PA Meta-Summary ①

HMMs

1. Matching: $p(\langle o_1, \dots, o_n \rangle | \lambda)$ FWD
2. Most likely sequence: $p(\langle o_1, \dots, o_n \rangle, \langle s_1, \dots, s_n \rangle)$ VITERBI
3. Training: $\lambda = (A, B, \pi)$ Baum-Welch-Formulas FWD/BACKWD

FORWARD

$$\alpha_t(i) = P(o_1, \dots, o_t, q_t = s_i | \lambda)$$

Init: $\alpha_1(i) = \pi_i \, b_i(o_1)$

Step: $\alpha_{t+1}(j) = \left(\sum_{i=1}^{N} \alpha_t(i) \cdot a_{ij} \right) \cdot b_j(o_{t+1})$

Term: $P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$

Backward

$$\beta_t(i) = P(o_{t+1} \dots o_T | q_t = s_i, \lambda)$$

Init: $\beta_T(i) = 1$

Step: $\beta_t(i) = \sum_{j=1}^{N} a_{ij} \cdot b_j(o_{t+1}) \cdot \beta_{t+1}(j)$

Baum-Welch-Formulas

Expectation

$$\xi_t(s_i, s_t) = \text{'}P(\text{"}s_i \to s_t \text{ at } t\text{"})\text{'}$$

$$\frac{\alpha_t(i) \cdot a_{ij} \cdot b_j(o_{t+1}) \cdot \beta_{t+1}(j)}{\sum_{k=1}^{M} \sum_{l=1}^{M} \alpha_t(k) \cdot a_{kl} \cdot b_l(o_{t+1}) \cdot \beta_{t+1}(l)}$$

$$\gamma_t(s_i) = \sum_{s_j=1}^{M} \xi_t(s_i, s_j)$$

'P("in state i at t")'

Maximization

$$\bar{\pi}_i = \gamma_1(s_i)$$

$$\bar{a}_{ij} = \frac{E\#(s_i, s_j)}{E\#(s_i)} = \frac{\sum_{t=1}^{T} \xi_t(s_i, s_j)}{\sum_{t=1}^{T} \gamma_t(s_i)}$$

$$b_{s_j}(k) = \frac{E\#(s_j) \text{ s.t. } o_t = k}{E\#(s_j)}$$

MRF

$$[f_{ij}] = \underset{f_{ij}}{\operatorname{argmax}} \; P([g_{ij}] \,|\, [f_{ij}]) \cdot P([f_{ij}])$$

Hammersley-Clifford-Theorem: MRF ⟺ GRF

$$p(\vec{x}) = \frac{1}{Z} \cdot e^{-H(\vec{x})}$$

$$H(\vec{x}) = \sum_{m \in S} V_m(\vec{x})$$

Submodularity: $E(0,0) + E(1,1) \leq E(0,1) + E(1,0)$

Pairwise Potential

$$E(f_{ij}, f' \in N(f_{ij})) = \|f_{ij} - f'\|_2^2$$

Unary Potential

$$-\log(N(g_{ij}, f_{ij}, \sigma)) = \frac{(f_{ij} - g_{ij})^2}{\sigma}$$

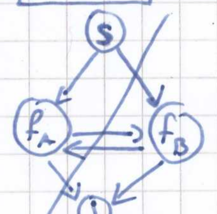
Minimization Problem

$$\sum_{i=1}^{N} \left(E(x_i) + \sum_{j=1}^{N} E(x_i, x_j) \right)$$

K-Expansion

```
while (change):
    for each label l:
        binaryGraphCut(l)
```

MinCut



$f_A = 0$

$f_B = 1$

$$E(f_A = 0) + E(f_A = 0, f_B = 1) + E(f_B = 1) + E(f_B = 1, f_A = 0)$$

PA Meta-Summary ②

Density - Estimation

Parzen-Kernel-window-Function

$$k(\vec{x}_i, x) = \begin{cases} 1, & \frac{|x_{ik} - x_k|}{h} \leq 1/2 \\ 0, & \text{otherwise} \end{cases}$$

$$p(\vec{x}) = \frac{1}{N \cdot h^d} \cdot \sum_{i=1}^{N} k_h(\vec{x}_i, \vec{x})$$

$$\mathcal{L}(h) = \prod_{j=1}^{N} p^{j}_{h,N-1}(\vec{x}_j)$$

$$\hat{h} = \underset{h}{\arg\max}\ \mathcal{L}(h)$$

Cross Validation

Mean-Shift-Algorithm

Compute Mean Vector

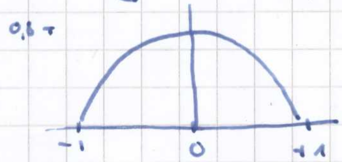
$$m(x) = \frac{\sum_{i=1}^{N} k(\|x_i - x\|^2) \cdot x_i}{\sum_{i=1}^{N} k(\|x_i - x\|^2)}$$

Update x

$$x^{(t+1)} = x^{(t)} + m(x^{(t)})$$

Epanechnikov - Kernel

$$k_E(x) = \begin{cases} c \cdot (1 - x^T x), & x^T x \leq 1 \\ 0, & \text{otherwise} \end{cases}$$



$$W(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(i')=k} d(x_i, x_{i'})$$

K-Means

Tibshirani Gap statistics

$$K^* = \underset{K}{\arg\min} \left\{ K \mid G(K) \geq G(K+1) - s'_{K+1} \right\}$$

$$G(K) = E(\log(W_K)) - \log(W(K))$$

$$s'_k = s_k \cdot \sqrt{1 + \frac{1}{B}}$$

Hierarchical Clustering

- top down / bottom up
- Single Linkage
- Complete Linkage
- Group Average

GMM: $$p(\vec{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\vec{x} \mid \vec{\mu}_k, \Sigma_k)$$

Model selection for GMMs

Gibbs sampler

For existing:

$$p_{ik} = p(z_i = k) \cdot p(\vec{x}_i \mid \mu_k, \Sigma_k) = \frac{\mu_k}{N + \alpha_0} \cdot \mathcal{N}(\vec{x}_i \mid \vec{\mu}_k, \Sigma_k)$$

For new cluster:

$$p_{i\,new} = \frac{\alpha_0}{N + \alpha_0} \cdot \mathcal{N}(\vec{x}_i \mid \mu_0, \Sigma_0)$$

Expectation

$$p_{ik} = \gamma(z_{nk}) = \frac{\pi_k \cdot \mathcal{N}(\vec{x} \mid \vec{\mu}_k, \Sigma_k)}{\underbrace{\sum_k \pi_k \cdot \mathcal{N}(\vec{x} \mid \vec{\mu}_k, \Sigma_k)}_{= p(\vec{x}_n)}}$$

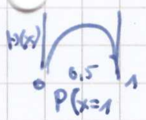
Maximization

$$\vec{\mu}_k^{new} = \frac{1}{N_k} \cdot \sum_{n=1}^{N} \gamma(z_{nk}) \cdot \vec{x}_n$$

$$\Sigma_k^{new} = \frac{1}{N_k} \cdot \sum_{n=1}^{N} \gamma(z_{nk}) \cdot (\vec{x} - \vec{\mu}_k^{new})(\vec{x} - \vec{\mu}_k^{new})^T$$

$$\pi_k^{new} = \frac{N_k}{N}$$

Decision Trees



Entropy

$$H(S_j) = -\sum_{c \in C} p(c) \cdot \log(p(c))$$

Information Gain

$$I = H(S_j) - \sum_{i \in \{L,R\}} \frac{|S_j^i|}{|S_j|} \cdot H(S_j^i)$$

Mitigate Overfitting

- Bagging
- Boosting
- Random Forests

Regression Trees

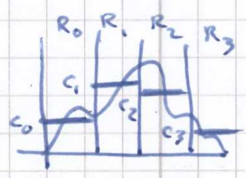
$$\hat{c}_j = \underset{c_j}{\operatorname{argmin}} \int_{R_j} \left(p(\vec{x}) - \sum_{j=1}^{K} c_j\, I(\vec{x} \in R_j)\right)^2 d\vec{x}$$

Entropy

$$H(S) = -\frac{1}{|S|} \sum_{x \in S} \int_{Y} p(y|x) \cdot \log(p(y|x))\, dy$$

Regression Split Optimization Task

$$\min_{d,s} \left\{ \min_{c_1} \sum_{\vec{x} \in R_1(d,s)} \left(\hat{p}(\vec{x}) - c_1\right)^2 + \min_{c_2} \sum_{\vec{x} \in R_2} \left(\hat{p}(\vec{x}) - c_2\right)^2 \right\}$$



$$J_{Reg} = \sum_{j=1}^{K} \int_{x \in R_j} \left(p(\vec{x}) - \hat{p}(\vec{x})\right)^2 d\vec{x} + \alpha \cdot K$$

minimize

Density Forest Entropy

$$H(S_j) = \frac{1}{2} \log\left((2\pi e)^d |\Lambda(S_j)|\right)$$

$$I(S_j, \vartheta) = \log(|\Lambda(S_j)|) - \sum_{i \in \{L,R\}} \frac{|S_j^i|}{|S_j|} \log(|\Lambda(S_j^i)|)$$

Manifold Learning

MDS Concept

$$D^2 = \operatorname{diag}(X^T X) \cdot \vec{1}^T + \vec{1} \cdot \operatorname{diag}(X^T X) - 2X^T X$$

Centered Distances

$$-\frac{1}{2} C D^2 C = \ldots = X^T X$$

Solution to eigendecomposition

$$X^T X = U \Sigma U^T$$
$$\Rightarrow X = \Sigma^{1/2} U^T$$

ISOMAP

- geodesic distance

Proportion of variance explained by p dimensions

$$\frac{\sum_{i=1}^{p} \lambda_i}{\sum_{i=1}^{n-1} \lambda_i}$$

λ = eigenvalue

Locally Linear Embedding

High Dim: $\min \sum_i \| x_i - \sum_{j \in N(x_i)} w_{ij} x_j \|_2^2$ s.t. $\sum_{j \in N(x_i)} w_{ij} = 1$ (linear constraint)

Low Dim: $\sum_i \| x_i' - \sum_{j \in N(x_i)} w_{ij} x_j' \|_2^2$ s.t. $\frac{1}{N} \sum_i x_i' \cdot x_i'^T = I$ (identity covariance)

and $\sum_i x_i' = \emptyset$ (zero mean)

Modification of step 2 (high dim)

translate x_i and x_j by $-t$

set $x_i = t$

$$\Rightarrow \min \| M_i \vec{w}_i \|_2^2$$
$$\Rightarrow M_i^T M_i w_i = \lambda w_i$$

PA Meta-Summary (4)

Laplacian Eigenmaps

1. Build Adjacency graph
2. compute affinities
3. Perform Eigendecomposition
4. Low-Dim Embedding

Graph Laplacian

$L = D - W$

$d_{ij} = \begin{cases} \sum_{k=1}^{N} w_{ik}, & i = j \\ \emptyset & \text{otherwise} \end{cases}$

$w_{ij} = \begin{cases} \text{weight } i \leftrightarrow j, & i \neq j \\ \emptyset & \text{otherwise} \end{cases}$

Heat Kernel

$w_{ij} = e^{-\|x_i - x_j\|_2^2}$

Binary Affinity

$w_{ij} = \begin{cases} 1, & \|x_i - x_j\| \leq t \\ \emptyset, & \text{otherwise} \end{cases}$

Objective Function

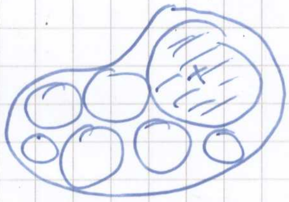
$$\min \sum_{i=1}^{N} \sum_{j=1}^{N} \|x_i' - x_j'\|_2^2 w_{ij}$$

$$= \min \vec{x}'^T L \vec{x}' \quad \text{s.t.} \quad \vec{x}'^T D \vec{x}' = 1$$

$$\Rightarrow D^{-1} L \vec{x}' = \lambda \vec{x}'$$

Manifold Forests

1. Partition Feature space via Density Forest Training
2. Use R_j's as neighborhoods $\rightarrow$ similarity matrix
3. Apply Laplacian Eigenmaps



Affinity Model

$w_{ij} = e^{-Q(x_i, x_j)}$ $\Leftarrow$ affinity

$Q(x_i, x_j)$ $\Leftarrow$ distance function

Mahalanobis

$Q(x_i, x_j) \begin{cases} d_{ij}^T \left(\Lambda_{\ell(x_i)}\right)^{-1} d_{ij} & \text{if in same leaf} \\ \infty & \text{otherwise} \end{cases}$

Binary

$Q(x_i, x_j) = \begin{cases} 0 & \text{if in same leaf} \\ 1 & \text{otherwise} \end{cases}$