Lecture 19: Mixture models, EM algorithm

CS 475: Machine Learning

Raman Arora

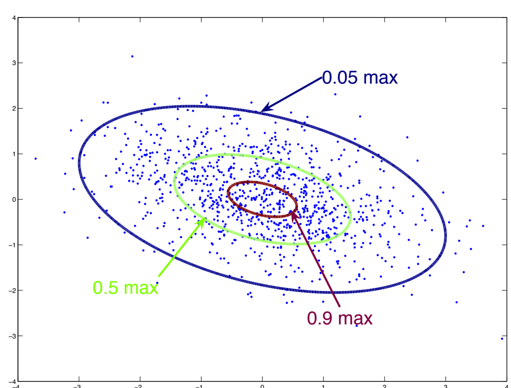
April 7, 2017

Slides credit: Greg Shakhnarovich

Review: multivariate Gaussians

$$\mathcal{N}(\mathbf{x};\, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \;=\; \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

- If eigenvalues are all the same: falls off exponentially as a function of (squared) Euclidean distance to the mean $\|\mathbf{x}-\boldsymbol{\mu}\|^2$;
- the *covariance matrix* $\boldsymbol{\Sigma}$ determines the shape of the density;



- Determinant $|\boldsymbol{\Sigma}|$ measures the "spread" (analogous to σ^2).
- $\mathcal{N}$ is the joint density of coordinates $x_1, \ldots, x_d$.

Review: Gaussian likelihood

$$\log \mathcal{N}(\mathbf{x};\, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \;=\; -\frac{d}{2}\log 2\pi \;-\; \frac{1}{2}\log|\boldsymbol{\Sigma}| \;-\; \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})$$

- Maximum likelihood for the mean:

$$\widehat{\boldsymbol{\mu}}_{ML} \;=\; \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$$

- Maximum likelihood for the covariance:

$$\widehat{\boldsymbol{\Sigma}}_{ML} \;=\; \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i-\boldsymbol{\mu})(\mathbf{x}_i-\boldsymbol{\mu})^{\mathsf{T}}.$$

Review: generative models, C classes

- Construct for each class c

$$\delta_c(\mathbf{x}) \;\triangleq\; \log p\left(\mathbf{x}\,|\,y=c\right) \;+\; \log p(y=c)$$

  based on our per-class (class-conditional) model $p\left(\mathbf{x}\,|\,y=c\right)$
- Generative classifier:

$$h^*(\mathbf{x}) = \underset{c}{\operatorname{argmax}}\,\delta_c(\mathbf{x}).$$

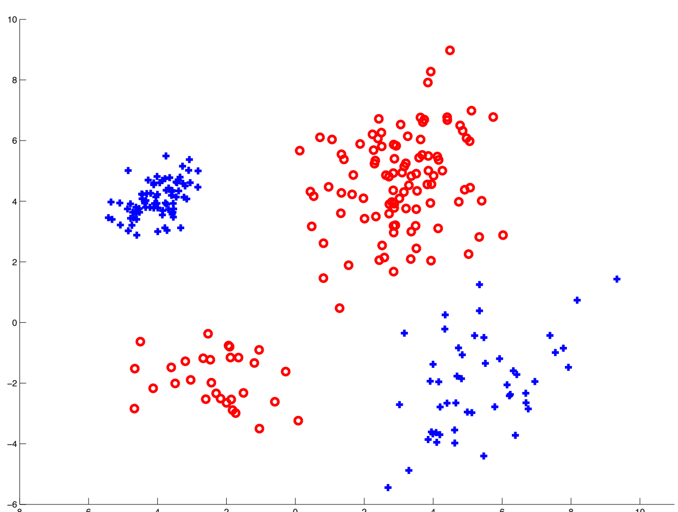
- If assume equal priors $p(y=c) = 1/C$, then $h^*(\mathbf{x}) = \operatorname{argmax}_c \log p\left(\mathbf{x}\,|\,y=c\right).$

Gaussian models – discriminant analysis

- Decision boundary between two classes: $\delta_q(\mathbf{x}) - \delta_k(\mathbf{x}) = 0$.
- If each class is a Gaussian $(\boldsymbol{\mu}_c, \boldsymbol{\Sigma})$:
  a linear discriminant (linear boundary)
- If allow per-class covariances, each class $(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$:
  quadratic decision boundary
- Compare to logistic regression (softmax), SVM: same form of the final classifier, but learned differently
  logistic regression: model $p\left(y \,|\, \mathbf{x}\right)$, log-loss;
  SVM: no probabilistic model, hinge loss;
  Gaussian discriminant analysis: model $p\left(\mathbf{x} \,|\, y\right), p(y)$; maximize per-class log likelihood of $\mathbf{x}$

Mixture models

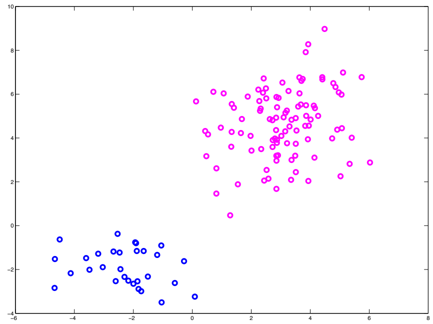
- So far, we have assumed that each class has a single coherent model.



- What if the examples (within the same class) are from a number of distinct “types”?

Mixture of Gaussians

- k underlying types (components);
- Each component is Gaussian;
- y_i is the identity of the component "responsible" for $\mathbf{x}_i$;
- y_i is a *hidden* (*latent*) variable: never observed.
- A *Gaussian mixture model*:



$$p(\mathbf{x}; \boldsymbol{\pi}, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \sum_{c=1}^{k} \pi_c \cdot \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c).$$

- π_cs are the *mixing probabilities*, $\pi_c = p(y = c)$

Likelihood of a mixture model

- Idea: estimate set of parameters that maximize likelihood given the observed data.
- The log-likelihood of $\boldsymbol{\pi}, \boldsymbol{\theta}$:

$$\log p(X; \boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots) = \sum_{i=1}^{N} \log \sum_{c=1}^{k} \pi_c \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c).$$

- No closed-form solution because of the sum inside log.
  - We need to take into account all possible components that could have generated $\mathbf{x}_i$.

Mixture density estimation

- Suppose that we do observe $y_i \in \{1, \ldots, k\}$ for each $i = 1, \ldots, N$.
- Let us introduce a set of binary *indicator variables* $\mathbf{z}_i = [z_{i1}, \ldots, z_{ik}]$ where

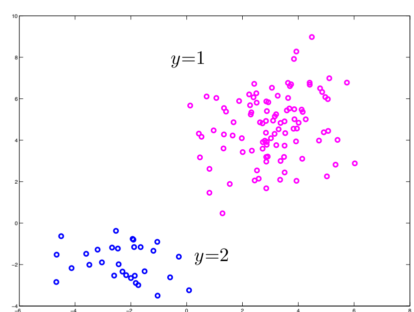
$$z_{ic} = 1 \;=\; \begin{cases} 1 & \text{if } y_i = c, \\ 0 & \text{otherwise.} \end{cases}$$

- The count of examples from c-th component:

$$N_c \;=\; \sum_{i=1}^{N} z_{ic}.$$

Mixture density estimation: known labels

- If we know $\mathbf{z}_i$, the ML estimates of the Gaussian components, just like in class-conditional model, are



$$\widehat{\pi}_c \;=\; \frac{N_c}{N},$$

$$\widehat{\boldsymbol{\mu}}_c \;=\; \frac{1}{N_c}\sum_{i=1}^{N} z_{ic}\mathbf{x}_i,$$

$$\widehat{\boldsymbol{\Sigma}_c} \;=\; \frac{1}{N_c}\sum_{i=1}^{N} z_{ic}(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_c)(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_c)^T.$$

Credit assignment

- When we don't know $\mathbf{z}_i$, we face a *credit assignment* problem: which component is responsible for $\mathbf{x}_i$?
- Suppose for a moment that we do know component parameters $\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ and mixing probabilities $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_k]$.
- Then, the posterior of each label using Bayes rule:

$$\gamma_{ic} \;=\; \widehat{p}(y = c \,|\, \mathbf{x};\, \boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots) \;=\; \frac{\pi_c \cdot p(\mathbf{x};\, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)}{\sum_{l=1}^{k} \pi_l \cdot p(\mathbf{x};\, \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}$$

- We will call γ_{ic} the *responsibility* of the c-th component for $\mathbf{x}$.
  - Note: $\sum_{c=1}^{k} \gamma_{ic} = 1$ for each i.

Expected likelihood

- The "complete data" likelihood (when $\mathbf{z}$ are known):

$$p(X, Z;\, \boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots) \;\propto\; \prod_{i=1}^{N} \prod_{c=1}^{k} \left(\pi_c \mathcal{N}\left(\mathbf{x}_i;\, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c\right)\right)^{z_{ic}}.$$

  and the log:

$$\log p(X, Z;\, \boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots) = \text{const} + \sum_{i=1}^{N} \sum_{c=1}^{k} z_{ic} \left(\log \pi_c \;+\; \log \mathcal{N}\left(\mathbf{x}_i;\, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c\right)\right).$$

- We can't compute it, but can take the *expectation* w.r.t. the posterior of $\mathbf{z}$, which is just γ_{ic}:

$$E_{z_{ic} \sim \gamma_{ic}} \left[\log p(\mathbf{x}_i, z_{ic};\, \boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots)\right].$$

Expected likelihood

$$\log p(X, Z; \boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots) = \text{const} + \sum_{i=1}^{N} \sum_{c=1}^{k} z_{ic} \left(\log \pi_c + \log \mathcal{N}\left(\mathbf{x}_i; \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c\right)\right).$$

- Expectation of z_{ic}:

$$E_{z_{ic} \sim \gamma_{ic}} \left[z_{ic} \right] = \sum_{z \in 0,1} z \cdot \gamma_{ic}^{z} = \gamma_{ic}.$$

- The expected likelihood of the data:

$$\begin{aligned} E_{z_{ic} \sim \gamma_{ic}} \left[\log p(X, Z; \boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots) \right] &= \text{const} \\ &+ \sum_{i=1}^{N} \sum_{c=1}^{k} \gamma_{ic} \left(\log \pi_c + \log \mathcal{N}\left(\mathbf{x}_i; \mu_c, \boldsymbol{\Sigma}_c\right)\right). \end{aligned}$$

Expectation maximization

$$E_{z_{ic} \sim \gamma_{ic}} \left[\log p(X_N, Z_N; \boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots) \right] = \sum_{i=1}^{N} \sum_{c=1}^{k} \gamma_{ic} \left(\log \pi_c + \log \mathcal{N}\left(\mathbf{x}_i; \mu_c, \boldsymbol{\Sigma}_c\right)\right)$$

- We can find $\boldsymbol{\pi}$, $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\Sigma}_k$ that maximize this *expected* likelihood – by setting derivatives to zero and, for $\boldsymbol{\pi}$, using Lagrange multipliers to enforce $\sum_c \pi_c = 1$.

$$\begin{aligned} \hat{\pi}_c &= \frac{1}{N} \sum_{i=1}^{N} \gamma_{ic}, \\ \hat{\boldsymbol{\mu}}_c &= \frac{1}{\sum_{i=1}^{N} \gamma_{ic}} \sum_{i=1}^{N} \gamma_{ic} \mathbf{x}_i, \\ \widehat{\boldsymbol{\Sigma}}_c &= \frac{1}{\sum_{i=1}^{N} \gamma_{ic}} \sum_{i=1}^{N} \gamma_{ic} (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_c)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_c)^T. \end{aligned}$$

Summary so far

- If we know the **parameters** and **indicators** (assignments) we are done.
- If we know the **indicators** but not the parameters, we can do ML estimation of the parameters – and we are done.
- If we know the **parameters** but not the indicators, we can compute the posteriors of indicators;
  - With known posteriors, we can estimate parameters that maximize the *expected* likelihood – and then we are done.
- But in reality we know neither the parameters nor the indicators.

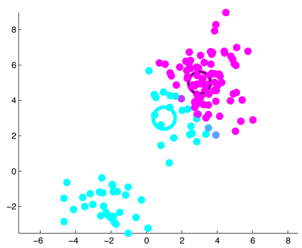
The EM algorithm

- Start with a guess of $\boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots$.
  - Typically, random Gaussians and $\pi_c = 1/k$.
- Iterate between:
  - E-step: Compute values of expected assignments, i.e. calculate γ_{ic}, using current estimates of $\boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots$.
  - M-step: Maximize the *expected* likelihood, under current γ_{ic}.
- Repeat until convergence.

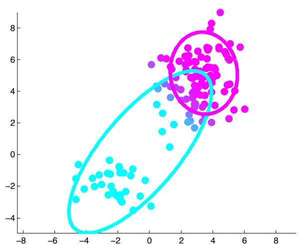
EM for Gaussian mixture: an example

- Colors represent γ_{ic} after the E-step.

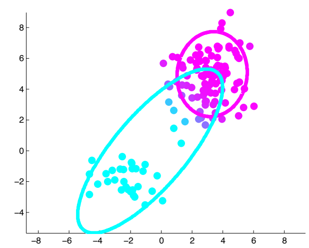
1st iteration



2nd iteration

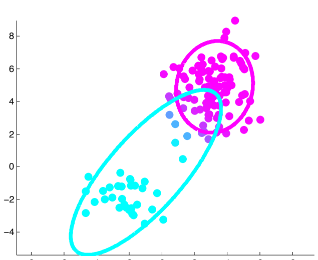


3rd iteration

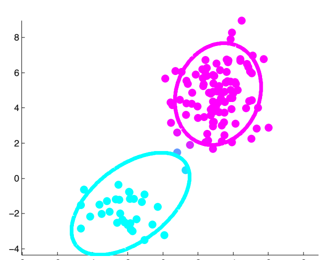


EM for Gaussian mixture: an example

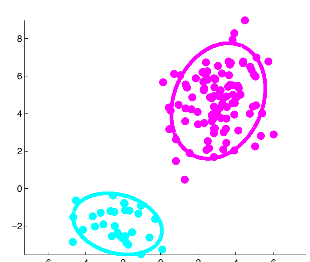
4th iteration

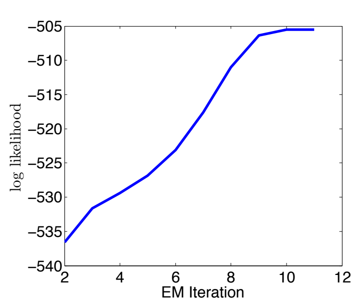


7th iteration



10th iteration





Log-likelihood progress with iterations

Generic EM for mixture models

- General mixture models: $p(\mathbf{x}) = \sum_{c=1}^{k} \pi_c \, p(\mathbf{x};\, \boldsymbol{\theta}_c)$
- Initialize $\boldsymbol{\pi}$, $\boldsymbol{\theta}^{old}$, and iterate until convergence:

E-step: compute responsibilities

$$\gamma_{ic} \;=\; \frac{\pi_c^{old}\, p(\mathbf{x}_i; \boldsymbol{\theta}_c^{old})}{\sum_{l=1}^{k} \pi_l^{old}\, p(\mathbf{x}_i; \boldsymbol{\theta}_l^{old})}.$$

M-step: re-estimate mixture parameters:

$$\boldsymbol{\pi}^{new},\, \boldsymbol{\theta}^{new} \;=\; \operatorname*{argmax}_{\boldsymbol{\theta},\boldsymbol{\pi}} \sum_{i=1}^{N} \sum_{c=1}^{k} \gamma_{ic} \left(\log \pi_c \,+\, \log p(\mathbf{x}_i;\, \boldsymbol{\theta}_c)\right).$$

The EM algorithm in general

- Observed data X, hidden variables Z.
  - E.g., *missing data*.
- Initialize θ^{old}, and iterate until convergence:

E-step: Compute the expected complete data log-likelihood as a function of θ.

$$Q\left(\theta;\, \theta^{old}\right) \;=\; E_{p(Z \,|\, X, \theta^{old})}\left[\log p(X, Z;\, \theta) \,|\, X,\, \theta^{old}\right]$$

M-step: Compute

$$\theta^{new} \;=\; \operatorname*{argmax}_{\theta} Q\left(\theta;\, \theta^{old}\right).$$

Why does EM work?

- Ultimately, we want to maximize likelihood of the *observed* data

$$\theta^* = \operatorname*{argmax}_\theta \log p(X;\theta).$$

- Let $\log p^{(t)}$ be $\log p(X;\theta^{new})$ after t iterations.
- Can show:

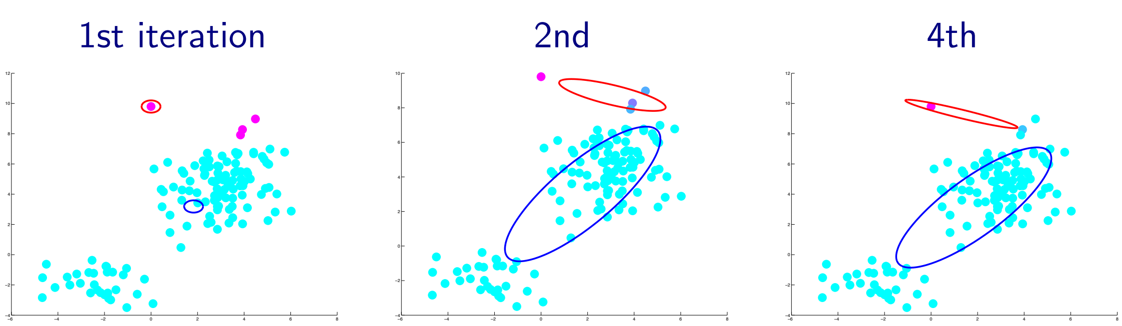
$$\log p^{(0)} \leq \log p^{(1)} \leq \ldots \leq \log p^{(t)} \ldots$$

- Caveat: log-sum may overflow; to prevent,

$$a = \log[\exp(a)] = \log[\exp(a+B)] - B$$

EM and overfitting

- We can be very unlucky with the initial guess.



- The problem:

$$\lim_{\sigma^2 \to 0} \mathcal{N}\left(\mathbf{x};\, \mu = \mathbf{x}, \mathbf{\Sigma} = \sigma^2 \mathbf{I}\right) = \infty.$$

Regularized EM

- Impose a prior on θ.
- Instead of maximizing the likelihood in the M-step, maximize the posterior:

$$\theta^{new} = \operatorname*{argmax}_{\theta} \left\{ E_{p(Z|X;\theta)} \left[\log p(X, Z; \theta) | X; \theta^{old} \right] + \log p(\theta) \right\}.$$

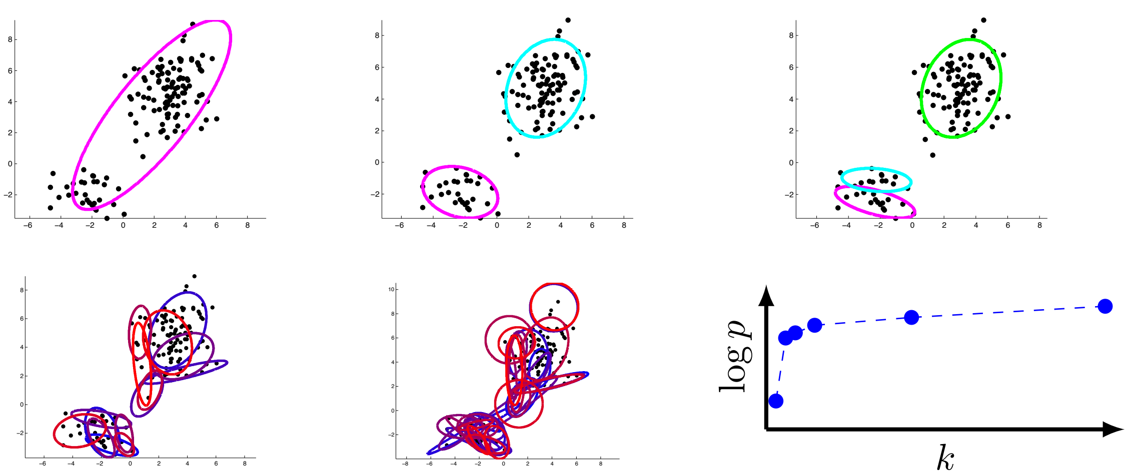
- A common prior on a covariance matrix: the *Wishart* distribution

$$p(\mathbf{\Sigma}; \mathbf{S}, n) \propto \frac{1}{|\mathbf{\Sigma}|^{n/2}} \exp\left(-\frac{1}{2} \operatorname{Tr}\left(\mathbf{\Sigma}^{-1} \mathbf{S} \right) \right)$$

- Intuition: $\mathbf{S}$ is the covariance of n "hallucinated" observations.

Model selection: setting k

- So far we have assumed known k.
- Idea: select k that maximizes the likelihood.



Overfitting mixture models

- Imagine placing a separate, very narow Gaussian component on every training example.
- This solution yields infinite log-likelihood!
- Solution 2: validate on a heldout data set
- Solution 1: penalize model complexity directly, e.g., the Bayesian Information Criterion
- For a model class $\mathcal{M}$ with parameters $\theta_{\mathcal{M}}$, we find ML (or MAP) estimates of the parameters on $X = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$:

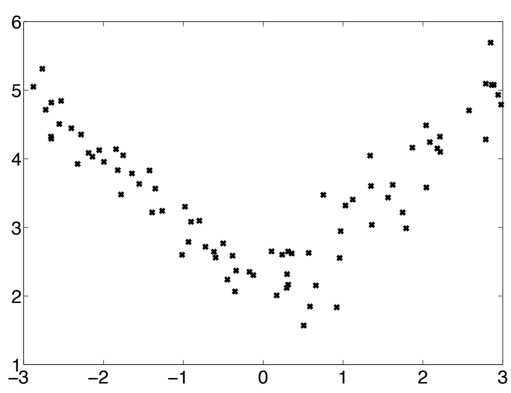
$$L^*(\mathcal{M}) \triangleq \max_{\theta_{\mathcal{M}}} \log p(X|\mathcal{M}; \theta_{\mathcal{M}}).$$

  e.g., $\mathcal{M} = \{$mixtures of 5 Gaussians$\}$
- The BIC score for the model $\mathcal{M}$ on data X:

$$BIC(\mathcal{M}) = L^*(\mathcal{M}) - \frac{1}{2}|\theta_{\mathcal{M}}| \log N.$$

Mixture model for regression

- Example:



- We can represent this as a mixture of two regression models
  - Two *experts*;
  - Need to switch between them according to $\mathbf{x}$.

Mixture of experts model

- Expert j holds a parametric model $p\left(y \,|\, \mathbf{x}; \theta_j\right)$, e.g.,

$$\theta_j = \{\mathbf{w}_j, \sigma_j^2\},$$

$$p\left(y \,|\, \mathbf{x}; \theta_j\right) = \mathcal{N}\left(y;\, \mathbf{w}_j^T \mathbf{x},\, \sigma_j^2\right)$$

- The distribution of y is a *conditional mixture* model:

$$p\left(y \,|\, \mathbf{x};\, \theta\right) = \sum_{j=1}^{c} p\left(j \,|\, \mathbf{x}\right) p\left(y \,|\, \mathbf{x}; \theta_j\right).$$

Gating network

$$p\left(y \,|\, \mathbf{x};\, \theta\right) = \sum_{j=1}^{c} p\left(j \,|\, \mathbf{x}\right) p\left(y \,|\, \mathbf{x}; \theta_j\right)$$

- A *gating network* specifies the conditional distribution $p\left(j \,|\, \mathbf{x};\, \eta\right)$
- Common choice for gaiting: softmax, $\eta = \{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$

$$p\left(j \,|\, \mathbf{x}; \eta\right) = \frac{e^{\mathbf{v}_j^T \mathbf{x}}}{\sum_{t=1}^{k} e^{\mathbf{v}_t^T \mathbf{x}}}.$$

- Can think of it as classification into which expert should be responsible for an example
- Responsibilities:

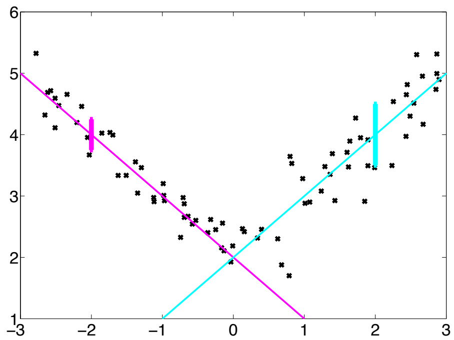
$$\gamma_{ij} = p\left(j \,|\, \mathbf{x}_i, y_i; \theta, \eta\right) = \frac{p\left(j \,|\, \mathbf{x}_i; \eta\right) p\left(y_i \,|\, \mathbf{x}_i; \theta_j\right)}{\sum_{c=1}^{k} p\left(c \,|\, \mathbf{x}_i; \eta\right) p\left(y_i \,|\, \mathbf{x}_i; \theta_c\right)}$$

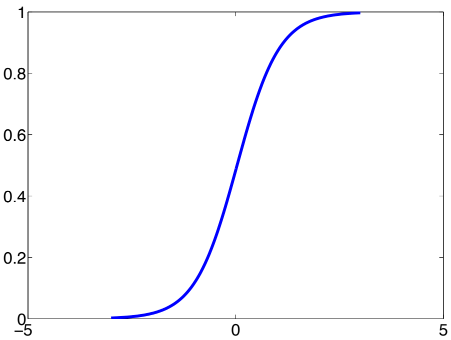
Conditional mixtures

- Parametrization
  - Regression models $p\,(y\,|\,\mathbf{x};\theta_j)$

    e.g., linear regressors, $\theta_j \;=\; \{\mathbf{w}_j,\sigma_j^2\}$.
  - Gating network $p\,(j\,|\,\mathbf{x};\eta)$

    e.g., logistic regression, $\eta \;=\; \{\mathbf{v}\}$





EM for mixtures of experts

Initialize random θ_j, σ_j^2, η.

E-step Compute responsibilities $\gamma_{ij} = p\left(j\,|\,\mathbf{x}_i, y_i;\theta^{old},\eta^{old}\right)$

M-step Separately:

- For each expert j estimate

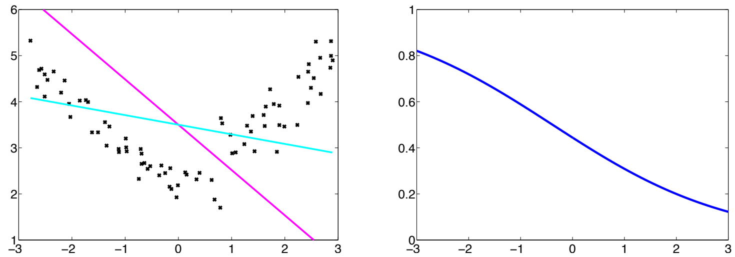
$$\theta_j^{new} \;=\; \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{N} \gamma_{ij} \log p\,(y_i\,|\,\mathbf{x}_i;\theta)$$

- Estimate the gating network:

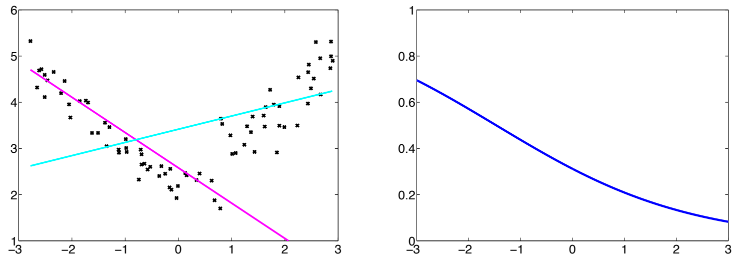
$$\eta^{new} \;=\; \underset{\eta}{\operatorname{argmax}} \sum_{i=1}^{N}\sum_{j=1}^{k} \gamma_{ij} \log p\,(j\,|\,\mathbf{x}_i;\eta)$$

EM for mixtures of experts: example

Iter 1

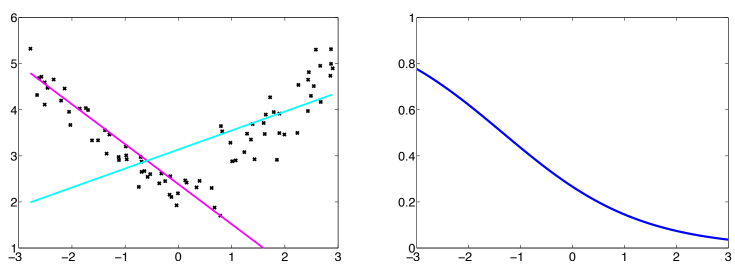


Iter 2



EM for mixtures of experts: example

Iter 3



Iter 7