta R = 0$
- Probabilistic setting: $\mathbb{E}[\nabla_\theta R] = 0$

- kmeans: global minimum may be fragile (wrt noise), nearest neighbor rule is the culprit, very dependent on init. config., indep. makes altern. opt. tractable
- Fuzzy assgnt. algo. doesn't operate in the state of solutions, we get a summary statistic of all highly probable solutions (under Gibbs distr.)
- DA: $T \rightarrow 0$ is k-means, assign each point to closest; $T \rightarrow \infty$ is uniform P, all centroids are equal to mean
- kmeans: hard assgmnts are problematic, calculus doesn't represent doubt/uncertainty about assgmnts
- Fuzzy: conditioning on params makes obj.s indep.
- $R^{SOM}(c,Y,X) = \sum_{i\leq n}\sum_{v\leq k} T_{c(i),v}\, D(x_i,y_v)$ - reduce data dims and display similarities among datapoints
- two sources: quantization error, codevector confusion
- $R^{cc}(c,Y,X)$: $1/n \cdot \sum_{i\leq n} D(x_i, y_{c(i)}) + \lambda \sum_{v\leq k} p_v C(p_v)$ number of clusters isn't set a priori
  - $C^{ent}(p_v)$: $-\log p_v$ (entropy constr.)
  - $C^{lb}(p_v)$: $(p_v)^{-s}$, $s \geq 1$ (load balancing)
  - $s=1$ is k-means, pay price for heavily used indices
- C^{ent}: Huffman coding, $-\lim_{p_v\rightarrow 0} p_v \log p_v = 0$, optimal soln is uniform quant., codebook ignores data, we can invest a centroid for one datapoint
- least angle: groups cosine-similar vectors, gives too much weight to large clusters

$R^{la}(M,X) = -1/n \sum_{v\leq k}\sum_{i\leq n}\sum_{j\geq i} M_{iv}M_{jv}\cos\phi_{ij}$
$= -1/2n \sum_{v\leq k}\sum_{i\leq n}\sum_{j\leq n} M_{iv}M_{jv}e_i^T e_j + 1$
$= -1/2n \sum_{v\leq k}(\sum_{i\leq n} M_{iv}e_i)^2 + 1$

can normalize the cost func by cluster sizes

EM algo ⇒ Init: θ^0 init with random values

E-step: compute $p(\bar{z} \mid X, \theta^0)$

M-step: $\theta \leftarrow \arg\max_\theta \mathbb{E}_{p(\bar{z}\mid X,\theta^0)}[\log p(X,\bar{z}\mid\theta)]$

Repeat: (If θ^0 and θ are close enough finish) otherwise set $\theta^0 \leftarrow \theta$ and go to E-step

prereq: efficient to calc. $p(\bar{z} \mid X, \theta)$ for any θ, efficient to calc. M-step for any θ_0

Markov chains: Irreducible if it is possible to get to any state from any state. Periodicity: a state i has period k if any return to state i must occur in multiples of k time steps. If $k=1$ state is aperiodic. Chain is aperiodic if every state is aperiodic. An irreducible chain needs one aperiod. state to imply all states are aperiodic.

MCMC: always accepts improvements, accepts cost deteriorations with small prob. - prob. of being in a state is Boltzmann weight for that state - DB guarantees Gibbs distr. is the stationary distr., it is sufficient but not necessary cond.
- proposal distr. reflects search strategy, a flat prior doesn't exploit local structure, topology plays no role - areas of low cost doesn't mean solutions look similar, they are only judged to be comparable

- simulated annealing: approx. global min in fixed time rather than finding precise local optimum
- MH: acceptance $A(x_t, x') = \min\{1, \frac{p(x')q(x_t\mid x')}{p(x_t)q(x'\mid x_t)}\}$
- Metropolis: MH with symmetric proposal: $q(x_t\mid x') = q(x'\mid x_t)$
- Simulated annealing: MH with non-homogeneous Markov chain $p_t(x) \propto p^{1/T_t}(x)$, $T_0 = 1$, many cooling schedules
- exponential: $T(t) = T_0\alpha^t$, linear: $T(t) = T_0 - \eta t$
- Gibbs: requires sampling from conditional posterior, reduces multidim sampling to sequence of 1D
- If the Markov chain is time-homogeneous, process is described by single, time-indep. matrix p_{ij}, π is called stationary distribution s.t. $0 \leq \pi_i \leq 1$, $\sum_{i\in S}\pi_i = 1$

$\pi_j = \sum_{i\in S}\pi_i p_{ij} \rightarrow \pi P = \pi$, P is transition matrix

$\pi = \frac{e}{\sum_i e_i}$ - e is the left eigvec of transition mat. P with eigval 1

- Stationarity: $\sum_x p(x)p(x\rightarrow y) = p(y)$ (DB implies stat.)

For $X \in \mathbb{R}^{m\times n}$, $C_m X$ removes means from columns, XC_n removes means from rows, scatter $S = XC_n(XC_n)^T = XC_nX$

$Q_n = I_n - 1/n\, U(n)$, $P^c = Q_nPQ_n$, $U(n) = 11^T$ (mat. of all 1's)

My implementation of CSE

1- Get D (assume 0 diag), 2- Symmetrize D
3- Centralize D: $D^c \leftarrow QDQ$, 4- similarities: $S^c \leftarrow -\frac{1}{2}D^c$
5- Shift sims: $\tilde{S} \leftarrow S^c - \lambda_n(S^c)I_n$, 6- $\tilde{D}_{(i,j)} \leftarrow \tilde{S}_{(i,i)} + \tilde{S}_{(j,j)} - 2\tilde{S}_{(i,j)}$
7- $\tilde{S}^c \leftarrow Q\tilde{S}Q$, 8- $\tilde{D}^c \leftarrow (-2)\tilde{S}^c$, 9- assert $\tilde{D}^c == Q\tilde{D}Q$

- Shifted D contains squared Euclidean distances in the high dimensional space

Gibbs free energy: goal is simultaneously max.ing entropy and minimizing cost, config space is X, real-valued energy function is $E(x)$

- $G(p) = \mathbb{E}_p[E] - 1/\beta \cdot H(p) = \sum_x p(x)E(x) + 1/\beta \sum_x p(x)\log p(x)$
- $p_\beta(x) = \exp[-\beta(E(x) - F(\beta))] = e^{-\beta E(x)} / [\sum_{x'} e^{-\beta E(x')}]$ Gibbs distr.
- $G(p) = 1/\beta \cdot D_{KL}(p \| p_\beta) + F(\beta)$, $G(p^* = p_\beta) = 0$, $G(p) \geq F(\beta)$
- $F(\beta) = -1/\beta \log Z(\beta)$, $Z(\beta) = \sum_x e^{-\beta E(x)}$
- $F(\beta) = -1/\beta \log[\sum_x e^{-\beta E(x)}]$ can be derived from above using $\sum_x p_\beta(x) = 1$

Locally linear embedding: do local linear approx. at each point and smoothly interpolate them (as manifold locally looks linear)
- Find k-NNs for each point, Z contains neighbors of X, define local covar. as $C = Z^TZ$, solve for $Cw = 1$, assign normalized w as weights of neighbors
- After computing all weights create sparse matrix $M = (I-W)^T(I-W)$ and get bottom d+1 eigenvector, discard the bottom (eigval 0). This gives the embed of original points
- using kNN means we take a fine-grained look when there is a lot of data, and coarse-grained view when there's little data
- LLE tends to handle non-uniform sample densities poorly because there is no fixed unit to prevent weights from drifting as various regions differ in sample densities

- Linearization trick:

$\exp(b^2/2a^2) = \int_{-\infty}^{\infty}(a/\sqrt{2\pi})\exp(-a^2x^2/2 + bx)\,dx$

$\exp(-b^2/2a^2) = \int_{-i\infty}^{i\infty}(a/\sqrt{2\pi})\exp(a^2x^2/2 - bx)\,dx$

- Gaussian integral: $\int_{-\infty}^{\infty} e^{-a(x+b)^2}dx = \sqrt{\pi/a}$

- Sum of norm. distr. vars: $X \sim N(\mu_x, \Sigma_x)$, $Y \sim N(\mu_y, \Sigma_y)$; $X+Y \sim N(\mu_x+\mu_y, \Sigma_x+\Sigma_y)$, $Y = c+BX \sim N(c+B\mu_x, B\Sigma_xB^T)$
- $I(A;B) = \mathbb{E}_{A,B}[\log(P(A,B)/P(A)P(B))]$
- $1/2(|x| \pm x) = \max\{0, \pm x\}$ (shifted correlation cluster)
- $cut(A,B) = \sum_{i\in A, j\in B} W_{ij}$ - $assoc(A,V) = \sum_{i\in A, j\in V} W_{ij}$
- assoc measures the total connection strengths from nodes in A to all nodes in the graph
- Glivenko-Cantelli: $P\{\sup_{x\in\mathbb{R}}|F(x) - \hat{F}_n(x)| \xrightarrow{n\rightarrow\infty} 0\} = 1$
- four axioms of surprise $S: [0,1] \rightarrow [0,\infty)$
  - certainty: $S(1) = 0$, no surprise for certain events
  - anti-monotonicity: $p \leq q \Rightarrow S(p) \geq S(q)$
  - additivity: $S(p\cdot q) = S(p) + S(q)$
  - continuity: $S(p)$ is a cont. func. of p, no surprise jumps for infinitesimal changes in p

- $f(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left[-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right] \rightarrow N(x\mid\mu,\sigma)$
- $f(x) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}}\exp\left[-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right] \rightarrow N(x\mid\mu,\Sigma)$
- binomial: $f(k,n,p) = Pr(X=k) = \binom{n}{k}p^k(1-p)^{n-k}$
- $Var[X] = \int_x (x-\mu)^2 p(x)\,dx$
- $Var[X] = E[(X-E(X))^2] = E[X^2] - E[X]^2$
- $Var[X+Y] = Var[X] + Var[Y] + 2cov[X,Y]$
- $Cov[X,Y] = E[(X-E(X))(Y-E(Y))]$
- $Cov[aX,bY] = ab\,Cov[X,Y]$
- $Cov(X,Y) = E[XY^T] - E[X]E[Y^T]$

Jensen's inequality: X is r.v., φ is convex ⇒ $\varphi(\mathbb{E}[X]) \leq \mathbb{E}[\varphi(X)]$

Cauchy-Schwarz: $|E(X,Y)|^2 \leq E(X)^2E(Y)^2$

$H(X) = -\sum_{i=1}^{n} P(x_i)\log P(x_i)$ evaluating X,Y simult.

$H(X,Y) = H(X\mid Y) + H(Y) = H(Y\mid X) + H(X)$

$H(X,Y) \leq H(Y) + H(X)$ → equal when X,Y indep.

cross-ent: $H(p,q) = -\sum_x p(x)\log q(x)$

cond ent: $H(X\mid Y) = -\sum_{i,j} p(x_i,y_j)\log\frac{p(x_i,y_j)}{p(y_j)}$

cross-ent: $H(p,q) = E_p[-\log q] = H(p) + D_{KL}(p\|q)$

$I(X;Y) = D_{KL}(P(X,Y) \| P(X)P(Y))$ $H(p) \leq H(p,q)$

$I(X;Y) \geq 0$, $I(X;Y) = I(Y;X)$

$I(X;Y) = H(X) - H(X\mid Y) = H(Y) - H(Y\mid X)$

$I(X;Y) = H(X,Y) - H(X\mid Y) - H(Y\mid X) = H(Y) + H(X) - H(X,Y)$

cond.mut.inf.: $I(X;Y\mid Z) = H(X\mid Z) - H(X\mid Y,Z)$

chain rule: $I(X;Y,Z) = I(X;Z) + I(X;Y\mid Z)$

$I(X;Y) = \sum_y\sum_x p(x,y)\log\frac{p(x,y)}{p(x)p(y)}$

D_{KL} (relative ent.): $D_{KL}(p(x)\|q(x)) = -\sum_x p(x)\log\frac{q(x)}{p(x)} = \sum_x p(x)\frac{p(x)}{q(x)}$

$D_{KL}(p(x)\|q(x)) \geq 0$

Clustering distr. data - Histogram clustering

we partition prob. simplex rather than Euclidean space

hc: lacks topology as KL-div is invariant to permut. of feat.s, problem in ordered feature spaces (natural topology: frequency ordering, edge directions, color circle etc. - feature similarity)
- heuristic hist. smoothing: distribute percentages to neighbors
- pdc: no longer permut. invariant, optimize through EM, no closed form soln for Gauss. weights or means, optimize μ_α through interval bisection, optimize weights pairwise: pick randomly two Gaussians shift weight from one to the other,
- rectified Gaussians: feature domain has finite support, cut off the mass that goes outside and scale it up (model assumpt.)

hc notation ⇒ i: object, j: feature, l: #observations, Z: dataset, r: observation, $q(j\mid\alpha)$: cluster feat. distr.
- pdc: feats depend only on cluster idx, not explicitly on site
- pdc update → E-step: $q_{iv} = \mathbb{E}[\mathbb{I}_{\{c(i)=v\}}] \propto \exp(-h_{iv}/T)$

$h_{iv} = -\log p_v - \sum_j \frac{1}{n}\hat{P}(y_j\mid x_i)\log(\sum_\alpha p(\alpha\mid v)\tilde{G}_\alpha(j))$ $\sum_\rho p(\rho\mid v) = 1$

M-step: $p_v = \frac{1}{n}\sum_i q_{iv}$, numerical solution for $p(\alpha\mid v)$
- IB: minimizing $I(X;C)$: generic clusters, peculiarities of x should be forgotten and what's generic of all the objects assigned to particular clusters should remain $\xrightarrow{\infty}$ assigning all objects to one cluster, entropy death, no structure, - counterforce: representations informative as possible relative to a feature space (quantization objective + distortion constr.)

$R^{IB} = I(X;C) - \lambda I(C;Y)$, where $C_{opt} = \arg\min_c R^{IB}$

$R(D) = \min_{\{p(c\mid x): \mathbb{E}_{x,c}d(x,c) \leq D\}} I(X;C)$

$F(p(c\mid x)) = I(X;C) + \beta\,\mathbb{E}_{x,c}[d(x,c)]$

Pairwise clustering (graph clustering)
- clustering algo.s always impose structure on data
- sim vs dissim: use large values only when you are pretty certain that these large values are doing the job, distances are difficult to measure accurately in most applics, very large mistakes with very large distances which are very rare might still influence your estimators
- dissim to sim: exponential scaling: $S_{ij} = \exp(-D_{ij}/\Delta)$ maps dissimilarities $D_{ij} \geq k\Delta$ into interval $[0, e^{-k}]$, beneficial as large dists hard to measure - alt. conversion: linear mapping: $S_{ij} = \max_{ij}D_{ij} - D_{ij}$
- cc: seems to put wrong emphasis in evaluation, large clusters have more weight (it is less expensive to add new nodes to big clusters than the small ones
- shifted cc: count sims only relative to a threshold, $S \in [-1, +1]$
- $R^{scc}(c;D) = -\frac{1}{2}\sum_{v\leq k}\sum_{(i,j)\in\xi_{vv}}(|S_{ij}+u| + S_{ij}+u) + \frac{1}{2}\sum_{v\leq k}\sum_{\substack{\mu\leq k\\ \mu\neq v}}\sum_{(i,j)\in\xi_{v\mu}}(|S_{ij}+u| - S_{ij} - u)$
- graph cut: strong balance for very unbalanced (very small/very large) clusters due to lack of normalization
- $R^{gp}(c;D) = \sum_{v\leq k}\sum_{(i,j)\in\xi_{vv}} D_{ij} = \sum_{(i,j)\in\xi} D_{ij} - \sum_{v,\mu:v\neq\mu}\sum_{(i,j)\in\xi_{v\mu}} D_{ij}$

$= const - \sum_{v\leq k} cut(\mathcal{G}_v(D), V\setminus\mathcal{G}_v(D)) = const_2 + \sum_{v\leq k} cut(\mathcal{G}_v(S), V\setminus\mathcal{G}_v(S))$

- $R = \sum_{v\leq k} cut(\mathcal{G}_v, V\setminus\mathcal{G}_v)/norm$ → normcut $= assoc(\mathcal{G}_v, V)$ → avg $= |\mathcal{G}_v|$ → minmaxcut: $assoc(\mathcal{G}_v, \mathcal{G}_v)$
- min-max cut has severe bias towards equipartitions
- PC shift-invariance ⇒ $\tilde{D}_{ij} = D_{ij} + D_0(1-\delta_{ij}) \rightarrow R^{pc}(c;\tilde{D}) = R^{pc}(c;D) + D_0 n$
- symmetry-invariance: $R(c; \frac{1}{2}(D_{ij}+D_{ji})) = R(c;D)$ $\frac{1}{\text{off-diagonal}}$
- $R^{pc} = R^{km}$ if $D_{ij} = \|x_i - x_j\|^2$
- CSE: $\tilde{D} = D + D_0(e_ne_n^T - I_n)$ corresponds to squared Euclidean distance where $D_0 = -2\lambda_n(S^c)$ is the minimal constant

Model selection validation for clustering

- Structure can be invalid in two ways: wrong model order and/or inappropriate model type
- validation methods can be external (= comp. with ground-truth) or internal
- we want to use converge guarantees for sums (e.g. law of large numbers) so we convert the big product term to a big sum over exponent
- Every method for valid. introduces a bias
- You may be grossly misled by naively counting the number of params (cf. VC-dim of sine)
- General approach: measure quality for diff. k with some discount (i.e. complexity penalty) - with more k we gain more bits, params can fit better

- MDL: minimize$[-\log p(X|\theta_k) - \log p(\theta_k)]$ $\hat{\theta}_k$: MLE of θ
  $= [-\log p(X|\hat{\theta}_k) + 1/2\, k' \log n$ k': #params of $\hat{\theta}_k$
- Bayes factor: $p(X|M_k)/p(X|M_l)$ where $p(X|M_k) = \int p(X|\theta_k, M_k)\, p(\theta_k|M_k)\, d\theta_k$, use Lapl. approx.
  $\log p(X|M_k) = \log p(X|\hat{\theta}_k, M_k) - k'/2 \cdot \log n + O(1)$
- MDL and BIC are formally equivalent, consistent
- In modern statistics more d.o.f. we have as we have more data - BIC exploits finite dim. param space and growing data (samples n) - classical limit of statistics, we understand it well, but not useful (parametric statistics)
- They are well-motivated, they do likelihood based optimization instead of posterior based, which is intrinsically hard

Gap statistic: find "knee" in costs, it uses max. discrepancy between actual data cost and that of unstructured ref. data (i.e. which cannot be clustered, so-called "null-model"), if the data are too noisy, the method may fail

- $\text{gap}_n(k) = \mathbb{E}^*_n[\log(W_k)] - \log(W_k)$, $D_{ij} = \|x_i - x_j\|^2$,
  $W_k = \sum_{v \le k} \frac{1}{2n_v} \sum_{(i,j)\in E_v} D_{ij}$ – $\mathbb{E}^*_n$ is the expect. value w.r.t. sample of size n from null-model
- $\hat{k} = \min\{k \mid \text{gap}_n(k) \ge \text{gap}_n(k+1) - \sigma_{k+1}$
- in practice approx. $\mathbb{E}^*_n[\log(W_k)]$ by bootstrap.
- gap stat. works satisfactorily for spherical and well-separated clusters (it assumes compact clusters), a structural bias, it is a fast heuristic for k-means-like criteria

Stability based valid: Idea → solutions on two datasets from same source should be similar, signal in both should be same but fluctuations should be different, model mismatch in soln transfer produces unstable clustering solutions

- type of classifier has a large influence, has to be selected with care (e.g. path-based clust. data with a nearest centroid classifier)
- two disjoint sets of equal size: overlap could determine group structure (stat. dependence), algo. should be able to find similar structure in both, cannot find if one is too small and such structure is no longer visible to algo.
- good performance in experimental datasets, has systematic bias towards too simplistic solutions - it is principled but with a deficit: it only tells you how reliable under repeated experimentation the algo. would react, but it doesn't take into account the informativeness of solution (disregards the tradeoff) - A tolerable decrease in stability of inferred patterns might be compensated by a substantial increase of their information content

Stability valid. procedure:

1- Transfer via prediction: construct classifier ϕ trained on (X,Y) where Y is the clustering solution to X as $Y := A_k(X)$. We consider labeling $\phi(X') := (\phi(X'_i))_{i\le n}$ as the extension of the clustering solution Y on dataset X to data set X'. These predicted labels can be compared to those generated on the clustering algo $A_k(X')$

2- compare solutions: a very natural distance measure for comparing labelings $\phi(X')$ and Y' is Hamming dist. (0-1 loss), this can be treated as the empirical misclassification risk of ϕ with regard to the training set

3- Permute solutions: to overcome non-uniqueness we optimally permute solutions to maximize agreement. Hungarian method does this in $O(n+k^3)$ by minimum bipartite matching

- Dissim.: $d_{S_k}(\phi(X'),Y') := \min_{\pi\in S_k} \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{\pi(\phi(X'_i)) \ne Y'_i\}$
- Stability: $S(A_k) := \mathbb{E}_{X,X'}\, d_{S_k}(\phi(X'),Y')$

smaller the values of the index $S(A_k) \in [0,1]$, the more stable are the solutions

4- Calculate relative stability w.r.t. stability achieved by random label guessing:
$\bar{S}(A_k) := S(A_k)/S(R_k)$, R_k assigns labels to objs with prob. $1/k$

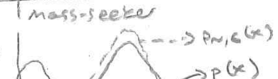
- D_{KL} penalizes approximations that underestimate likely events
- entropy of X ≈ uncertainty of X ≈ amount of info. of X ≈ minimum expected number of questions to guess X
- Forward KL, difference between $p(x)$ and $q(x)$ is weighted by $p(x)$. During the optimization process then, whenever $p(x)=0$, $q(x)$ would be ignored. The difference between $p(x)$ and $q(x)$ will be minimized if $p(x)>0$. It is known as zero avoiding, as it is avoiding $q(x)=0$ whenever $p(x)>0$

→ $p_{N,6}$ as mass-seeker: $\underset{p_{N,6}}{\text{argmin}}\, D_{KL}(p\|p_{N,6})$

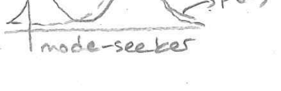
- Reverse KL, as we switch the two distr. position in the equation, now $q(x)$ is the weight. Here, it is better to fit some portion of $p(x)$ as long as that approx. is good. Consequently reverse KL will try to avoid spreading the approximate. As those properties suggest, this form of KL-div is known as zero-forcing as it forces $q(x)$ to be 0 on some areas, even if $p(x)>0$

→ $p_{N,6}$ as mode-seeker: $\underset{p_{N,6}}{\text{argmin}}\, D_{KL}(p_{N,6}\|p)$

$D_{KL}(p\|p_{N,6})$:



$D_{KL}(p_{N,6}\|p)$:



Model valid by info. theory

Typical sets - Asymptotic equipartition property (AEP):

- $-\frac{1}{n}\log p(X_1,\dots,X_n) \xrightarrow{\text{in prob}} H(X)$, for i.i.d. $X_1,\dots,X_n \sim p(X)$

A typical set $A_\epsilon^{(n)}$ wrt $p(x)$ is a set of sequences $(x_1,\dots,x_n) \in \mathcal{X}^n$ with property
$2^{-n(H(X)+\epsilon)} \le p(x_1,\dots,x_n) \le 2^{-n(H(X)-\epsilon)}$

- If $(x_1,\dots,x_n) \in A_\epsilon^n$ then $H(X)-\epsilon \le -\frac{1}{n}\log p(x_1\dots x_n) \le H(X)+\epsilon$
- $P\{A_\epsilon^{(n)}\} > 1-\epsilon$ for n sufficiently large
- $|A_\epsilon^{(n)}| \le 2^{n(H(X)+\epsilon)}$, where $|A_\epsilon^{(n)}|$ is the cardinality of the typical set
- $|A_\epsilon^{(n)}| \ge (1-\epsilon)\, 2^{n(H(X)-\epsilon)}$ for n sufficiently large

The typical set $A_\epsilon^{(n)}$ has prob. almost 1, all elements of the typical set are nearly equiprobable, and the number of typical solutions is nearly 2^{nH}

Approx. weights $w: \mathcal{C}\times\mathcal{X}\times\mathbb{R}_+ \to [0,1]$, s.t. $(c,X,\beta) \mapsto w_\beta(c,X)$ weights are non-neg, maximal weight allocated to global minimizer $c^\perp$ and it is normalized to one: $w_\beta(c^\perp,X)=1$. Solutions with large approx. weights $w_\beta(c,X) \ge 1-\epsilon$, $\epsilon \ll 1$ can be accepted as substitutes of the global minimizers. Posterior becomes
$p(c|X) = \frac{w_\beta(c,X)}{\sum_{c'} w_\beta(c',X)}$ inverse order constraints: $R(c,X) \le R(\tilde{c},X) \Leftrightarrow w_\beta(c,X) \ge w_\beta(\tilde{c},X)$

Example (unnormalized) weights:

- Boltzmann: $w_\beta(c,X) = \exp(-\beta R(c,X))$
- Fermi: $w_{\beta,\gamma}(c,X) = (1+\exp(-\beta(R(c,X)-\gamma)))^{-1}$
- Approx: $w_\gamma(c,X) = \begin{cases} 1 & \text{if } R(c,X) \le R(c^\perp,X)+\gamma \\ 0 & \text{otherwise} \end{cases}$

Normalized Boltzmann weights:
$w_\beta(c,X) = \exp(-\beta\Delta R(c,X))$, $\Delta R(c,X) = R(c,X) - R(c^\perp,X)$

- $\beta=0$ → all weights $w_\beta(c,X)=1$ ind. of costs. $Z_q = |\mathcal{C}(X)|$ indicates the size of the hypothesis space
- high β → all weights are small compared to $w_\beta(c^\perp,X)$ Z_q essentially counts the number of globally opt. sols
- Intermediate β: Z_q is the effective number of patterns that approx. fit the dataset X, where β defines the precision of approx. - Noise in measurements X reduces this resolution and thus coarsens hypo. class
- weight sum: measures total weight of hypotheses with low costs, aka partition function when we use Boltzmann weights

Equivariance transformations: idea is shifting posterior
Assume transformation $\mathbb{T}$ s.t. → posterior invariant
$\forall \tau',\tau''\in\mathbb{T}$, $\|p(c|\tau'\circ X) - p(c|\tau''\circ X)\|_1 > 0$
implies $|\mathbb{T}| \le |\mathcal{C}|$ for discrete hypo. spaces
$\sum_{\tau\in\mathbb{T}} p(c|\tau\circ X) \in \left[\frac{|\mathbb{T}|}{|\mathcal{C}|}(1-\rho), \frac{|\mathbb{T}|}{|\mathcal{C}|}\right]$ ρ measures inhomogeneity of transformation

$\tau\circ c(X) = c(\tau\circ X)$ → equivariance of algorithm
examples: permutation for graph clustering, rotation for SVD, translation for mean estimation, scaling for linear regression, permutation and scaling for sparse linear regression

Communication scenario

Sender and receiver both receive an instance X' from problem generator, calculate $p(c|X')$ and agree on a set of m randomly drawn transform. $T = \{\tau_1,\dots,\tau_m\}$ with $p(\tau) = |\mathbb{T}|^{-1}$. Here posteriors $p_\beta(c|\tau_j\circ X')$ play the role of codewords in Shannon's random coding theory.

Sender selects a transformation $\tau_s\in T$ as message and sends it to the problem generator. PG generates new instance $X''\sim p(X)$ and applies τ_s, which yields $\tilde{X} = \tau_s\circ X''$. PG sends $\tilde{X}$ to receiver without revealing either τ_s or X''. So receiver lacks both the knowledge of τ_s and suffers from the stochastic variability of X. Then receiver calculates the expected posterior $p_\beta(c|\tilde{X})$ and decodes the message $\hat{\tau}$:

$$\hat{\tau} \in \underset{\tau\in\mathbb{T}}{\text{argmax}}\, \mathbb{E}_{c\sim p_\beta(c|\tau\circ X')}\, p_\beta(c|\tilde{X})$$

We introduce kernel function to be max'ed in decoding:

$$k_{\tau_j\tau_s}(X',X'') = \mathbb{E}_{c\sim p_\beta(c|\tau_j\circ X')}\, p_\beta(c|\tilde{X})$$
$$= \sum_{c\in\mathcal{C}} p_\beta(c|\tau_j\circ X')\, p_\beta(c|\tau_s\circ X'') \in [0,1]$$

Posterior agreement kernel for $\tau_j=\tau_s$:
$k(X',X'') = \sum_{c\in\mathcal{C}} p_\beta(c|X')\, p_\beta(c|X'')$ measures the similarity of X' and X'' that is induced by the posterior distribution of width β

- essentially, the posterior specifies a sampling procedure how to choose hypotheses c that are highly likely given data X

$$Z_q = Z(X^{(q)}) = \sum_{c\in\mathcal{C}(X^{(q)})} \exp(-\beta\Delta R(c,X^{(q)})),\quad q=1,2$$
$$Z_{12} = Z(X^{(1)},X^{(2)}) = \sum_{c\in\mathcal{C}(X^{(2)})} \exp(-\beta(\Delta R(c,X^{(1)}) + \Delta R(c,X^{(2)})))$$
$$\hat{\tau} \in \underset{\tau\in\mathbb{T}}{\text{argmax}} \sum_{c\in\mathcal{C}(X^{(1)})} \exp(-\beta(R(c,\tau\circ X^{(1)}) + R(c,\underset{\tau_s\circ X^{(2)}}{\tilde{X}})))$$

vanishing error rate: $\rho \le I_\beta(\tau_s,\hat{\tau}) = \frac{1}{n}\log\left(\frac{|\{\tau_s\}|\, Z_{12}}{Z_1 Z_2}\right)$

approx. capacity: $CAP(\tau_s,\hat{\tau}) = \max_\beta I_\beta(\tau_s,\hat{\tau})$

WASC application: randomly split X into $X^{(1)}$ and $X^{(2)}$, for each candidate cost function $R(c,X) \in \mathcal{R}$ compute mutual info and max. it wrt β then select R that achieves highest capacity at the best resolution β^*

$$I_\beta(\tau_s,\hat{\tau}) = \frac{1}{n}\left(\log\frac{|\{\tau_s\}|}{Z_1} + \log\frac{|\mathcal{C}^{(2)}|}{Z_2} - \log\frac{|\mathcal{C}^{(2)}|}{Z_{12}}\right)$$

$|\{\tau_s\}|$: card. of set of possible transformations

$$P(\hat{\tau}\ne\tau_s|\tau_s) = P\left(\max_{j\ne s} k_{\tau_j\tau_s}(X',X'') \ge k(X',X'') \mid \tau_s\right)$$
(union) $\le \sum_{j\ne s} P(k_{\tau_j\tau_s} \dots)$
$= \sum_{j\ne s} \mathbb{E}_{X',X''}\left[P(k_{\tau_j\tau_s} \ge k \mid \tau_s, X', X'')\right]$
(Markov) $\le \sum_{j\ne s} \mathbb{E}_{X',X''} \frac{\mathbb{E}_{\tau_j} k_{\tau_j\tau_s}(X',X'')}{k(X',X'')}$