Information-Theoretic Clustering in Nonlinear Encoder Models

Felix Agakov

(University of Edinburgh, UK)

David Barber

(IDIAP, Switzerland)

July 4, 2005

Overview

Information-theoretic clustering in encoder models:

- conceptually simple
- probabilistic (soft)
- kernelizable and applicable to unsupervized learning of kernel functions
- computationally attractive (no need to compute eigenvalues or inverses of the Gram matrix)
- favorably compares with common clustering methods in some cases

Overview

Information-theoretic clustering in encoder models:

- conceptually simple
- probabilistic (soft)
- kernelizable and applicable to unsupervized learning of kernel functions
- computationally attractive (no need to compute eigenvalues or inverses of the Gram matrix)
- favorably compares with common clustering methods in some cases

Work in progress...

Clarifying the definitions

What is a cluster?

- “Cluster’s members should be close to each other”
- “Bunch’s organs should shut together” (automatic translation into Russian and back)
- “A cluster is something found by a clustering algorithm” (an anonymous machine learner)

Clarifying the definitions

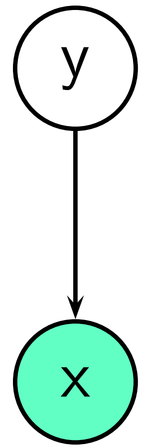
What is a cluster?

- “Cluster’s members should be close to each other”
- “Bunch’s organs should shut together” (automatic translation into Russian and back)
- “A cluster is something found by a clustering algorithm” (an anonymous machine learner)

- a set whose members should satisfy local smoothness constraints (need to constrain the model)
- it is undesirable to assign unique labels to outliers (high marginal entropy of cluster labels?)

Encoder *vs* Generative Models

Let $\mathsf{x} \in \mathbb{R}^{|\mathsf{x}|}$ be a visible pattern, and $y \in \{y_1, \ldots, y_{|y|}\}$ its discrete unknown cluster label

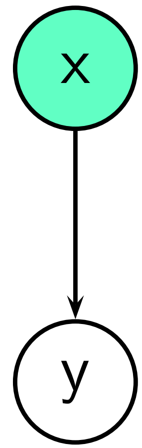


$\mathcal{M}_L$

- generative models: $\mathcal{M}_L \stackrel{\text{def}}{=} p(\mathsf{y})p(\mathsf{x}|\mathsf{y})$
- maximizing the likelihood $\mathcal{L} = \log p(\mathsf{x}^{(1)}, \ldots, \mathsf{x}^{(M)})$
- **problems with under-constrained models**

Encoder *vs* Generative Models

Let $\mathsf{x} \in \mathbb{R}^{|\mathsf{x}|}$ be a visible pattern, and $y \in \{y_1, \ldots, y_{|y|}\}$ its discrete unknown cluster label



$\mathcal{M}_I$

- encoder models: $\mathcal{M}_I \stackrel{\text{def}}{=} \tilde{p}(\mathsf{x})p(\mathsf{y}|\mathsf{x})$
- an "*unsupervised discriminative*" framework
- maximizing the likelihood is meaningless...

Information-Theoretic Clustering

- Goal: learn a mapping $\mathsf{x} \to y$
- interprete cluster labels y as unknown codes
- maximize coding efficiency

$$I(\mathsf{x}, y) \stackrel{\text{def}}{=} H(\mathsf{x}) - H(\mathsf{x}|y) \equiv H(y) - H(y|\mathsf{x})$$

- $H(y) \equiv -\langle \log p(y) \rangle_{p(y)}$, $H(y|\mathsf{x}) \equiv -\langle \log p(y|\mathsf{x}) \rangle_{p(y|\mathsf{x})\tilde{p}(\mathsf{x})}$, $\tilde{p}(\mathsf{x})$ is the empirical distribution
- (Arimoto, Blahut ’72; Linsker ’88)

Information-Theoretic Clustering

- Goal: learn a mapping $\mathsf{x} \to y$
- interprete cluster labels y as unknown codes
- maximize coding efficiency

$$I(\mathsf{x}, y) \stackrel{\text{def}}{=} H(\mathsf{x}) - H(\mathsf{x}|y) \equiv H(y) - H(y|\mathsf{x})$$

- $H(y) \equiv -\langle \log p(y) \rangle_{p(y)}$, $H(y|\mathsf{x}) \equiv -\langle \log p(y|\mathsf{x}) \rangle_{p(y|\mathsf{x})\tilde{p}(\mathsf{x})}$, $\tilde{p}(\mathsf{x})$ is the empirical distribution
- (Arimoto, Blahut ’72; Linsker ’88)
- Generally quite difficult (entropy of a mixture $H(y)$)...
- but tractable for clustering

Information-Theoretic Clustering: Motivation

- Generative *vs* encoder models: what is more attractive?

Information-Theoretic Clustering: Motivation

- Generative *vs* encoder models: what is more attractive?

Generative models:

- $p(\mathrm{x}|y)$ **must** be a correctly normalized distribution in $|\mathrm{x}|$-dimensional space
- $p(\mathrm{x})$ will typically be a mixture of simple distributions (e.g. Gaussians)
- a poor fit to curved manifolds unless $|y|$ is large

Information-Theoretic Clustering: Motivation

- Generative *vs* encoder models: what is more attractive?

Generative models:

- $p(\mathrm{x}|y)$ **must** be a correctly normalized distribution in $|\mathrm{x}|$-dimensional space
- $p(\mathrm{x})$ will typically be a mixture of simple distributions (e.g. Gaussians)
- a poor fit to curved manifolds unless $|y|$ is large

Encoder models:

- $p(y|\mathrm{x})$ may be very complex
- $I(\mathrm{x}, y) = H(y) - H(y|\mathrm{x})$ implicitly favors *equiprobable deterministic* cluster assignments

Learning Optimal Parameters

- Constrain $p(y|\mathsf{x})$ to satisfy local smoothness
- A simple choice of the encoder is

$$p(y_j|\mathsf{x}^{(i)}) \propto \exp\{-\|\mathsf{x}^{(i)} - \mathsf{w}_j\|^2/s_j + b_j\},$$

(probability of assigning $\mathsf{x}^{(i)}$ to cluster y_j)

- maximize $I(\mathsf{x}, y)$ for cluster centers $\mathsf{w}_j \in \mathbb{R}^{|\mathsf{x}|}$, dispersions s_j, and biases b_j

Learning Optimal Parameters

- Constrain $p(y|\mathsf{x})$ to satisfy local smoothness
- A simple choice of the encoder is

$$p(y_j|\mathsf{x}^{(i)}) \propto \exp\{-\|\mathsf{x}^{(i)} - \mathsf{w}_j\|^2/s_j + b_j\},$$

(probability of assigning $\mathsf{x}^{(i)}$ to cluster y_j)

- maximize $I(\mathsf{x}, y)$ for cluster centers $\mathsf{w}_j \in \mathbb{R}^{|\mathsf{x}|}$, dispersions s_j, and biases b_j
- $p(y|\mathsf{x})$ is similar to the posterior of Gaussian mixtures
- $\mathcal{M}_I = \tilde{p}(\mathsf{x})p(y|\mathsf{x})$ is trained by maximizing $I(\mathsf{x}, y)$

Learning Optimal Parameters (Cont.)

- Nonlinear ascent on $I(\mathsf{x}, y)$ with

$$\frac{\partial I(\mathsf{x}, y)}{\partial \mathsf{w}_j} = \frac{1}{M} \sum_{m=1}^{M} p(y_j|\mathsf{x}^{(m)}) \frac{(\mathsf{x}^{(m)} - \mathsf{w}_j)}{s_j} \alpha_j^{(m)}$$

$$\frac{\partial I(\mathsf{x}, y)}{\partial s_j} = \frac{1}{M} \sum_{m=1}^{M} p(y_j|\mathsf{x}^{(m)}) \frac{\|\mathsf{x}^{(m)} - \mathsf{w}_j\|^2}{2s_j^2} \alpha_j^{(m)}$$

- Coefficients $\alpha_j^{(m)}$:

$$\alpha_j^{(m)} \stackrel{\text{def}}{=} \log \frac{p(\mathsf{x}^{(m)}|y_j)}{p(\mathsf{x}^{(m)})} - KL\left(p(y|\mathsf{x}^{(m)}) \| \langle p(y|\mathsf{x}) \rangle_{\tilde{p}(\mathsf{x})}\right)$$

- (*cf* ML for mixtures of Gaussians)

Clustering in Nonlinear Encoder Models

- Nonlinear encoders:

$$p(y_j|\mathrm{x}^{(i)}) \propto \exp\{-\|\boldsymbol{\phi}(\mathrm{x}^{(i)}) - \mathrm{w}_j\|^2/s_j + b_j\},$$

- $\boldsymbol{\phi}(\mathrm{x}^{(i)}) \in \mathbb{R}^{|\boldsymbol{\phi}|}$ is a *feature* vector for pattern $\mathrm{x}^{(i)}$
- $|\boldsymbol{\phi}|$ may be ∞-dimensional.
- $\mathrm{x}^{(i)}$, $\mathrm{x}^{(k)}$ are likely to be clustered as y_j if they lie close to an unknown cluster center w_j in a *feature space*

Clustering in Nonlinear Encoder Models

- Nonlinear encoders:

$$p(y_j|\mathsf{x}^{(i)}) \propto \exp\{-\|\boldsymbol{\phi}(\mathsf{x}^{(i)}) - \mathsf{w}_j\|^2/s_j + b_j\},$$

- Kernelization is straight-forward:

$$\mathsf{K} \stackrel{\text{def}}{=} \{K_{ij}\} \stackrel{\text{def}}{=} \{\boldsymbol{\phi}(\mathsf{x}^{(i)})^T\boldsymbol{\phi}(\mathsf{x}^{(j)})\} = \mathcal{K}(\boldsymbol{\Theta}) \in \mathbb{R}^{M\times M}$$

- $\mathsf{w}_j = \sum_{m=1}^{M} \alpha_{mj}\boldsymbol{\phi}(\mathsf{x}^{(m)}) + \mathsf{w}_j^{\perp}$, where $(\mathsf{w}_j^{\perp})^T\boldsymbol{\phi}(\mathsf{x}^{(m)}) = 0$
- maximize $I(\mathsf{x}, y)$ for $\{\alpha_{jm}\}$, s_j, b_j, and kernel parameters $\boldsymbol{\Theta}$
- (Again, numerical ascent on $I(\mathsf{x}, y)$)

Learning Kernels

$$\frac{\partial I(\mathsf{x}, y)}{\partial \mathbf{\Theta}} = \frac{1}{M}\sum_{m=1}^{M} KL(p(y|\mathsf{x}^{(m)})\|p(y))\sum_{k=1}^{|y|}\frac{\partial f_k(\mathsf{x}^{(m)})}{\partial \mathbf{\Theta}} p(y_k|\mathsf{x}^{(m)}) -$$

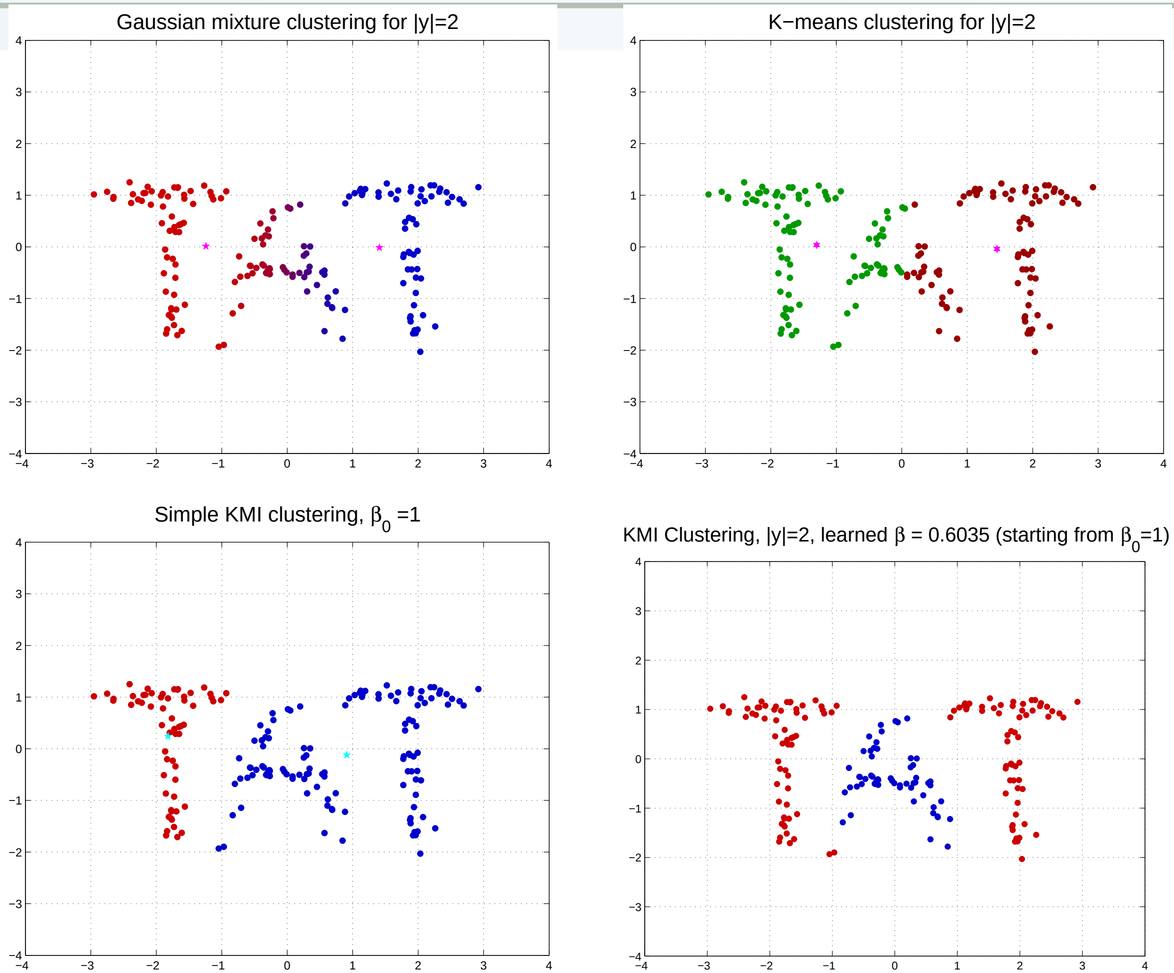
$$\frac{1}{M}\sum_{m=1}^{M}\sum_{j=1}^{|y|}\frac{\partial f_j(\mathsf{x}^{(m)})}{\partial \mathbf{\Theta}} p(y_j|\mathsf{x}^{(m)}) \log \frac{p(y_j|\mathsf{x}^{(m)})}{p(y_j)}$$

- $p(y_j|\mathsf{x}^{(m)}) \propto \exp\{-f_j(\mathsf{x}^{(m)})\}$
- potentials $f_j(\mathsf{x}^{(m)})$:

$$f_j(\mathsf{x}^{(m)}) \equiv \left\{-\left(K_{mm} - 2\mathsf{k}^T(\mathsf{x}^{(m)})\mathsf{a}_j + \mathsf{a}_j^T \mathsf{K} \mathsf{a}_j + c_j\right)/s_j\right\}$$

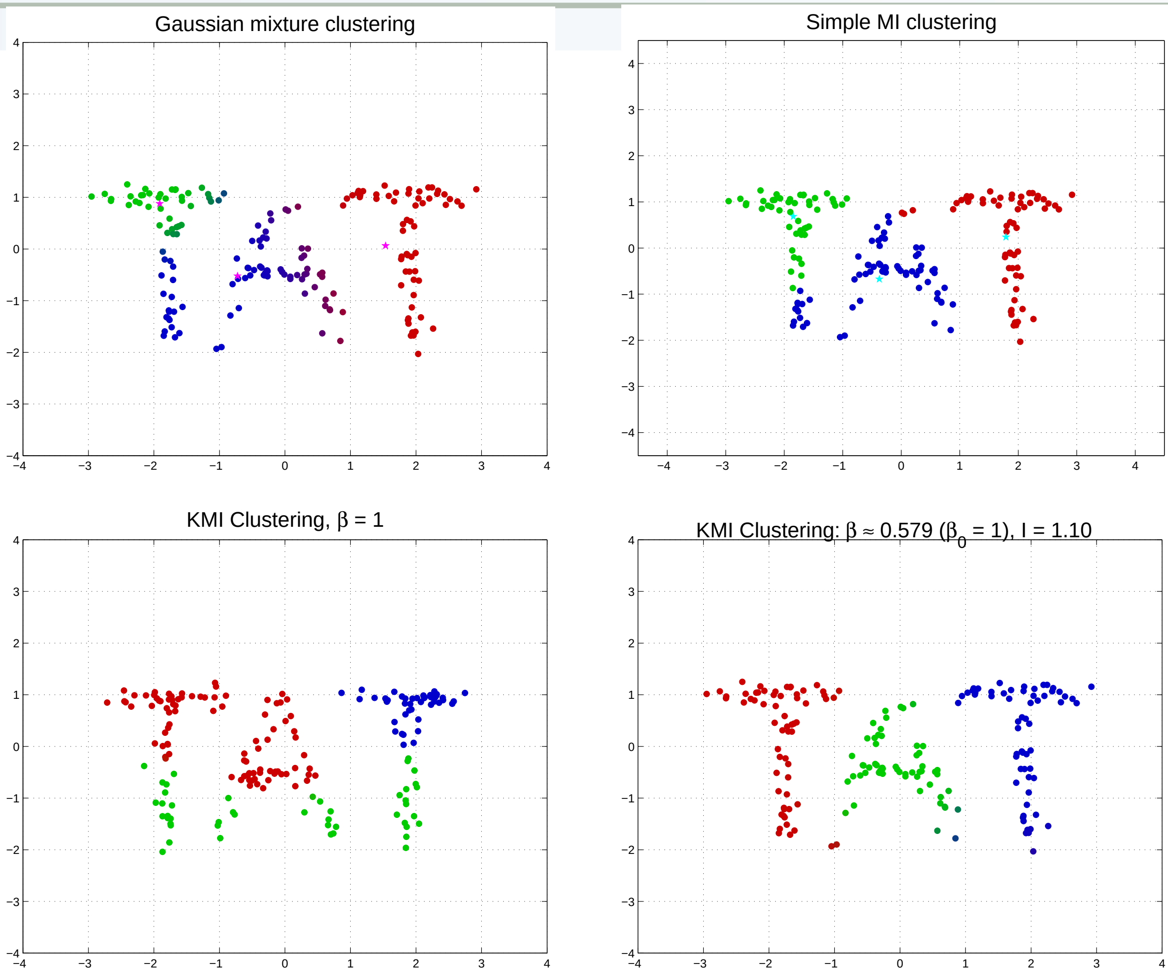
- numerical ascent on $I(\mathsf{x}, y) \sim O(M|y|^2)$
- no need to compute eigenvalues of $\mathsf{K} \in \mathbb{R}^{M\times M}$

Experiments:



- Clustering: $p(y_j|\mathrm{x}^{(i)}) \propto \{-\|\boldsymbol{\phi}(\mathrm{x}^{(i)}) - \mathrm{w}_j\|^2/s_j\}$
- unsupervised clustering, nonlinear encoder
- favorably compares with GMMs, k-means, kernel k-means, normalized cuts [Ng et. al. '01], non-kernelized MI, fixed-kernel KMI

Experiments:



- Clustering: $p(y_j|\mathrm{x}^{(i)}) \propto \{-\|\phi(\mathrm{x}^{(i)}) - \mathrm{w}_j\|^2/s_j\}$
- unsupervised clustering, nonlinear encoder
- favorably compares with GMMs, k-means, kernel k-means, normalized cuts [Ng et. al. '01], non-kernelized MI, fixed-kernel KMI

Summary

- unsupervised information-theoretic clustering
- extracts clusters directly from the dataset
- conceptually simple
- suggests a principled way to learn the kernels
- potentially generalizable to other encoder models

Still need: practical applications; theoretical analysis (links to wheighted annealed feature-space k-means?)