Module 3: Bayesian Nonparametrics

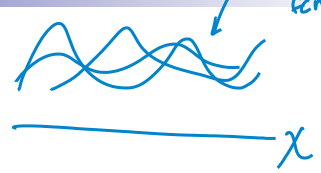
Gaussian Processes for Regression Wrapup

STAT/BIOSTAT 527, University of Washington
Emily Fox
April 24th, 2014

©Emily Fox 2014

Gaussian Processes

GP is a dist on fcns



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

$\Updownarrow$ iff $\forall n$ and any $x_1,\dots,x_n$

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

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

©Emily Fox 2014

GPs for Regression

- Noisy scenario: observe a noisy version of underlying function
$$y = f(x) + \epsilon \quad \epsilon \sim N(0, \sigma_y^2)$$
  - Not required to interpolate, just come "close" to observed data
$$\mathrm{cov}(y|X) = \mathrm{cov}(f) + \mathrm{cov}(\epsilon) = K + \sigma_y^2 I_n \triangleq K_y$$
$[y_1, \ldots, y_n]^T$
- Training data $\mathcal{D} = \{(x_i, y_i), i = 1, \ldots, n\}$
- Test data locations X^* → predict f^*
- Jointly, we have $\begin{pmatrix} y \\ f^* \end{pmatrix} \sim N\left(0, \begin{pmatrix} K_y & K_* \\ K_*^T & K_{**} \end{pmatrix}\right)$

for simplicity

as before

cond. on this

$K(X^*, X^*)$ as before

- Therefore, $p(f^* \mid X^*, X, y) = N(f^* \mid K_*^T K_y^{-1} y, K_{**} - K_*^T K_y^{-1} K_*)$

GPs for Regression

$$p(f^* \mid X^*, X, y) = N(K_*^T K_y^{-1} y, K_{**} - K_*^T K_y^{-1} K_*)$$

- For a single point x^*

$$p(f^* \mid x^*, X, y) = N(k_*^T K_y^{-1} y, k_{**} - k_*^T K_y^{-1} k_*)$$

so

$$\bar{f}^* = k_*^T K_y^{-1} y = \sum_{i=1}^{n} \alpha_i K(x_i, x^*)$$

predictive mean

will see this later

Estimating Hyperparameters

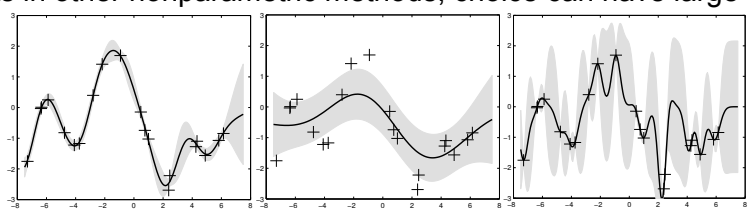
- How should we choose the kernel parameters?
  - Example: squared exponential kernel parameterization

$$\kappa(x, x') = \sigma_f^2 \exp\left(\frac{-1}{2}(x_p - x_q)^T M (x'_p - x'_q)\right) + \sigma_y^2 \delta_{pq}$$

  - Hyperparameters $\theta = \{M, \sigma_f^2, \sigma_y^2\}$
  - As we saw before, can choose

$$M = \ell^{-2} I \quad M = \mathrm{diag}(\ell_1^{-2}, \ldots, \ell_d^{-2}) \quad M = \Lambda\Lambda' + \mathrm{diag}(\ell_1^{-2}, \ldots, \ell_d^{-2}) \ldots$$

- As in other nonparametric methods, choice can have large effect



©Emily Fox 2014

Estimating Hyperparameters

- Options:
  - #1: Define a grid of possible values and use cross validation

    can be slow...
  - #2: Full Bayesian analysis: Place prior on hyperparameters and integrate over these as well in making predictions

    some challenges in practice
  - #3: Maximize the marginal likelihood

$$p(y \mid X, \theta) = \int p(y \mid f, X) p(f \mid X, \theta) df$$

$$\log p(y \mid X, \theta) =$$

©Emily Fox 2014

Estimating Hyperparameters

$$\log p(y \mid X, \theta) = -\frac{1}{2} y^T K_y^{-1} y - \frac{1}{2} \log |K_y| - \frac{n}{2} \log 2\pi$$

- For short length-scale, the fit is good, but *K* is nearly diagonal
- For large length-scale, the fit is bad, but *K* is almost all 1's

- Can show:

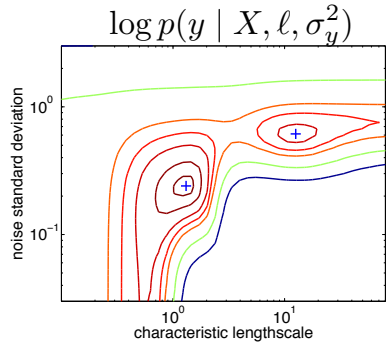
$$\frac{\partial}{\partial \theta_j} \log p(y \mid X, \theta) = \frac{1}{2} y^T K_y^{-1} \frac{\partial K_y}{\partial \theta_j} K_y^{-1} y - \frac{1}{2} \mathrm{tr}\left(K_y^{-1} \frac{\partial K_y}{\partial \theta_j} \right)$$
$$= \frac{1}{2} \mathrm{tr}\left((\alpha\alpha^T - K_y^{-1}) \frac{\partial K_y}{\partial \theta_j} \right)$$

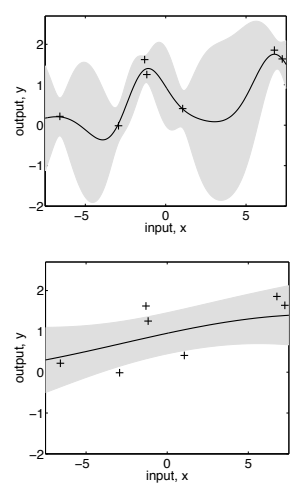
  - Optimize to choose hyperparameters
  - Complexity is
  - Objective is non-convex, so local minima are a problem

©Emily Fox 2014

Example of Estimating Hypers

$$\log p(y \mid X, \ell, \sigma_y^2) \qquad \sigma_f^2 = 1$$





©Emily Fox 2014

Relating GPs to Kernel Methods

- GPs as linear smoothers
  - Recall that the predictive posterior mean of a GP is

$$\bar{f}(x^*) = k_*^T (K + \sigma_y^2 I_n)^{-1} y$$

- In kernel regression, the weight function was derived from a smoothing kernel instead of a Mercer kernel
  - Clear that smoothing kernels have local support
  - Less clear for GPs since the weight function depends on the inverse of *K*
- For some GP kernels, can analytically derive ***equivalent kernel***
  - As with smoothing kernels,
  - Computing a linear combination, but not a convex combination of y_i's
  - Interestingly, the weight function is local even when the GP kernel is not
  - Furthermore, the effective bandwidth of the GP equivalent kernel automatically decreases with *n*, where as in kernel smoothing such tuning must be done by hand

©Emily Fox 2014

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
- Therefore, the effective degrees of freedom is
- Remember that this specifies how "wiggly" the curve is

©Emily Fox 2014

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

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

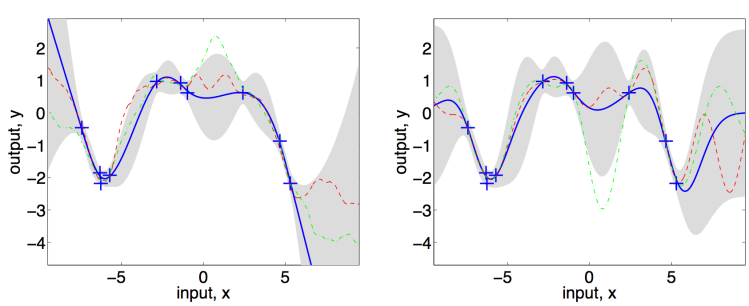
where

- One can show that the MAP estimate of *f*(*x*) is a ***cubic smoothing spline*** when $p(\beta_j) \propto 1$
- Penalty parameter λ is now given by σ_y^2 / σ_f^2

©Emily Fox 2014

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

©Emily Fox 2014

GP Regression Recap

	Linear Basis Expansion	Gaussian Process
Prior	$\beta \sim N(0, \alpha^{-1} I_M)$ $f(x) = \sum_{m=1}^{M} \beta_m \phi_m(x)$	$f \sim \mathrm{GP}(0, \kappa(x, x'))$
Distribution on $x_1, \ldots, x_n$	$f \sim N(0, \alpha^{-1} \Phi \Phi^T)$	$f \sim N(0, K)$
Choices	• *Choose M* • *Choose bases*	• *Choose* $\kappa(x, x')$ • *Choose covariance hyperparameters*

©Emily Fox 2014

GP Regression Recap

Linear Basis Expansion	GP regression	Splines	Kernels
$\{\phi_m(x)\}$	$\kappa(x, x')$		
$\downarrow$	$\downarrow$		
f	f		
$\downarrow$	$\downarrow$		
y	y		

©Emily Fox 2014

Choice of Covariance Function

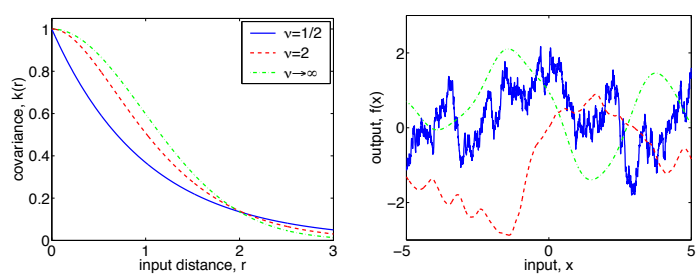
- Definitions
  - ***Stationary*** kernel – only depends on $x - x'$
  - ***Isotropic*** kernel – furthermore only depends on $||x - x'||$

- 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 **

©Emily Fox 2014

Sample Paths using Matern Kernel

- Can produce very rough sample paths

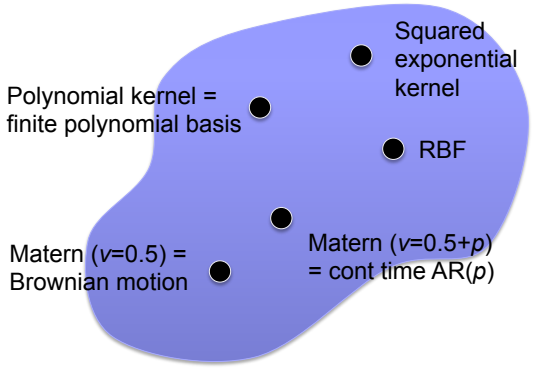


(a) (b)

Figure from Rasmussen and Williams 2006

©Emily Fox 2014

Family of Gaussian Processes



©Emily Fox 2014

Module 3: Bayesian Nonparametrics

Finite Mixture Models

STAT/BIOSTAT 527, University of Washington

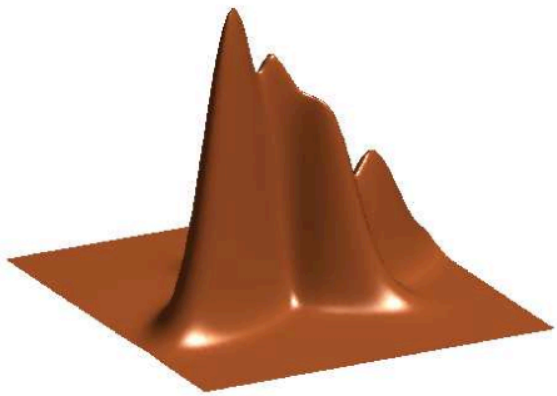
Emily Fox

April 24th, 2014

©Emily Fox 2014

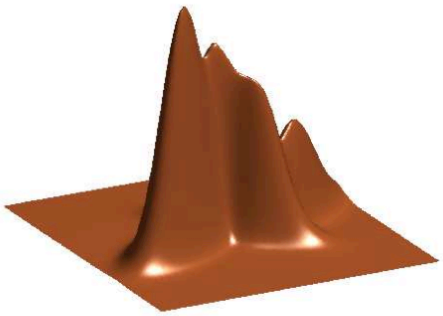
Density Estimation

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

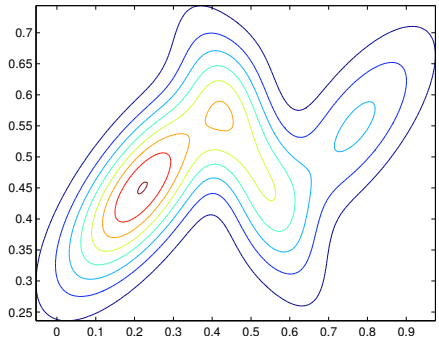


©Emily Fox 2014

Density Estimation



Contour Plot of Joint Density

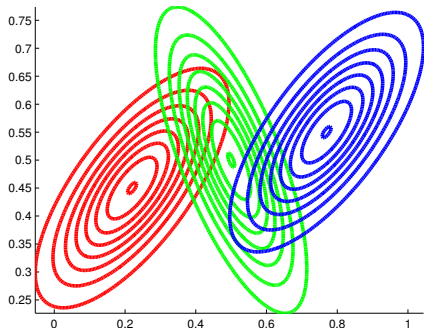


©Emily Fox 2014

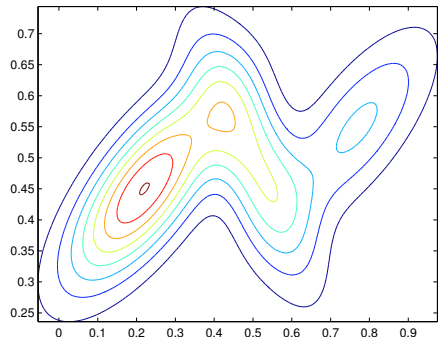
Density as Mixture of Gaussians

- Approximate density with a mixture of Gaussians

Mixture of 3 Gaussians



Contour Plot of Joint Density

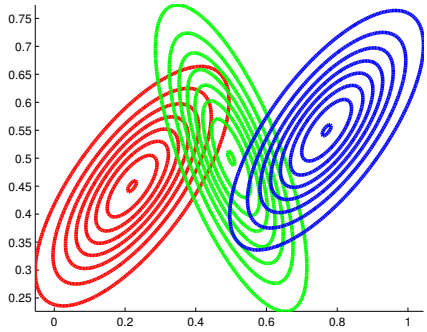


©Emily Fox 2014

Density as Mixture of Gaussians

- Approximate density with a mixture of Gaussians

Mixture of 3 Gaussians



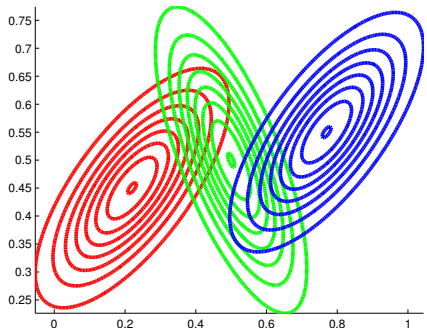
$$p(x_i \mid \pi, \mu, \Sigma) =$$

©Emily Fox 2014

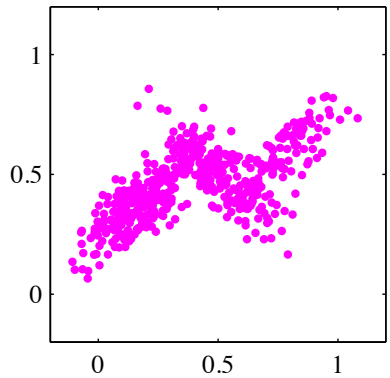
Density as Mixture of Gaussians

- Approximate with density with a mixture of Gaussians

Mixture of 3 Gaussians



Our actual observations

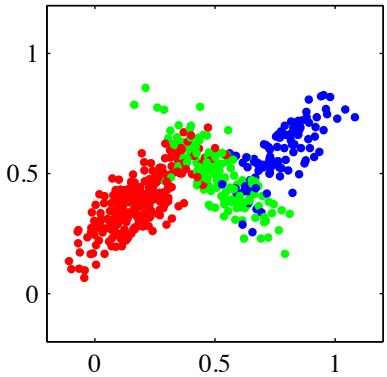


C. Bishop, Pattern Recognition & Machine Learning

©Emily Fox 2014

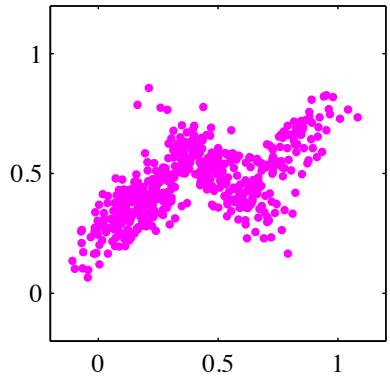
Clustering our Observations

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



Complete data labeled by true cluster assignments

Our actual observations

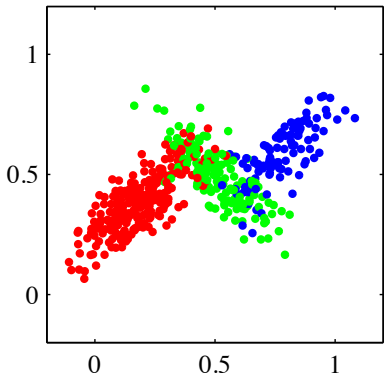


C. Bishop, Pattern Recognition & Machine Learning

©Emily Fox 2014

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
- Then we have

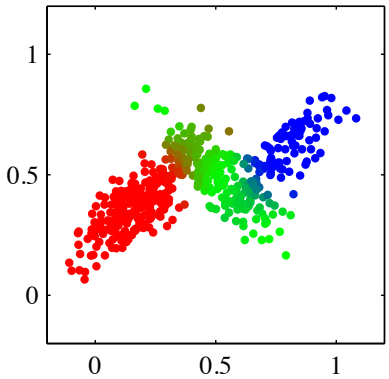
$$p(x_i \mid z_i, \pi, \mu, \Sigma) =$$

C. Bishop, Pattern Recognition & Machine Learning

©Emily Fox 2014

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:

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

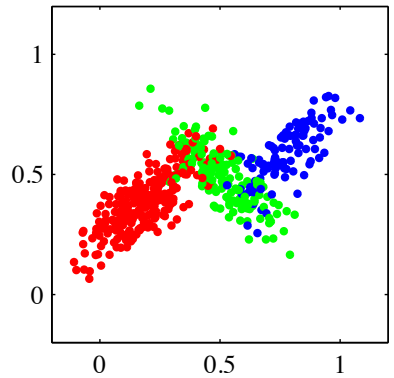
C. Bishop, Pattern Recognition & Machine Learning

©Emily Fox 2014

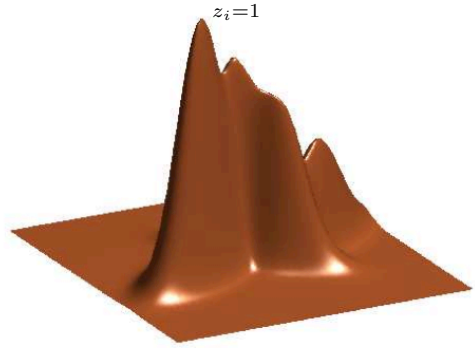
Summary of GMM Concept

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

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



Complete data labeled by true cluster assignments



Surface Plot of Joint Density, Marginalizing Cluster Assignments

©Emily Fox 2014

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 $\pi_k, \quad \sum_{k=1}^{K} \pi_k = 1$

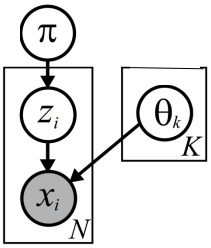
Gaussian mixture marginal and conditional likelihood **:**

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

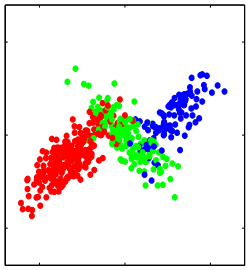
$$p(x_i \mid z_i, \pi, \mu, \Sigma) = \mathcal{N}(x_i \mid \mu_{z_i}, \Sigma_{z_i})$$

©Emily Fox 2014

Generative Model

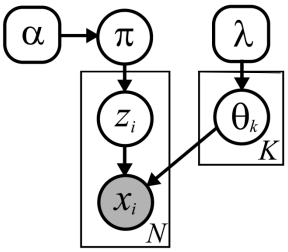


- We can think of *sampling* observations from the model
- For the GMM, define model parameters
  - Cluster means and covariances
  - Cluster weights
- For each observation *i*,
  - Sample a cluster assignment
  - Sample the observation from the selected Gaussian

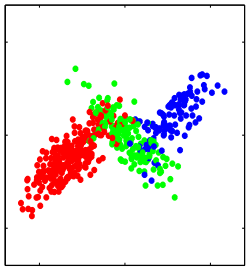


©Emily Fox 2014

A Bayesian GMM



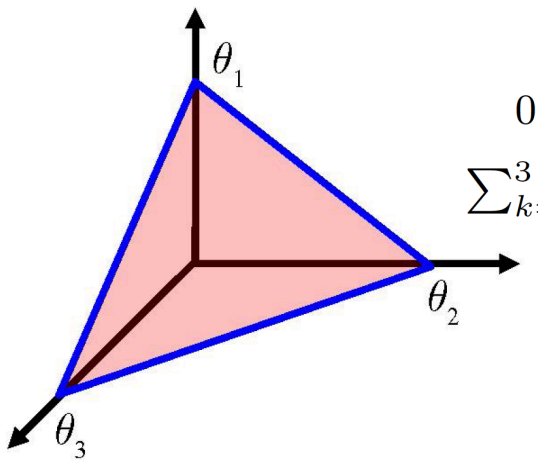
- In a Bayesian approach, we place priors on the model parameters
- Conjugate priors are a computationally convenient choice
- Conjugate prior for θ_k
  - Known variance: Gaussian prior on mean
  - Unknown mean & variance: *normal inverse-Wishart*
- Conjugate prior for π ???



©Emily Fox 2014

The Simplex in 3D

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



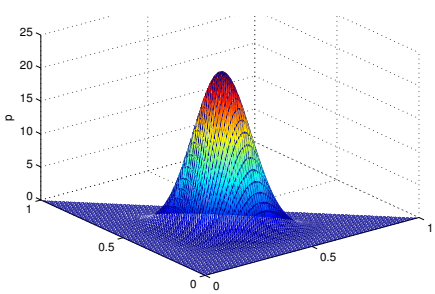
$$0 \leq \theta_k \leq 1$$

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

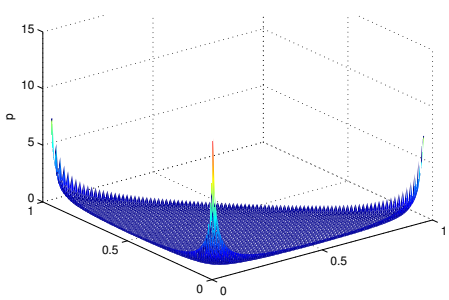
©Emily Fox 2014

Dirichlet Distributions

- The Dirichlet distribution is defined on the simplex



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

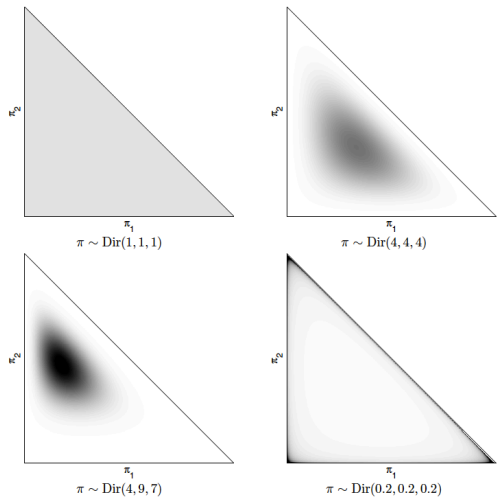


Moments: $\mathbb{E}_\alpha[\pi_k] = \frac{\alpha_k}{\alpha_0}$

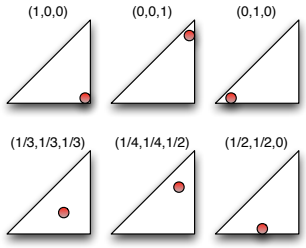
$$\mathrm{Var}_\alpha[\pi_k] = \frac{K-1}{K^2(\alpha_0+1)}$$

©Emily Fox 2014

Dirichlet Probability Densities



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

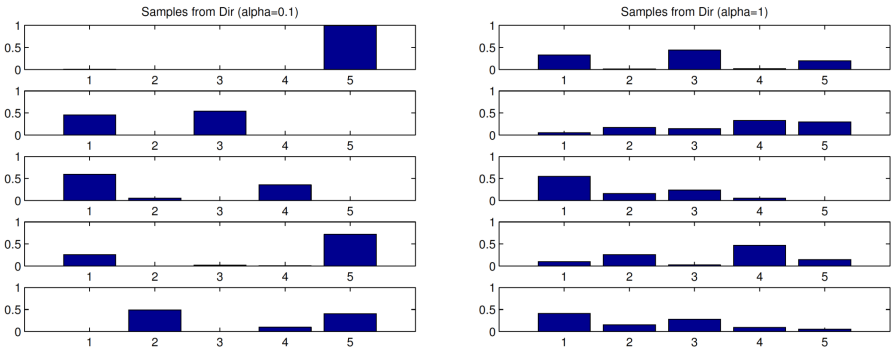


©Emily Fox 2014

Dirichlet Samples

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

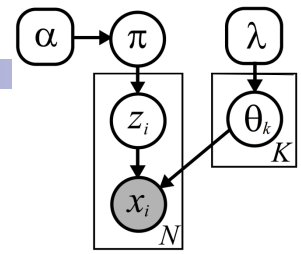
- Samples are ***sparse*** for small values of α_i



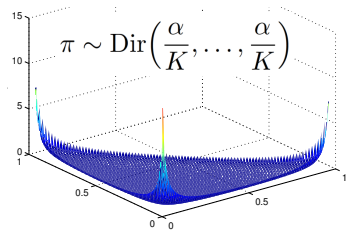
$\mathrm{Dir}(\pi \mid 0.1, 0.1, 0.1, 0.1, 0.1)$ $\mathrm{Dir}(\pi \mid 1.0, 1.0, 1.0, 1.0, 1.0)$

©Emily Fox 2014

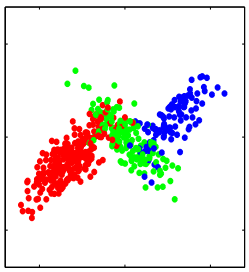
Model Summary



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



  - Normal inverse Wishart prior for θ_k



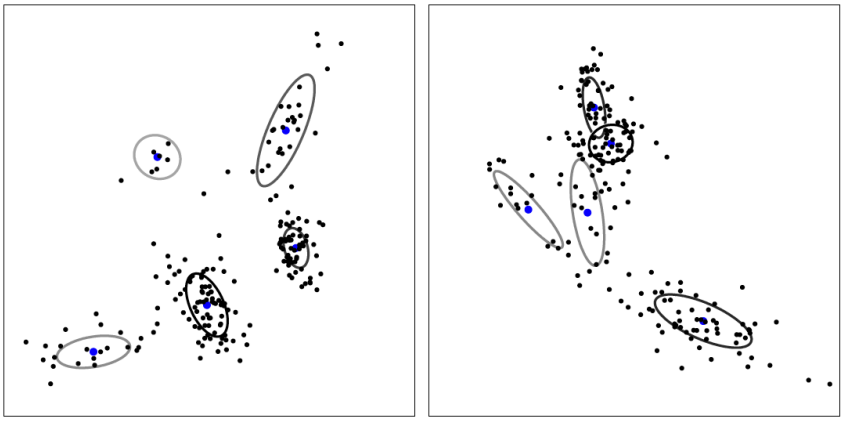
- Sample observations as

$$z_i \sim \pi$$

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

©Emily Fox 2014

Samples Generated from GMM



©Emily Fox 2014

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

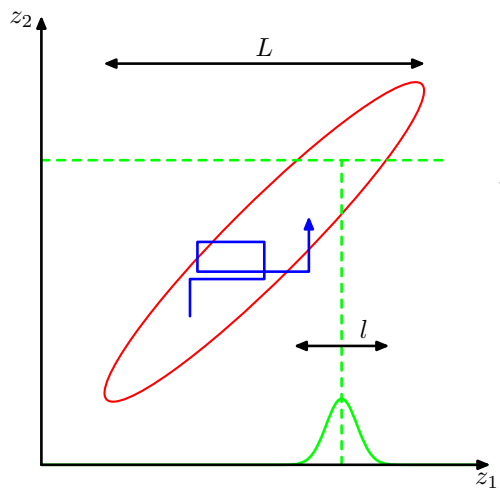
©Emily Fox 2014

Gibb Sampling

- Let *z* indicate the set of ***all variables in the model***: e.g., cluster indicators and parameters
- Want draws:
- Construct Markov chain whose steady state distribution is
- Simplest case:

©Emily Fox 2014

Gibbs Sampler for a 2D Gaussian



General Gibbs Sampler

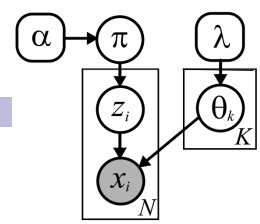
$$z_i^{(t)} \sim p(z_i \mid z_{\setminus i}^{(t-1)}) \qquad i = i(t)$$
$$z_j^{(t)} = z_j^{(t-1)} \qquad j \neq i(t)$$

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

C. Bishop, Pattern Recognition & Machine Learning

©Emily Fox 2014

Example – GMM



- Recall model
  - Observations: $x_1, \ldots, x_N$
  - Cluster indicators: $z_1, \ldots, z_N$
  - Parameters: π, θ_k

$$\pi = [\pi_1, \ldots, \pi_K]$$
$$\theta_k = \{\mu_k, \Sigma_k\}$$

  - 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

©Emily Fox 2014

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

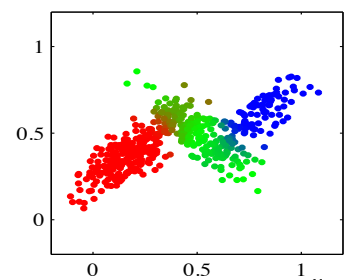
- We have $z_i \sim \pi$

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

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

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

- Sample each cluster indicator as



©Emily Fox 2014

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 $z_i = k$
  - Then,

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

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

©Emily Fox 2014

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

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

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

- 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,

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

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

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

©Emily Fox 2014

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:

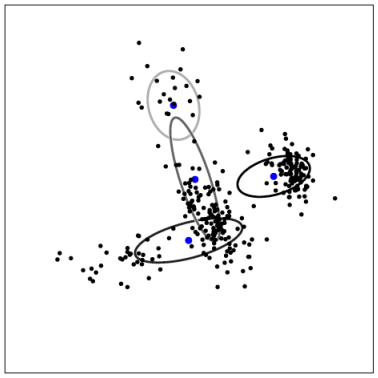
$$\pi^{(t)} \sim \mathrm{Dir}(N_1 + \alpha/K, \ldots, N_K + \alpha/K) \qquad N_k = \sum_{i=1}^N \delta(z_i^{(t)}, k)$$

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

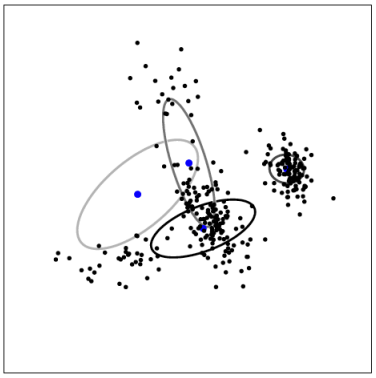
$$\theta_k^{(t)} \sim p(\theta_k \mid \{x_i \mid z_i^{(t)} = k\}, \lambda)$$

©Emily Fox 2014

Standard Sampler: 2 Iterations



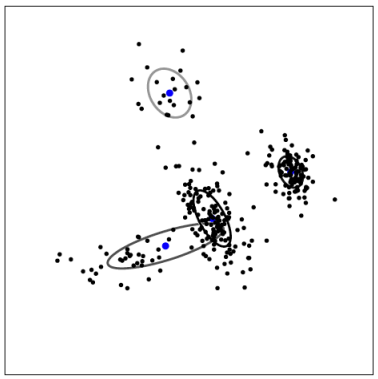
log p(x | π, θ) = –539.17



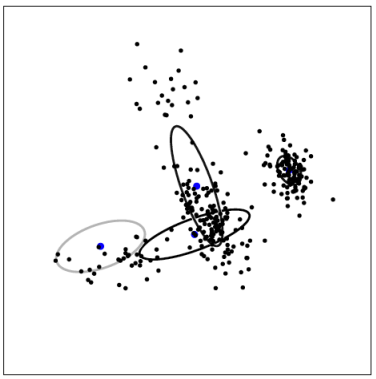
log p(x | π, θ) = –497.77

©Emily Fox 2014

Standard Sampler: 10 Iterations



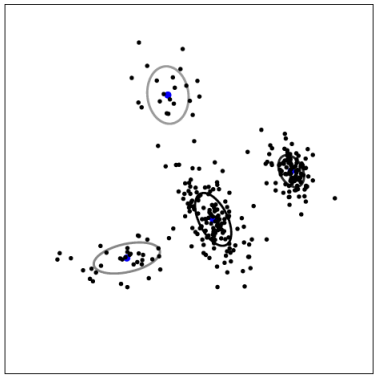
log p(x | π, θ) = –404.18



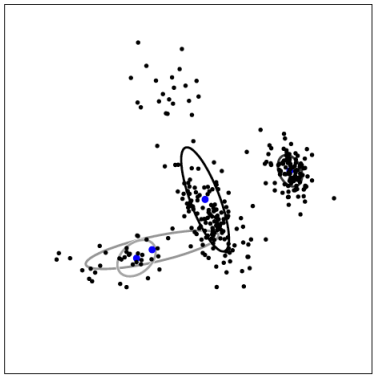
log p(x | π, θ) = –454.15

©Emily Fox 2014

Standard Sampler: 50 Iterations



log p(x | π, θ) = –397.40



log p(x | π, θ) = –442.89

©Emily Fox 2014

Acknowledgements

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

©Emily Fox 2014