**Module 3: Bayesian Nonparametrics**

# Gaussian Processes for Regression Wrapup

STAT/BIOSTAT 527, University of Washington

Emily Fox

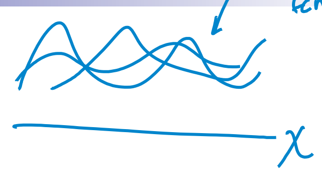
April 24th, 2014





# 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 $x_1, \ldots, x_n$

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

$\begin{bmatrix} \vdots \end{bmatrix} \sim N(\quad)$

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

# 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

$$\text{cov}(y|X) = \text{cov}(f) + \text{cov}(\epsilon) = K + \sigma_y^2 I_n \triangleq K_y$$

$\{y_1, \ldots, y_n\}$

- Training data $\mathcal{D} = \{(x_i, y_i), i = 1, \ldots, n\}$
- Test data locations $X^*$ → predict *f**

for simplicity

as before

- Jointly, we have $\begin{pmatrix} y \\ f^* \end{pmatrix} \sim N\left(0, \begin{pmatrix} K_y & K_* \\ K_*^T & K_{**} \end{pmatrix}\right)$

cond. on this

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

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

closed-form pred dist.



# 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**

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

so

$$\boxed{\bar{f}^* = \underbrace{k_*^T K_y^{-1}}_{\text{"hat matrix"}} y = \sum_{i=1}^{n} \alpha_i K(x_i, x^*)}$$

predictive mean

will see this later

remember for later

# Estimating Hyperparameters

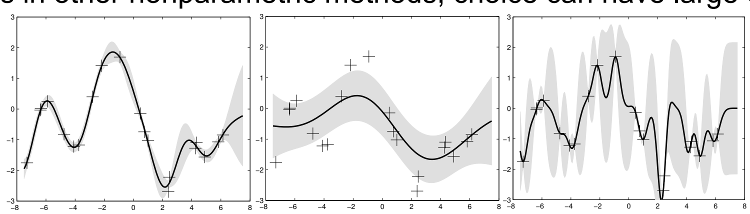
- How should we choose the kernel parameters?
  - Example: squared exponential kernel parameterization

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

  - 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





# 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 ← think of $f(x_1), \ldots, f(x_n)$ as params

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

$\prod_{i=1}^{n} N(y_i \mid f(x_i), \sigma_y^2)$ $\quad N(f \mid 0, K_\theta)$

$= N(y \mid 0, K_y)$

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

# Estimating Hyperparameters

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

fit, complexity, const.

- For short length-scale, the fit is good, but *K* is nearly diagonal
  - ⇒ $\log|K_y|$ large
- For large length-scale, the fit is bad, but *K* is almost all 1's
  - ⇒ $\log|K_y|$ small

- Can show:

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

$$= \frac{1}{2} \mathrm{tr}\left((\alpha\alpha^T - K_y^{-1}) \frac{\partial K_y}{\partial \theta_j}\right)$$

big inverse

as defined before

- Optimize to choose hyperparameters
- Complexity is $O(n^3)$ for $K_y^{-1}$, $O(n^2)$ for gradient / hyper.
- Objective is non-convex, so local minima are a problem



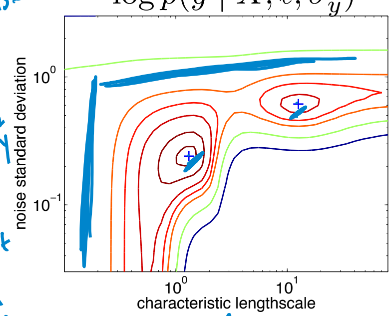
# Example of Estimating Hypers

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

high noise

$\sigma_y^2$

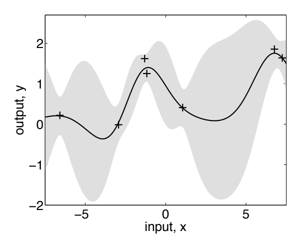
invariant to length scale

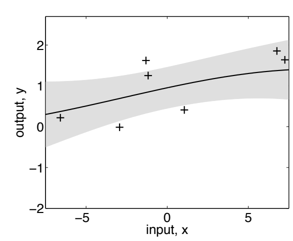



$\ell$

low lengthscale

f is interpolator, so insensitive to $\sigma_y^2$



# Relating GPs to Kernel Methods

- GPs as linear smoothers
  - Recall that the predictive posterior mean of a GP is $[(K+\sigma_y^2 I_n)^{-1} k_*]_i$

$$\bar{f}(x^*) = \underbrace{k_*^T (K + \sigma_y^2 I_n)^{-1}}_{L} y = \sum_i l_i(x^*) y_i$$

- 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, $\sum l_i(x^*) = 1$ but some $l_i(x^*)$ can $< 0$
  - 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



# Effective Degrees of Freedom

- For the training set, the fit is given by

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

- 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

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

can grow w/ n

fcn of how quickly eigvals decay

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

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

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

- One can show that the MAP estimate of *f*(*x*) is a ***cubic smoothing spline*** when $p(\beta_j) \propto 1$ ← don't penalize $0^{th}$ + $1^{st}$ order terms

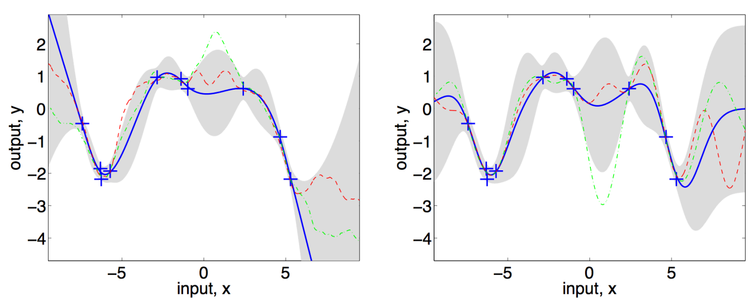
  $\beta_0, \beta_1$

- Penalty parameter λ is now given by $\sigma_y^2/\sigma_f^2$




# 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

special case w/ finite rank kernel

| | Linear Basis Expansion (Bayesian formulation) | Gaussian Process |
|---|---|---|
| **Prior** | $\beta \sim N(0, \alpha^{-1} I_M)$ <br> $\underset{\text{fcn}}{f(x)} = \sum_{m=1}^{M} \beta_m \phi_m(x)$ <br> specify basis | $\underset{\text{fcn}}{f} \sim \mathrm{GP}(0, \kappa(x, x'))$ <br> specify cov. fcn |
| **Distribution on $x_1, \ldots, x_n$** | $\underline{f} \sim N(0, \alpha^{-1} \Phi \Phi^T)$ | $\underline{f} \sim N(0, K)$ <br> $\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix}$ |
| **Choices** | • *Choose M* <br> • *Choose bases* | • *Choose* $\kappa(x, x')$ <br> • *Choose covariance hyperparameters* |



# GP Regression Recap

| Linear Basis Expansion | GP regression | Splines | Kernels |
|---|---|---|---|
| parametric family $\{\phi_m(x)\}$ | parametric family $\kappa(x, x')$ | - example of LBE | - choice of kernel |
| $\downarrow$ | $\downarrow$ | | |
| parametric form $f$ | flexible "nonparam" $f$ | - for smoothing splines, choose $\lambda$ | - choice of bandwidth |
| $\downarrow$ | $\downarrow$ | | |
| $y \sim N(f(x), \sigma^2)$ $y$ | $y \sim N(f(x), \sigma^2)$ $y$ | | |

effective # params can adapt w/ n

# 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 **

# Sample Paths using Matern Kernel

- Can produce very rough sample paths

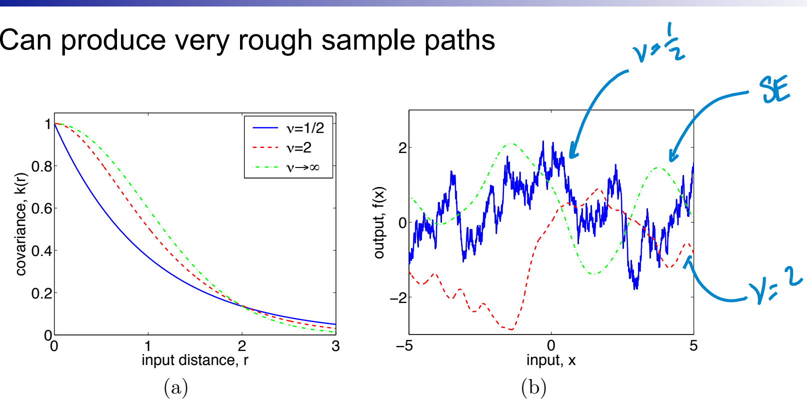
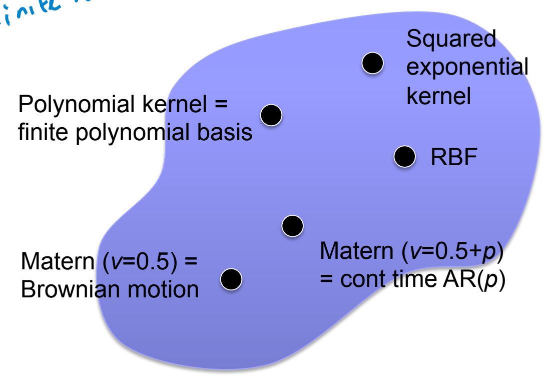



Figure from Rasmussen and Williams 2006

# 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

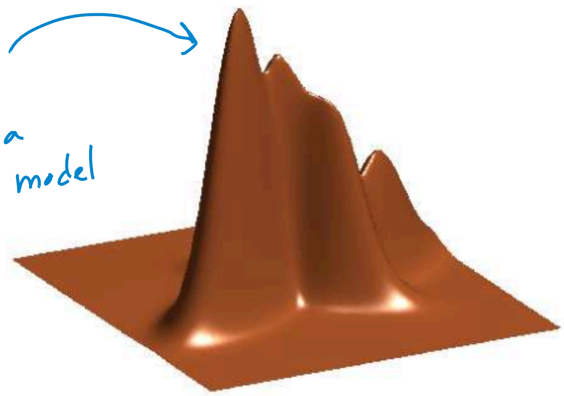
April 24th, 2014

# Density Estimation

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

$x_1,\ldots,x_n \sim p$

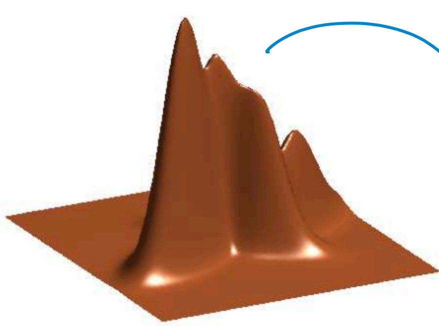
Let's consider a parametric model



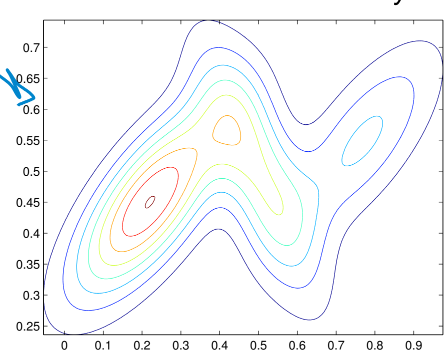



# Density Estimation

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



Contour Plot of Joint Density

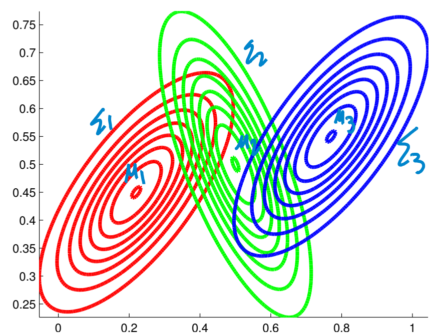



bird's eye view

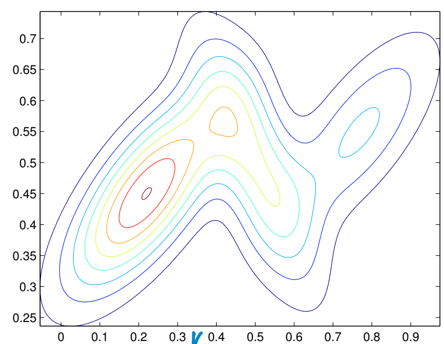
# Density as Mixture of Gaussians

- Approximate density with a mixture of Gaussians

*Mixture of K= 3 Gaussians*




*Contour Plot of Joint Density*



Each Gaussian has weight $\pi_k$ w/ $\sum_{k=1}^{K} \pi_k = 1$

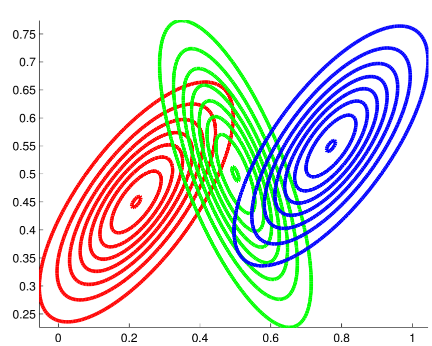
and shape params $\{\mu_k, \Sigma_k\}$




# Density as Mixture of Gaussians

- Approximate density with a mixture of Gaussians

*Mixture of 3 Gaussians*



$[\pi_1, \dots, \pi_K]$ $\{\mu_k, \Sigma_k\}$

p=

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

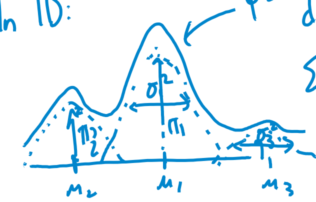
$$\sum_{k=1}^{K} \pi_k N(x_i \mid \mu_k, \Sigma_k)$$

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

In 1D:

p = target density

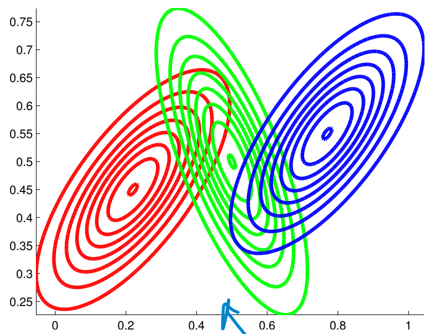
$\sum \pi_k = 1$



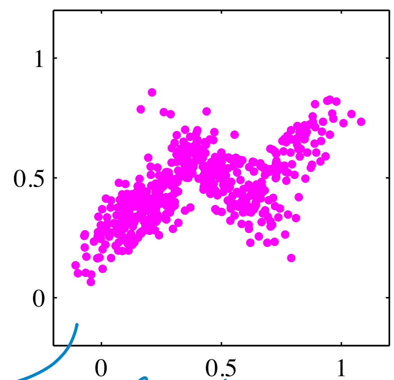
# Density as Mixture of Gaussians

- 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*

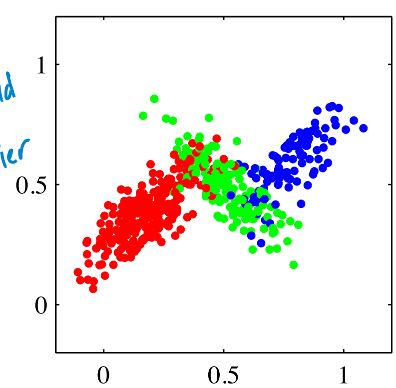




# Clustering our Observations

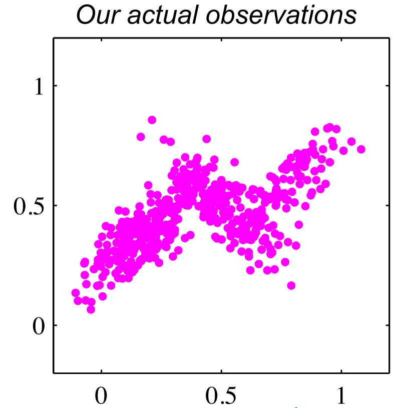
- Imagine we have an assignment of each $x_i$ to a Gaussian

life would be easier



*Complete data labeled by true cluster assignments*

*Our actual observations*

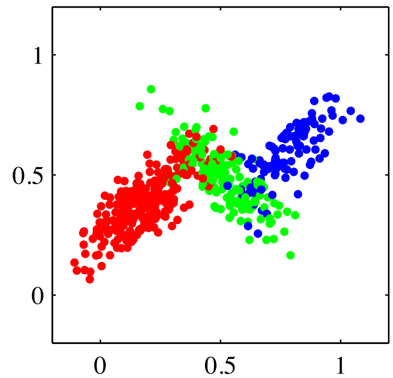


"incomplete data"

*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 $z_i$

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

  $Pr(z_i = k) = \pi_k$

- Then we have

$$p(x_i \mid \underbrace{z_i}_{}^{=k}, \pi, \mu, \Sigma) =$$

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

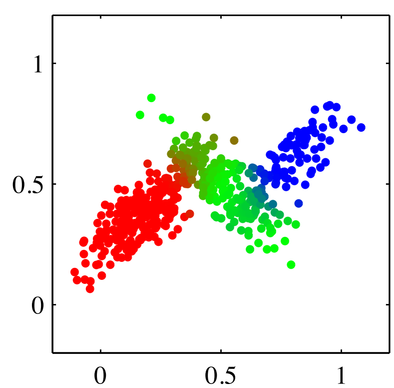
param. est. is easy if we have $\{z_i\}$ $\Rightarrow$ decouples into K Gauss. est

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.

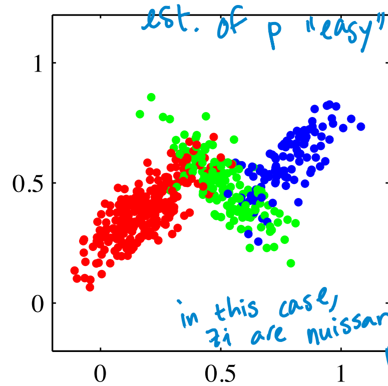
C. Bishop, Pattern Recognition & Machine Learning

# Summary of GMM Concept

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

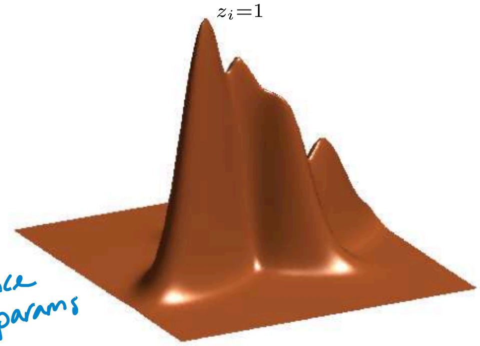
introducing $\{z_i\}$ make est. of p "easy"

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




in this case, $z_i$ are nuissance params

*Complete data labeled by true cluster assignments*



*Surface Plot of Joint Density, Marginalizing Cluster Assignments*




# Summary of GMM Components

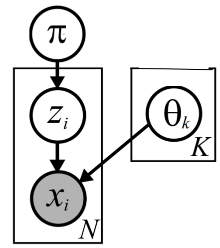
- Observations $\quad x_i \in \mathbb{R}^d, \quad i = 1, 2, \ldots, N$
- Hidden cluster labels $\quad z_i \in \{1, 2, \ldots, K\}, \quad i = 1, 2, \ldots, N$
- Hidden mixture means $\quad \mu_k \in \mathbb{R}^d, \quad k = 1, 2, \ldots, K$
- Hidden mixture covariances $\quad \Sigma_k \in \mathbb{R}^{d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

***Gaussian mixture marginal and conditional likelihood*** :

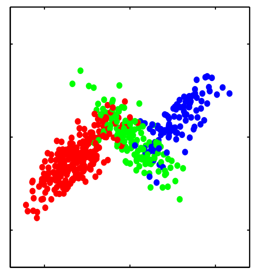
$$p(x_i \mid \pi, \mu, \Sigma) = \sum_{z_i=1}^{K} \pi_{z_i} \mathcal{N}(x_i \mid \mu_{z_i}, \Sigma_{z_i})$$

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

# Generative Model

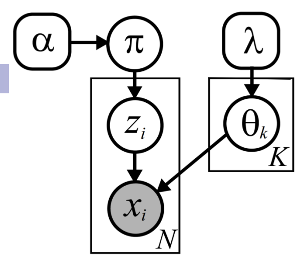



- We can think of *sampling* observations from the model
- For the GMM, define model parameters
  - Cluster means and covariances $\{\mu_k, \Sigma_k\}$
  - Cluster weights $\pi = [\pi_1, \dots, \pi_K]$
- For each observation *i*,
  - Sample a cluster assignment
    $z_i \sim \pi$
    us china sweden
  - Sample the observation from the selected Gaussian
    $x_i \mid z_i \sim N(x_i \mid \mu_{z_i}, \Sigma_{z_i})$

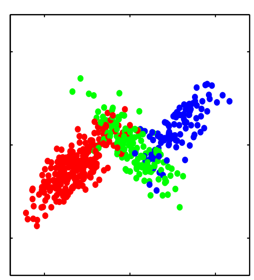





# A Bayesian GMM

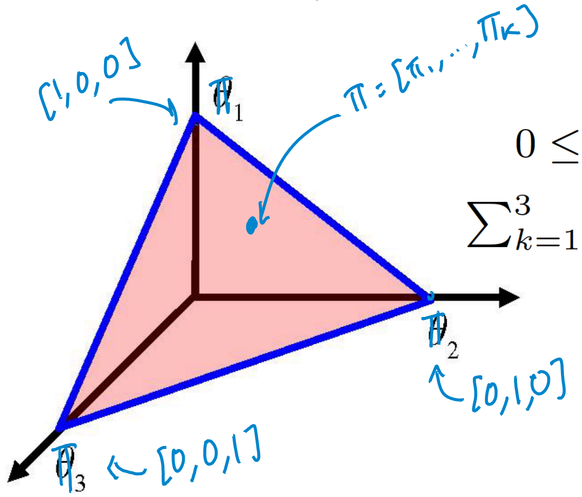



- In a Bayesian approach, we place priors on the model parameters
- Conjugate priors are a computationally convenient choice
- Conjugate prior for $\theta_k = \{\mu_k, \Sigma_k\}$
  - Known variance: Gaussian prior on mean
  - Unknown mean & variance: *normal inverse-Wishart* (NIW)
- Conjugate prior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



# The Simplex in 3D

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




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

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

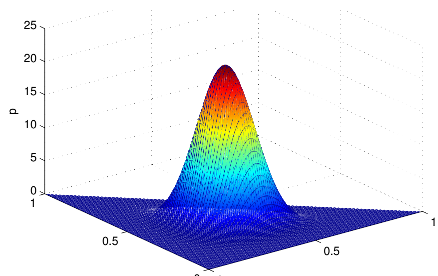


# Dirichlet Distributions

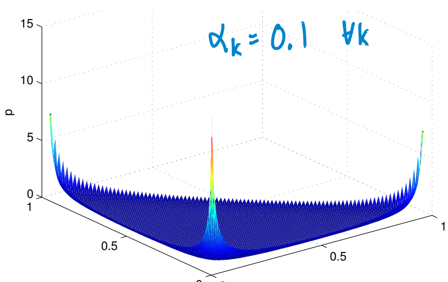
- The Dirichlet distribution is defined on the simplex

$\alpha_k = 10 \ \forall k$

$\pi \sim Dir(\alpha_1, \ldots, \alpha_K)$
$\Rightarrow \sum \pi_k = 1$




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



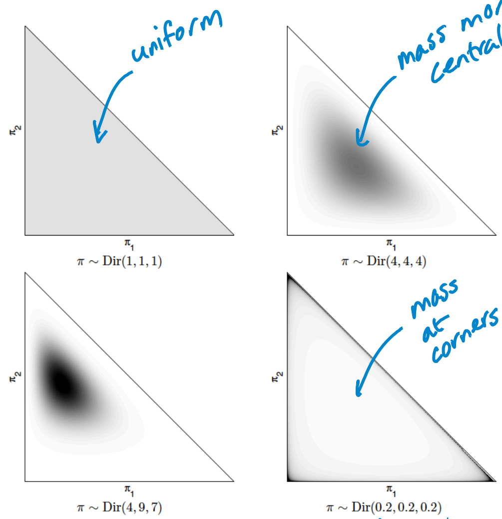

*Moments:* $\mathbb{E}_\alpha[\pi_k] = \dfrac{\alpha_k}{\alpha_0}$

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

# Dirichlet Probability Densities

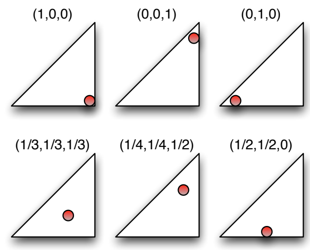
uniform

mass more centralized




mass at corners

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



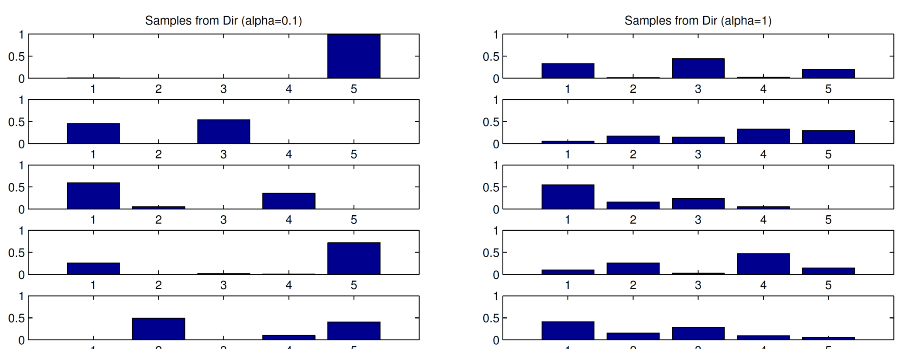

draws from Dir



# Dirichlet Samples

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

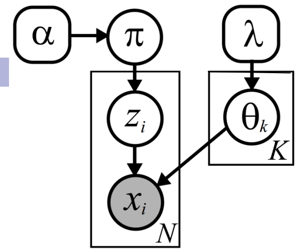
- Samples are ***sparse*** for small values of $\alpha_i$



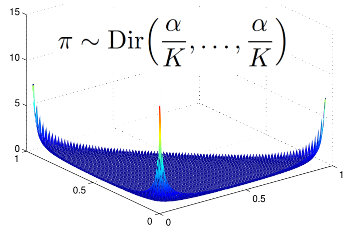

$\mathrm{Dir}(\pi \mid 0.1, 0.1, 0.1, 0.1, 0.1)$ $\mathrm{Dir}(\pi \mid 1.0, 1.0, 1.0, 1.0, 1.0)$

puts mass at corners uniform

# Model Summary




- Prior on model parameters
  - E.g., symmetric Dirichlet for $\pi$

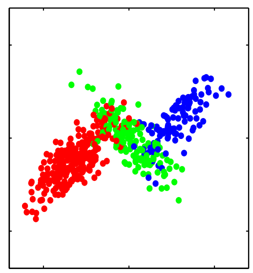



  - Normal inverse Wishart prior for $\theta_k$

- Sample observations as

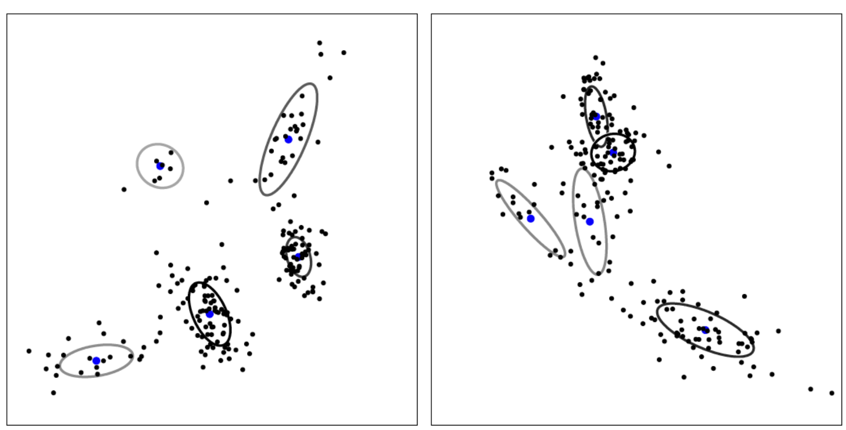
$$z_i \sim \pi$$

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





# Samples Generated from GMM



# Acknowledgements

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