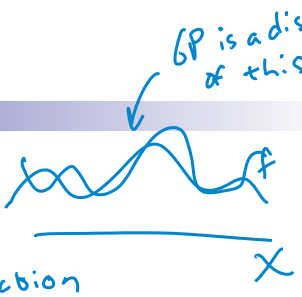
Module 3: Bayesian Nonparametrics

Gaussian Processes cont'd

STAT/BIOSTAT 527, University of Washington
Emily Fox
May 2nd, 2013

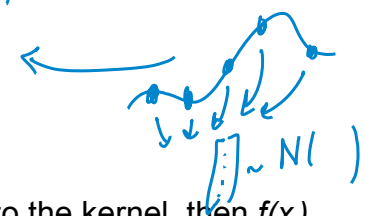
Gaussian Processes

- Distribution on functions
  - $f \sim GP(m,\kappa)$
    - m: mean function
    - κ: covariance function = kernel function



⇕ iff $\forall n$ and any $x_1, \ldots, x_n$

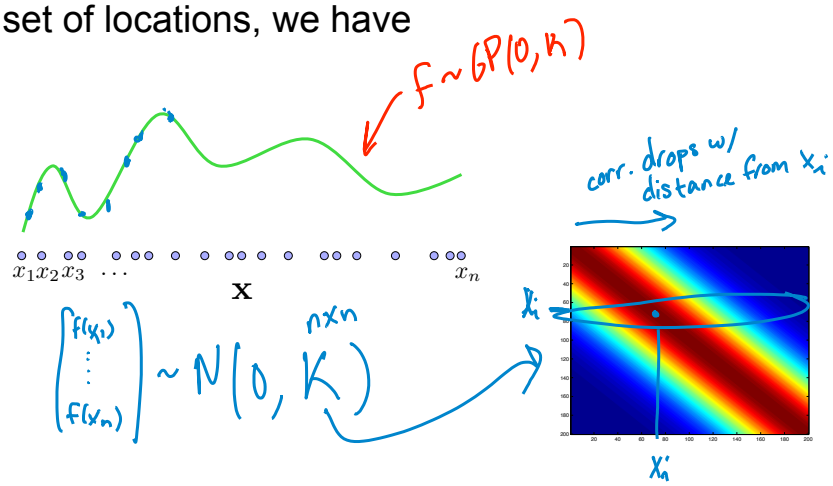
  - $p(f(x_1), \ldots, f(x_n)) \sim N_n(\mu, K)$
    - $\mu = [m(x_1),\ldots,m(x_n)]$
    - $K_{ij} = \kappa(x_i,x_j)$ Gram matrix



- Idea: If x_i, x_j are similar according to the kernel, then $f(x_i)$ is similar to $f(x_j)$ similar outputs captured by K

Induced Multivariate Gaussian

- Evaluating the GP-distributed function at any set of locations, we have



Relating GPs to Splines

- Recall smoothing spline objective

$$\min_f \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \int f''(x)^2 dx$$

- Consider the following model

$$f(x) = \beta_0 + \beta_1 x + r(x)$$

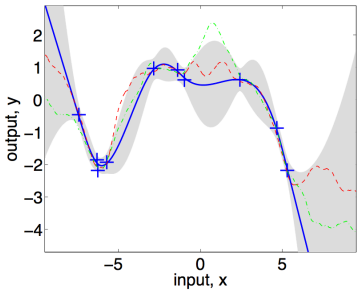
where $r(x) \sim GP(0, \sigma_f^2 K_{sp}(x, x'))$

$K_{sp}(x, x') \triangleq \int_0^1 (x-u)_+ (x'-u)_+ du$

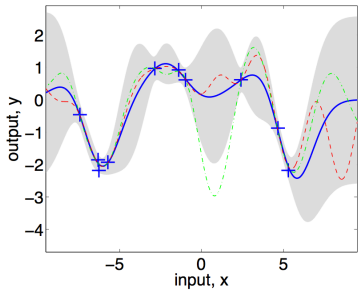
- One can show that the MAP estimate of $f(x)$ is a ***cubic smoothing spline*** when $p(\beta_j) \propto 1$ (β_0, β_1) ← don't penalize 0th + 1st order terms
- Penalty parameter λ is now given by σ_y^2 / σ_f^2

Relating GPs to Splines

- The spline kernel leads to a smooth posterior mode/mean, but posterior samples are not smooth.
  - Again, as in lasso, regularizers do not always make good priors



(a), spline covariance



(b), squared exponential cov.

Figure from Rasmussen and Williams 2006

- See Rasmussen and Williams 2006 for more details

GP Regression Recap

Bayesian treatment of **Linear Basis Expansion** ← special case w/ a finite rank kernel — **Gaussian Process**

Prior

$$\beta \sim N(0, \alpha^{-1} I_M)$$

$$f(x) = \sum_{m=1}^{M} \beta_m \phi_m(x)$$

specify basis

$$f \sim \mathrm{GP}(0, \kappa(x, x'))$$

specify cov. fcn

Distribution on $x_1, \ldots, x_n$

$$\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix} = f \sim N(0, \alpha^{-1} \Phi \Phi^T)$$

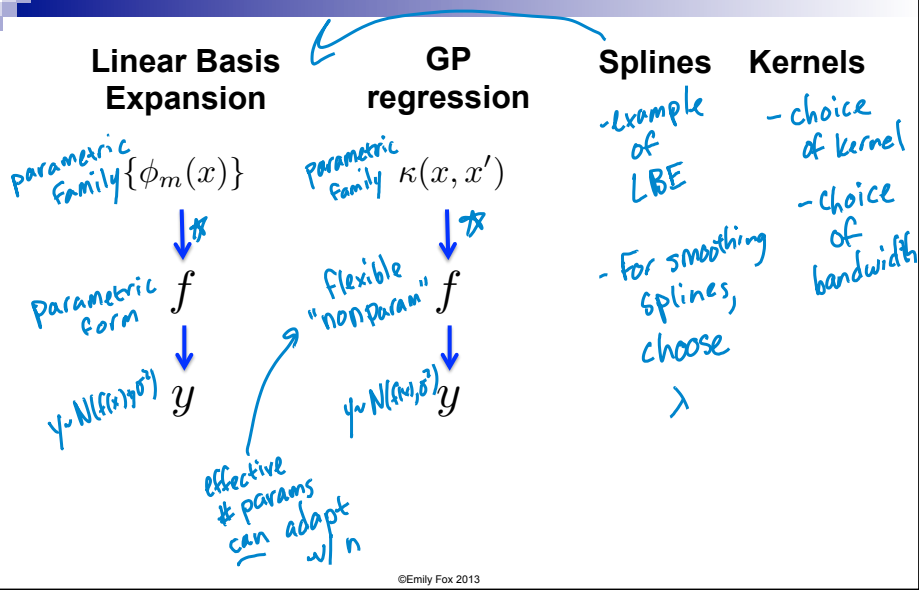
$$f \sim N(0, K)$$

Choices

- *Choose M*
- *Choose bases*

- *Choose* $\kappa(x, x')$
- *Choose covariance hyperparameters*

GP Regression Recap



Effective Degrees of Freedom

- For the training set, the fit is given by

$$\hat{f} = K(K + \sigma_y^2 I_n)^{-1} y$$

- Since *K* is a positive definite Gram matrix, it has eigendecomp

$$K = \sum_{i=1}^{n} \lambda_i u_i u_i^T$$

- Using this, one can show that $K(K + \sigma_y^2 I_n)^{-1}$ has eigenvals

$$\frac{\lambda_i}{\lambda_i + \sigma_y^2}$$

- Therefore, the effective degrees of freedom is

$$\nu = \mathrm{tr}\left(K(K + \sigma_y^2 I_n)^{-1}\right) = \sum_{i=1}^{n} \frac{\lambda_i}{\lambda_i + \sigma_y^2}$$

can grow w/ n

fcn of how quickly eigvals decay

- Remember that this specifies how "wiggly" the curve is

Choice of Covariance Function

- Definitions
  - ***Stationary*** kernel – only depends on $x - x'$ e.g. SE
  - ***Isotropic*** kernel – furthermore only depends on $||x - x'||$ e.g. RBF
- Examples
  - ***Squared exponential*** – $\kappa_{SE}(r) = e^{-\frac{r}{2\ell^2}}$
    - Kernel is infinitely differentiable → GP has mean square derivatives of all orders → resulting functions are very smooth
  - ***Matern*** – $\kappa_{Matern}(r) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu} r}{\ell} \right)^{\nu} K_v \left(\frac{\sqrt{2\nu} r}{\ell} \right)$
    - When $\nu \to \infty$: squared exponential
    - When $\nu = \frac{1}{2}$: exponential kernel $\kappa_{exp}(r) = e^{-\frac{r}{\ell}}$ ** equal to Brownian motion in 1D **

Sample Paths using Matern Kernel

- Can produce very rough sample paths

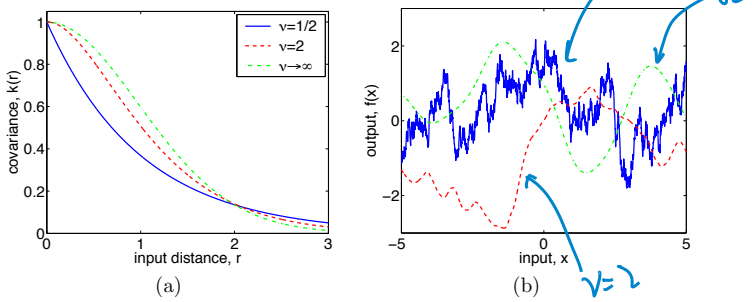
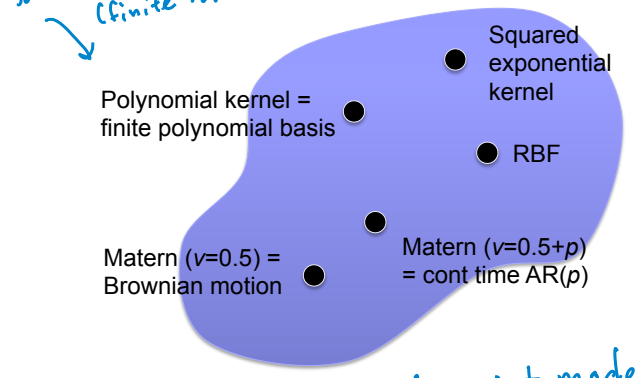


Figure from Rasmussen and Williams 2006

Family of Gaussian Processes

saw this example (finite rank kernel)



Many processes we know + models we consider can be posed as GPs.

Module 3: Bayesian Nonparametrics

Finite Mixture Models

for density estimation

STAT/BIOSTAT 527, University of Washington

Emily Fox

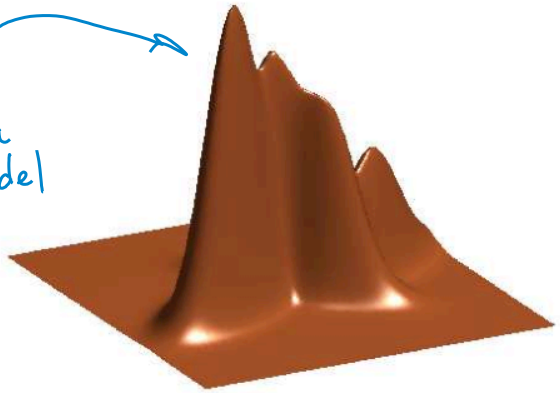
May 2nd, 2013

Density Estimation

- Estimate a density based on $x_1,\ldots,x_N$

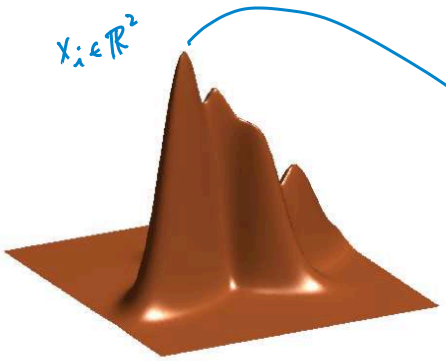
$x_1,\ldots,x_N \sim P$

Let's consider a parametric model

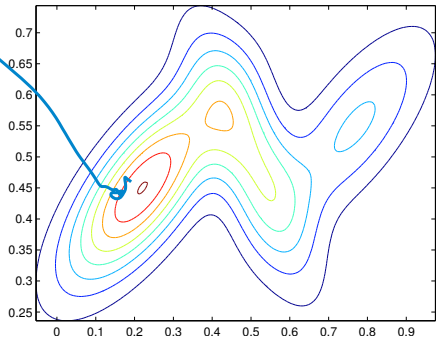


Density Estimation

$x_i \in \mathbb{R}^2$



Contour Plot of Joint Density

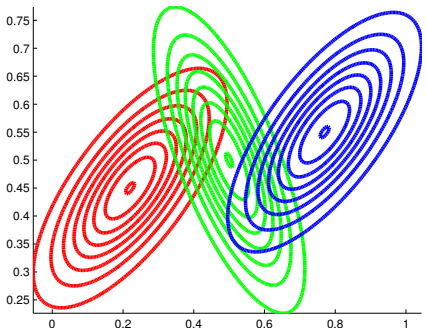


bird's eye view

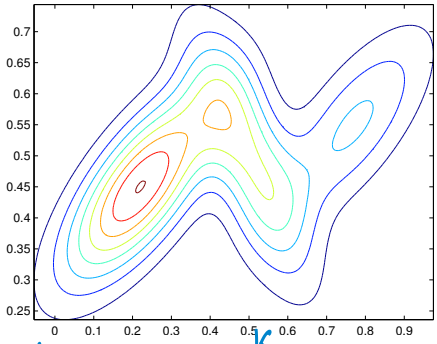
Density as Mixture of Gaussians

- Approximate density with a mixture of Gaussians

Mixture of 3 Gaussians



Contour Plot of Joint Density

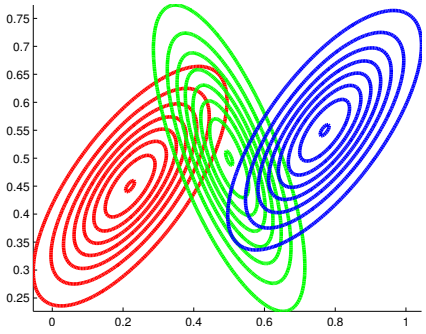


Each Gaussian has weight π_k w/ $\sum_{k=1}^{K} \pi_k = 1$ and shape params μ_k, Σ_k

Density as Mixture of Gaussians

- Approximate density with a mixture of Gaussians

Mixture of 3 Gaussians



$$p(x_i \mid \pi, \mu, \Sigma) = \sum_{k=1}^{K} \pi_k \, N(x_i \mid \mu_k, \Sigma_k)$$

Gauss. kernel, just like in KDE, but not centered at obs.

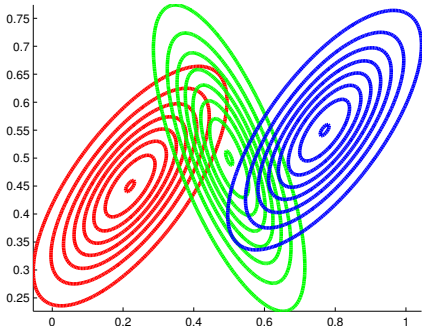
In 1D: P = target density

$\sum \pi_k = 1$

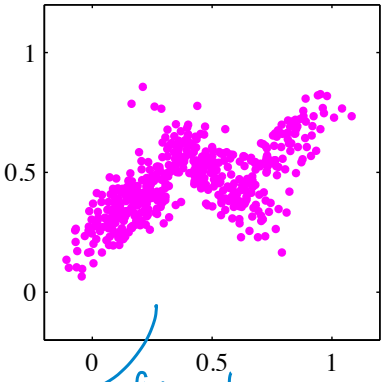
Density as Mixture of Gaussians

- Approximate with density with a mixture of Gaussians

Mixture of 3 Gaussians



Our actual observations



How?!?!

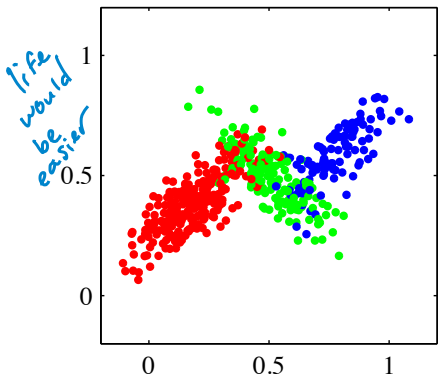
from obs., est. model params

C. Bishop, Pattern Recognition & Machine Learning

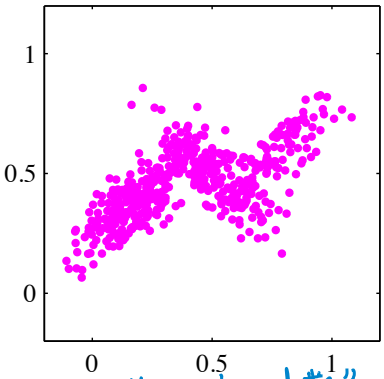
Clustering our Observations

- Imagine we have an assignment of each x_i to a Gaussian

Our actual observations



Complete data labeled by true cluster assignments

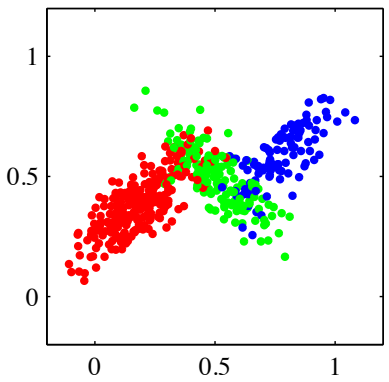


"Incomplete data"

C. Bishop, Pattern Recognition & Machine Learning

Clustering our Observations

- Imagine we have an assignment of each x_i to a Gaussian



Complete data labeled by true cluster assignments

- Introduce latent cluster indicator variable z_i

$z_i \in \{1, \ldots, K\}$

$\Pr(z_i = k) = \pi_k$

- Then we have

$$p(x_i \mid z_i = k, \pi, \mu, \Sigma) =$$

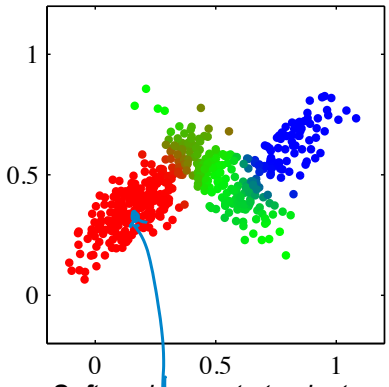
$$N(x_i \mid \mu_{z_i = k}, \Sigma_{z_i = k})$$

param. est. is easy if we have $\{z_i\}$ $\Rightarrow$ decoupled into K Gauss. est.

C. Bishop, Pattern Recognition & Machine Learning

Clustering our Observations

- We must infer the cluster assignments from the observations



Soft assignments to clusters

- Posterior probabilities of assignments to each cluster *given* model parameters:

$$r_{ik} = p(z_i = k \mid x_i, \pi, \theta) =$$

$$= \frac{\pi_k N(x_i \mid \mu_k, \Sigma_k)}{\sum_j \pi_j N(x_i \mid \mu_j, \Sigma_j)}$$

motivates an iterative alg.

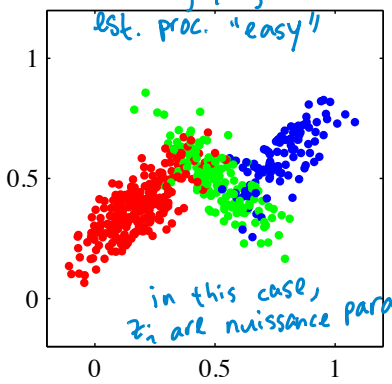
C. Bishop, Pattern Recognition & Machine Learning

Summary of GMM Concept

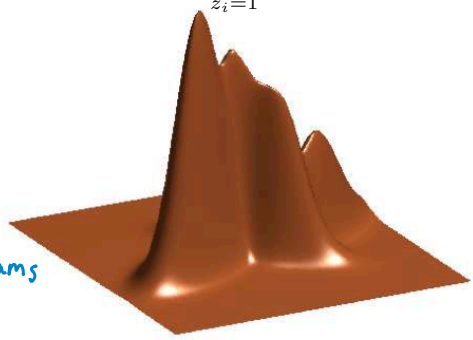
- Estimate a density based on $x_1, \ldots, x_N$

$$p(x_i \mid \pi, \mu, \Sigma) = \sum_{z_i=1}^{K} \pi_{z_i} \mathcal{N}(x_i \mid \mu_{z_i}, \Sigma_{z_i})$$

introducing $\{z_i\}$ makes P= est. proc. "easy"



Complete data labeled by true cluster assignments



Surface Plot of Joint Density, Marginalizing Cluster Assignments

Summary of GMM Components

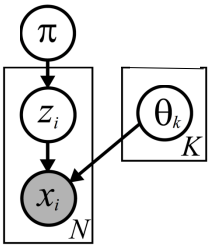
- Observations $x_i \in \mathbb{R}^d, \quad i = 1, 2, \ldots, N$
- Hidden cluster labels $z_i \in \{1, 2, \ldots, K\}, \quad i = 1, 2, \ldots, N$
- Hidden mixture means $\mu_k \in \mathbb{R}^d, \quad k = 1, 2, \ldots, K$
- Hidden mixture covariances $\Sigma_k \in \mathbb{R}^{d \times d}, \quad k = 1, 2, \ldots, K$
- Hidden mixture probabilities $\pi_k, \quad \sum_{k=1}^{K} \pi_k = 1$

Gaussian mixture marginal and conditional likelihood **:**

$$p(x_i \mid \pi, \mu, \Sigma) = \sum_{z_i=1}^{K} \pi_{z_i} \mathcal{N}(x_i \mid \mu_{z_i}, \Sigma_{z_i})$$

$$p(x_i \mid z_i, \pi, \mu, \Sigma) = \mathcal{N}(x_i \mid \mu_{z_i}, \Sigma_{z_i})$$

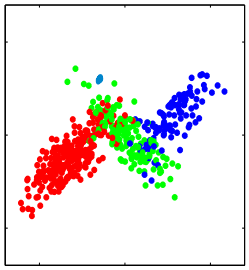
Generative Model



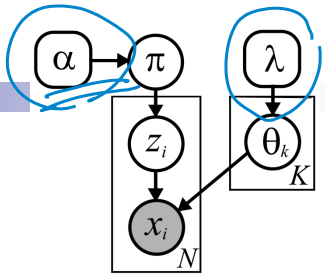
- We can think of *sampling* observations from the model
- For the GMM, define model parameters
  - Cluster means and covariances $\{\mu_k, \Sigma_k\} \triangleq \theta_k$
  - Cluster weights $\pi = [\pi_1, \ldots, \pi_K]$
- For each observation *i*,
  - Sample a cluster assignment

    $z_i \sim \pi$
  - Sample the observation from the selected Gaussian

    $x_i | z_i \sim N(x_i | \mu_{z_i}, \Sigma_{z_i})$



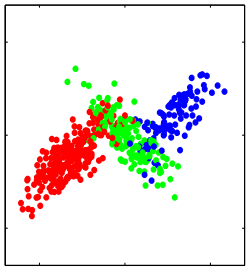
A Bayesian GMM



- In a Bayesian approach, we place priors on the model parameters
- Conjugate priors are a computationally convenient choice
- Conjugate prior for $\theta_k = \{\mu_k, \Sigma_k\}$
  - Known variance: Gaussian prior on mean
  - Unknown mean & variance: *normal inverse-Wishart* (NIW)
- Conjugate prior for π ???

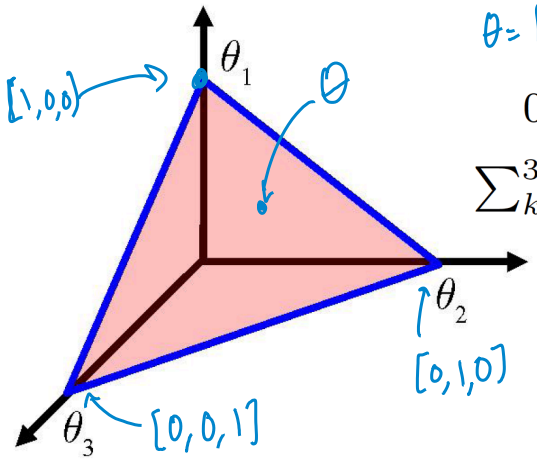
  Recall $\sum \pi_k = 1$

  $\Rightarrow \pi$ lives on the simplex



The Simplex in 3D

- The simplex defines the hyperplane of vectors that sum to 1



$\theta = [\theta_1, \ldots, \theta_K]$

$$0 \leq \theta_k \leq 1$$

$$\sum_{k=1}^{3} \theta_k = 1$$

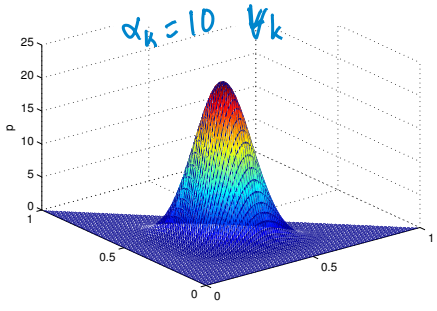
Dirichlet Distributions

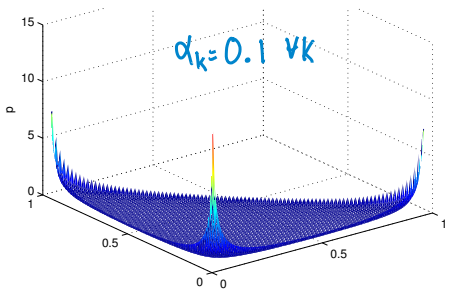
- The Dirichlet distribution is defined on the simplex

$\pi \sim \text{Dir}(\alpha_1, \ldots, \alpha_K)$

$\Rightarrow \sum \pi_k = 1$

$$p(\pi \mid \alpha) = \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \prod_{k=1}^{K} \pi_k^{\alpha_k - 1}$$

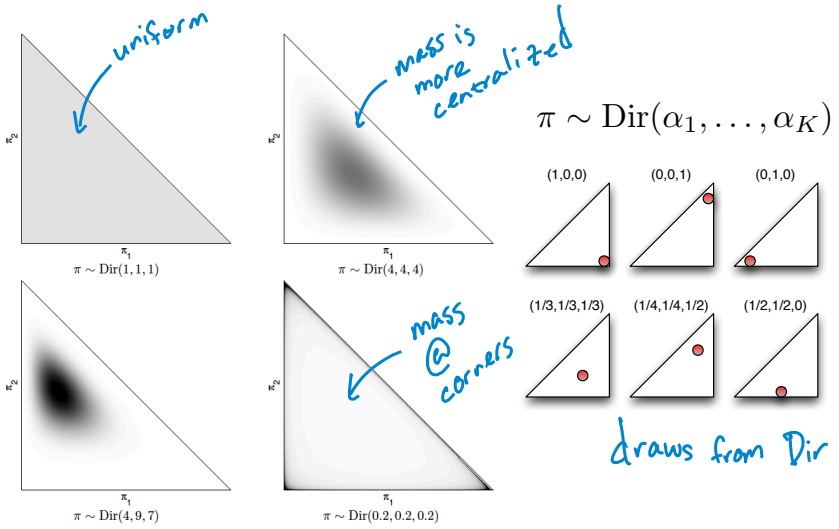




Moments: $\mathbb{E}_\alpha[\pi_k] = \frac{\alpha_k}{\alpha_0}$

$$\text{Var}_\alpha[\pi_k] = \frac{K-1}{K^2(\alpha_0 + 1)}$$

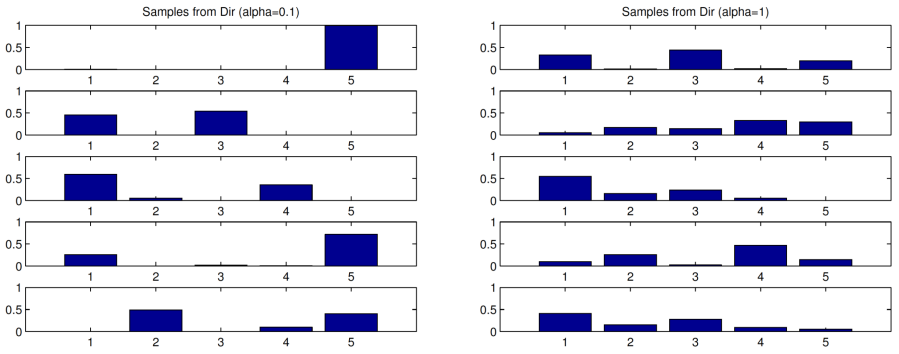
Dirichlet Probability Densities



Dirichlet Samples

$$\mathbb{E}_\alpha[\pi_k] = \frac{\alpha_k}{\alpha_0}$$

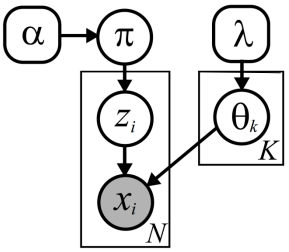
- Samples are ***sparse*** for small values of α_i



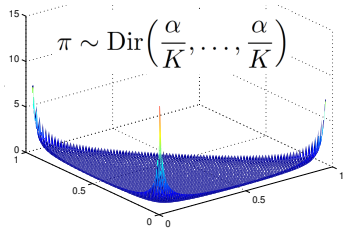
$\mathrm{Dir}(\pi \mid 0.1, 0.1, 0.1, 0.1, 0.1)$ puts mass @ corners

$\mathrm{Dir}(\pi \mid 1.0, 1.0, 1.0, 1.0, 1.0)$ uniform

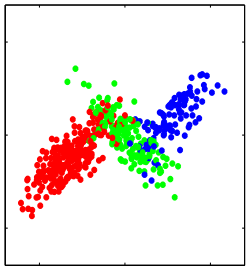
Model Summary



- Prior on model parameters
  - E.g., symmetric Dirichlet for π



  - Normal inverse Wishart prior for θ_k

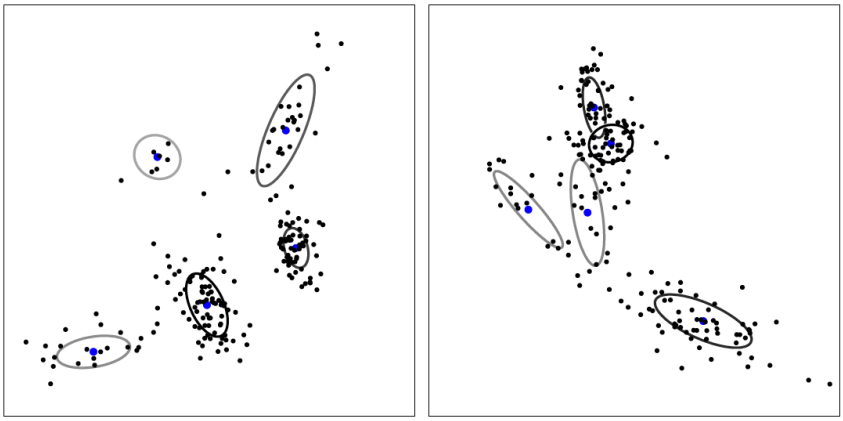


- Sample observations as

$$z_i \sim \pi$$

$$x_i \mid z_i \sim N(\mu_{z_i}, \Sigma_{z_i})$$

Samples Generated from GMM



Posterior Computations

- From our observations, we want to infer model params
- MAP estimation can be done using expectation maximization (EM) algorithm:

$$\hat{\theta}^{MAP} = \arg\max_{\theta} p(\theta \mid x)$$

- What if we want a full characterization of the posterior?
  - Maintain a measure of uncertainty
  - Estimators other than posterior mode (different loss functions)
  - Predictive distributions for future observations

$$p(x_{N+1} \mid x_1, \ldots, x_N) = \int p(x_{N+1} \mid \theta) p(\theta \mid x_1, \ldots, x_N) d\theta$$

- Often no closed-form characterization (e.g., mixture models)
- Alternatives:
  - Markov chain Monte Carlo (MCMC) providing samples from posterior
  - Variational approximations to posterior

Gibb Sampling

- Let z indicate the set of ***all variables in the model***: e.g., cluster indicators and parameters
- Want draws:

$$(z_1, \ldots, z_n) \sim \pi(\underline{z})$$

can think of $\pi(\underline{z}) = p(\theta \mid x)$ (desired posterior)

← issue: can't directly sample

- Construct Markov chain whose steady state distribution is $\pi(\underline{z})$
- Simplest case:

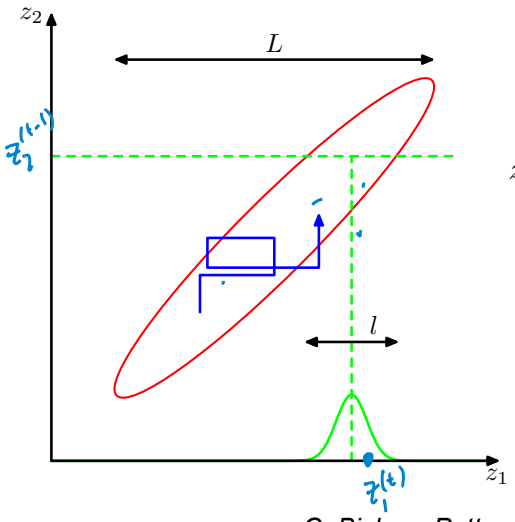
for $t = 1, \ldots, N_{iter}$

  for $i = 1, \ldots, n$

$$z_i^{(t)} \sim p(z_i \mid z_1^{(t)}, \ldots, z_{i-1}^{(t)}, z_{i+1}^{(t-1)}, \ldots, z_n^{(t-1)})$$

Gibb's sampling assumes that this has closed form & can sample

Gibbs Sampler for a 2D Gaussian



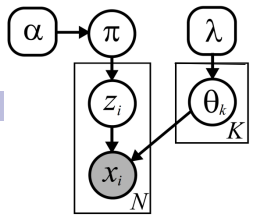
General Gibbs Sampler

$$z_i^{(t)} \sim p(z_i \mid z_{\setminus i}^{(t-1)}) \qquad i = i(t)$$
$$z_j^{(t)} = z_j^{(t-1)} \qquad j \neq i(t)$$

Under mild conditions, converges assuming all variables are resampled infinitely often (order can be fixed or random)

C. Bishop, Pattern Recognition & Machine Learning

Example – GMM



- Recall model
  - Observations: $x_1, \ldots, x_N$
  - Cluster indicators: $z_1, \ldots, z_N$
  - Parameters: π, θ_k → $\pi = [\pi_1, \ldots, \pi_K]$, $\theta_k = \{\mu_k, \Sigma_k\}$

want these

  - Generative model:

$$\pi \sim \mathrm{Dir}(\alpha_1, \ldots, \alpha_K) \qquad z_i \sim \pi$$
$$\{\mu_k, \Sigma_k\} \sim \mathrm{NIW}(\lambda) \qquad x_i \mid z_i, \{\theta_k\} \sim N(\mu_{z_i}, \Sigma_{z_i})$$

- Iteratively sample

$z_i \mid \pi, \{\theta_k\}, \{x_i\}$ $\quad i = 1, \ldots, N$

$\pi \mid \{z_i\}, \{\theta_k\}, \{x_i\}$

$\theta_k \mid \{z_i\}, \{x_i\}$ $\quad k = 1, \ldots K$

Complete Conditional $p(z_i \mid \pi, \{\theta_k\}, \{x_i\})$

- We have

$$z_i \sim \pi$$

$$x_i \mid z_i, \{\theta_k\} \sim N(\mu_{z_i}, \Sigma_{z_i})$$

- As before, we can compute the "responsibility" of each cluster to the observation

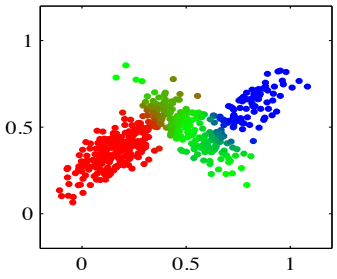
$$r_{ik} = p(z_i = k \mid x_i, \pi, \theta) = \frac{\pi_k p(x_i \mid \theta_k)}{\sum_{\ell=1}^{K} \pi_\ell p(x_i \mid \theta_\ell)}$$

desired complete cond.

- Sample each cluster indicator as

$z_i \sim r_i \quad i = 1, \ldots, N$

$r_i = [r_{i1}, \ldots, r_{iK}]$



Complete Conditional $p(\pi \mid \{z_i\})$

- Recall conjugate Dirichlet prior

$$\pi \sim \mathrm{Dir}(\alpha_1, \ldots, \alpha_K) \qquad p(\pi \mid \alpha) = \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \prod_k \pi_k^{\alpha_k - 1}$$

- Dirichlet posterior
  - Assume we condition on cluster indicators $z_i \sim \pi$
  - Count occurrences of $z_i = k$ $N_k = |\{z_i : z_i = k\}|$
  - Then,

$$p(\pi \mid \alpha, z_1, \ldots, z_N) \propto \prod_i p(z_i \mid \pi)\, p(\pi \mid \alpha)$$

$$\propto \prod_k \prod_{i : z_i = k} \pi_k \cdot \pi_k^{\alpha_k - 1} \propto \prod_k \pi_k^{N_k + \alpha_k - 1}$$

$$= \mathrm{Dir}(N_1 + \alpha_1, \ldots, N_K + \alpha_K)$$

  - Conjugacy: This **posterior** has same form as **prior**

Complete Conditional $p(\theta_k \mid \{z_i\}, \{x_i\})$

- Recall NIW prior…Let's consider 1D example → N-IG

$$\mu_k \mid \sigma_k^2 \sim N(0, \gamma\sigma_k^2) \quad \sigma_k^2 \sim \text{IG}\left(\frac{\nu_0}{2}, \frac{\nu_0 S_0}{2}\right)$$

- Normal inverse gamma posterior
  - Consider observation indices *i* such that $z_i = k$
  - For these observations, $x_i \mid z_i = k \sim N(\mu_k, \Sigma_k)$
  - Then,

$$\mu_k \mid \sigma_k^2, \{z_i\}, \{x_i\} \sim N\left(\frac{1}{N_k + \gamma^{-1}} \sum_{i:z_i=k} x_i, \frac{1}{N_k + \gamma^{-1}}\sigma_k^2\right)$$

only summing over obs. generated from kth cluster

$$\sigma_k^2 \mid \{z_i\}, \{x_i\} \sim \text{IG}\left(\frac{\nu_0 + N_k}{2}, \frac{\nu_0 S_0 + \sum_{i:z_i=k} x_i^2 - (N_k + \gamma^{-1})^{-1}(\sum_{i:z_i=k} x_i)^2}{2}\right)$$

  - Conjugacy: This **posterior** has same form as **prior**

Standard Finite Mixture Sampler

Given mixture weights $\pi^{(t-1)}$ and cluster parameters $\{\theta_k^{(t-1)}\}_{k=1}^K$ from the previous iteration, sample a new set of mixture parameters as follows:

1. Independently assign each of the N data points x_i to one of the K clusters by sampling the indicator variables $z = \{z_i\}_{i=1}^N$ from the following multinomial distributions:

$$z_i^{(t)} \sim \frac{1}{Z_i} \sum_{k=1}^K \pi_k^{(t-1)} f(x_i \mid \theta_k^{(t-1)})\, \delta(z_i, k) \qquad Z_i = \sum_{k=1}^K \pi_k^{(t-1)} f(x_i \mid \theta_k^{(t-1)})$$

2. Sample new mixture weights according to the following Dirichlet distribution:

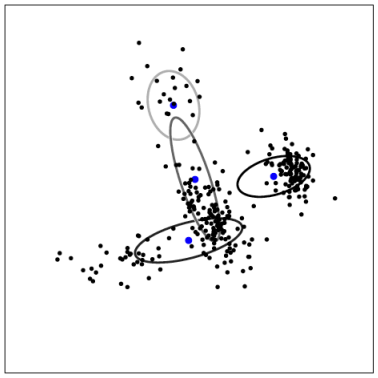
$$\pi^{(t)} \sim \text{Dir}(N_1 + \alpha/K, \ldots, N_K + \alpha/K) \qquad N_k = \sum_{i=1}^N \delta(z_i^{(t)}, k)$$

3. For each of the K clusters, independently sample new parameters from the conditional distribution implied by those observations currently assigned to that cluster:

$$\theta_k^{(t)} \sim p(\theta_k \mid \{x_i \mid z_i^{(t)} = k\}, \lambda)$$

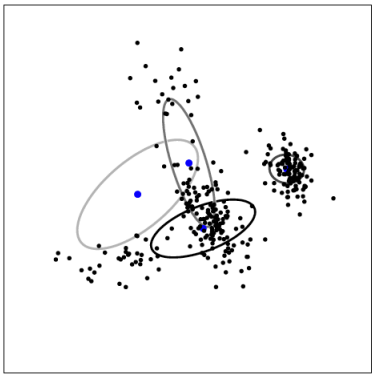
Standard Sampler: 2 Iterations

random init #1



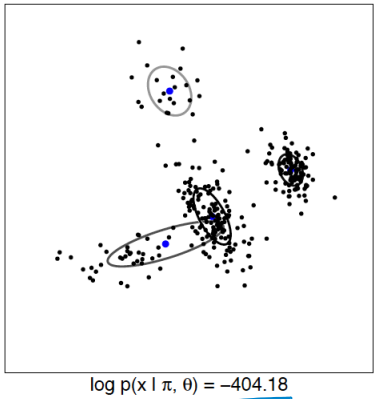
log p(x | π, θ) = –539.17

random init #2



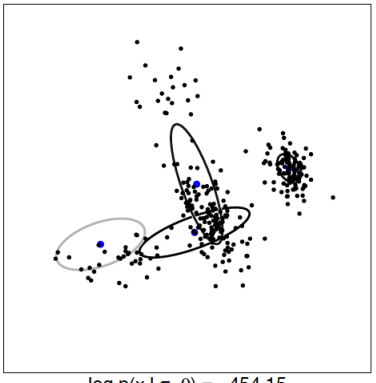
log p(x | π, θ) = –497.77

Standard Sampler: 10 Iterations



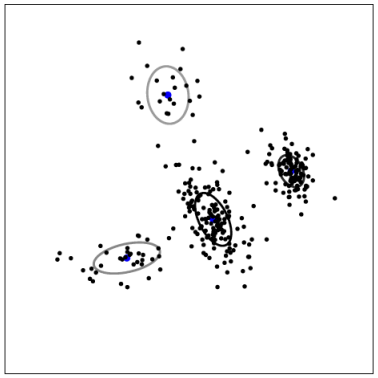
log p(x | π, θ) = –404.18

better

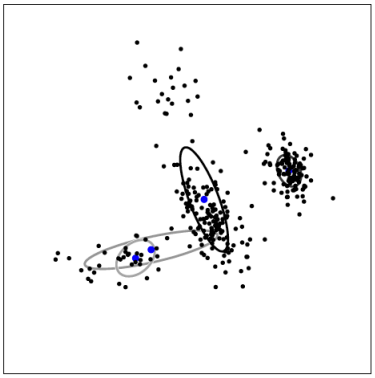


log p(x | π, θ) = –454.15

Standard Sampler: 50 Iterations



log p(x | π, θ) = –397.40



log p(x | π, θ) = –442.89

can get stuck for long time...

Acknowledgements

Slides based on parts of the lecture notes of Erik Sudderth for "Applied Bayesian Nonparametrics" at Brown University