iterative alg. for MLE

Expectation Maximization

Machine Learning – CSE546
Emily Fox
University of Washington

November 6, 2013

Iterative Algorithm

- Motivates a coordinate ascent-like algorithm:
  1. Infer missing values z^i given estimate of parameters $\hat{\theta}$
  2. Optimize parameters to produce new $\hat{\theta}$ given "filled in" data z^i
  3. Repeat

$z_i \xleftarrow{\text{est}} \hat{\theta}$, $z_i \xrightarrow{\text{max}} \hat{\theta}$

$\hat{\theta} = \{\hat{\phi}_k, \pi_k\}$

- Example: MoG (derivation soon… + HW)
  1. Infer "responsibilities"

prev. iter.

soft weights

$$r_{ik} = p(z^i = k \mid x^i, \hat{\theta}^{(t-1)}) = \frac{\pi_k^{(t-1)} p(x_i \mid \phi_k^{(t-1)})}{\sum_j \pi_j^{(t-1)} p(x_i \mid \phi_j^{(t-1)})}$$

  2. Optimize parameters

$$\max \text{ w.r.t. } \pi_k : \quad \hat{\pi}_k^{(t)} = \frac{1}{N} \sum_i r_{ik} = \frac{r_k}{N}$$

← soft counts!

$$\max \text{ w.r.t. } \mu_k, \Sigma_k :$$

$$\hat{\mu}_k^{(t)} = \frac{\sum r_{ik} x_i}{\cancel{N}\ r_k} \qquad r_k = \sum_i r_{ik}$$

← weighted mean

$$\hat{\Sigma}_k^{(t)} = \left(\frac{1}{r_k} \sum r_{ik} x_i x_i^T \right) - \hat{\mu}_k^{(t)} \hat{\mu}_k^{(t)T}$$

weighted covariance

Expectation Maximization (EM) – Setup

- More broadly applicable than just to mixture models considered so far
- Model: x observable – *"incomplete" data* ← what we have
  y not (fully) observable – *"complete" data* ← what we wish we had
  θ parameters
  introduce
- Interested in maximizing (wrt θ):
  max θ
  $$p(x \mid \theta) = \sum_y p(x, y \mid \theta) = \sum_y p(x \mid y, \theta)\, p(y \mid \theta)$$
- Special case: ← non-invertible, deterministic fcn
  $$x = g(y)$$
  e.g. $y = \begin{bmatrix} z \\ x \end{bmatrix}$ ← class labels ← obs. in standard mix. models

Expectation Maximization (EM) – Derivation

- Step 1
  - Rewrite desired likelihood in terms of complete data terms
  $$p(y \mid \theta) = p(y \mid x, \theta)\, p(x \mid \theta)$$
  $p(y, x \mid \theta)$ ← bc $x = g(y)$
  ← quantity of interest, max θ
  $$\Rightarrow \underbrace{\log p(x \mid \theta)}_{L_x(\theta)} = \log p(y \mid \theta) - \log p(y \mid x, \theta)$$
  $L_x(\theta)$ ← want max
  ↖ you can compute easily
- Step 2
  - Assume estimate of parameters $\hat{\theta}$
  - Take expectation with respect to $p(y \mid x, \hat{\theta})$ "$E[\cdot \mid x, \hat{\theta}]$"

  max expected log likelihood
  $$L_x(\theta) = \underbrace{E[\log p(y \mid \theta) \mid x, \hat{\theta}]}_{U(\theta, \hat{\theta})} + \underbrace{E[-\log p(y \mid x, \theta) \mid x, \hat{\theta}]}_{V(\theta, \hat{\theta})}$$

Expectation Maximization (EM) – Derivation

- Step 3
  - Consider log likelihood of data at any θ relative to log likelihood at $\hat{\theta}$

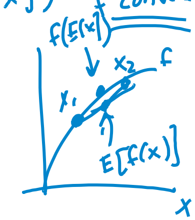
advantage of moving to θ when $\hat{\theta}$

$$\underbrace{L_x(\theta) - L_x(\hat{\theta})}_{\text{want } \geq 0} = [U(\theta,\hat{\theta}) - U(\hat{\theta},\hat{\theta})] + \underbrace{[V(\theta,\hat{\theta}) - V(\hat{\theta},\hat{\theta})]}_{\geq 0}$$

as long as $U(\theta,\hat{\theta}) \geq U(\hat{\theta},\hat{\theta})$ making progress. ✓

- **Aside: Gibbs Inequality** $E_p[\log p(x)] \geq E_p[\log q(x)]$

Proof: Use Jensen's Ineq. $E[f(x)] \leq f(E[x])$ for any f convex



Here:

$$E_p[\log q] - E_p[\log p] = E_p\left[\log \frac{q}{p}\right]$$

$$\leq \log E_p\left[\frac{q}{p}\right] = \log \int_x \cancel{p(x)} \frac{q(x)}{\cancel{p(x)}} dx = 0$$

Expectation Maximization (EM) – Derivation

pedal going down hill

$$L_x(\theta) - L_x(\hat{\theta}) = [U(\theta,\hat{\theta}) - U(\hat{\theta},\hat{\theta})] + \underbrace{[V(\theta,\hat{\theta}) - V(\hat{\theta},\hat{\theta})]}_{\geq 0 \checkmark}$$

- Step 4
  - Determine conditions under which log likelihood at θ exceeds that at $\hat{\theta}$

  Using Gibbs inequality:

$$V(\theta,\hat{\theta}) = E[-\log p(y|x,\theta) \mid x,\hat{\theta}] \geq E[-\log p(y|x,\hat{\theta}) \mid x,\hat{\theta}]$$
$$= V(\hat{\theta},\hat{\theta}) \quad \forall \theta$$

If $U(\theta,\hat{\theta}) \geq U(\hat{\theta},\hat{\theta})$

Then

$$L_x(\theta) \geq L_x(\hat{\theta})$$

making progress

$\boxed{\text{choosing } \theta \text{ s.t. this is true means we're moving in the right direction (or at least not wrong)}}$

Motivates EM Algorithm

- Initial guess: $\hat{\theta}^{(0)}$
- Estimate at iteration *t*: $\hat{\theta}^{(t)}$

get function $U(\theta, \hat{\theta}^{(t)})$

- **E-Step**

  Compute $U(\theta, \hat{\theta}^{(t)}) = E[\log p(y \mid \theta) \mid x, \hat{\theta}^{(t)}]$

max function U (pedal forwards)

- **M-Step**

  Compute $\hat{\theta}^{(t+1)} = \arg\max_{\theta} U(\theta, \hat{\theta}^{(t)})$

From before, $U(\hat{\theta}^{(t+1)}, \hat{\theta}^{(t)}) \geq U(\hat{\theta}^{(t)}, \hat{\theta}^{(t)})$

$\Rightarrow L_x(\hat{\theta}^{(t+1)}) \geq L_x(\hat{\theta}^{(t)})$

$\log P(y \mid \theta) = \log \prod_i P(x^i, z^i \mid \theta) = \sum_i \log P(x^i, z^i \mid \theta) = \sum \log \pi_z P(x^i \mid \phi)$

Example – Mixture Models

$P(z, x \mid \pi, \mu, \Sigma)$

Computing r_{ik}

- **E-Step** Compute $U(\theta, \hat{\theta}^{(t)}) = E[\log p(y \mid \theta) \mid x, \hat{\theta}^{(t)}]$
- **M-Step** Compute $\hat{\theta}^{(t+1)} = \arg\max_{\theta} U(\theta, \hat{\theta}^{(t)})$ ← estimating π, μ, Σ

prob point belongs to cluster z_i | prob location generated by $N(\mu_i, \Sigma_i)$

- Consider $y^i = \{z^i, x^i\}$ i.i.d.

$P(z = z^i \mid \pi)$, μ_i, Σ_i

$p(x^i, z^i \mid \theta) = \pi_{z^i} p(x^i \mid \phi_{z^i}) = \prod_{k=1}^{K} \left(\pi_k P(x^i \mid \phi_k) \right)^{\mathbb{1}(z^i = k)}$

$\max_{\pi, \mu, \Sigma} E_{q_t}[\log p(y \mid \theta)] = \sum_i E_{q_t}[\log p(x^i, z^i \mid \theta)] = \sum E\left[\sum_k \mathbb{1}(z^i = k) \log \pi_k P(x^i \mid \phi_k)\right]$

$= \sum_{i=1}^{N} \sum_{k=1}^{K} \underbrace{E[\mathbb{1}(z^i = k) \mid x^i, \hat{\theta}^{(t)}]}_{P(z^i \mid x^i, \hat{\theta}^{(t)}) = r_{ik}} \left(\log \pi_k P(x^i \mid \phi_k) \right) = \underbrace{\sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \log \pi_k P(x^i \mid \phi_k)}_{\text{weighted log likelihood}}$

same as optimizing over weighted data

e.g., $\max_{\pi} \Rightarrow \pi_k = \frac{\sum_i r_{ik}}{N}$

Coordinate Ascent Behavior

$V(\theta,\theta^{(t)}) - V(\theta^{(t)},\theta^{(t)}) \geq 0$

$V(\theta,\theta^{(t)}) \geq V(\theta^{(t)},\theta^{(t)})$

- Bound log likelihood:

max θ

$$L_x(\theta) = U(\theta, \hat{\theta}^{(t)}) + V(\theta, \hat{\theta}^{(t)})$$
$$\geq U(\theta, \hat{\theta}^{(t)}) + V(\theta^{(t)}, \theta^{(t)}) \equiv LB_x(\theta, \hat{\theta}^{(t)})$$

max a concave lower bound

lower bound tight at $\hat{\theta}^{(t)}$

$$L_x(\hat{\theta}^{(t)}) = U(\hat{\theta}^{(t)}, \hat{\theta}^{(t)}) + V(\hat{\theta}^{(t)}, \hat{\theta}^{(t)}) = LB_x(\hat{\theta}^{(t)}, \hat{\theta}^{(t)})$$

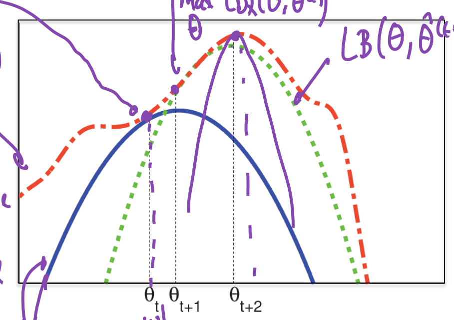


Figure from KM textbook

EM: sequence of max of concave lower bound to log likelihood

Comments on EM

- Since Gibbs inequality is satisfied with equality only if *p=q*, any step that changes θ should strictly **increase likelihood**
  or converged & likelihood doesn't change
- In practice, can replace the **M-Step** with increasing *U* instead of maximizing it (**Generalized EM**) e.g., a gradient step
- Under certain conditions (e.g., in exponential family), can show that EM **converges to a stationary point** of $L_x(\theta)$ + latent variables
- Often there is a **natural choice for *y*** … has physical meaning
  y = (cluster assignment, x)
- If you want to choose any *y*, not necessarily *x=g(y)*, replace $p(y \mid \theta)$ in *U* with $p(y, x \mid \theta)$

Initialization

- In mixture model case where $y^i = \{z^i, x^i\}$ there are many ways to initialize the EM algorithm
- Examples: B K-means
  - Choose K observations at random to define each cluster. Assign other observations to the nearest "centriod" to form initial parameter estimates
  - Pick the centers sequentially to provide good coverage of data
  - Grow mixture model by splitting (and sometimes removing) clusters until K clusters are formed
- Can be quite important to convergence rates in practice or quality of solution

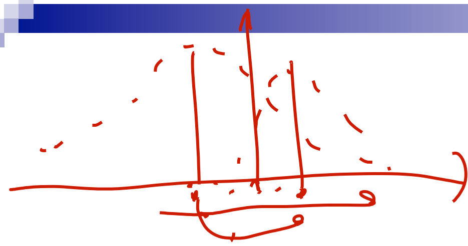
 11

What you should know

- K-means for clustering:
  - algorithm
  - converges because it's coordinate ascent
- EM for mixture of Gaussians:
  - How to "learn" maximum likelihood parameters (locally max. like.) in the case of unlabeled data
- Be happy with this kind of probabilistic analysis
- Remember, E.M. can get stuck in local minima, and empirically it DOES
- EM is coordinate ascent

 12

Curve ....



Dimensionality Reduction PCA

Machine Learning – CSE4546
Carlos Guestrin
University of Washington

November 13, 2013

Dimensionality reduction

- Input data may have thousands or millions of dimensions!
  - e.g., text data has

X ↳ 10 000 – 1 00 000000 dims

- **Dimensionality reduction**: represent data with fewer dimensions
  - easier learning – fewer parameters
  - visualization – hard to visualize more than 3D or 4D
  - discover "intrinsic dimensionality" of data
    - high dimensional data that is truly lower dimensional

Lower dimensional projections

- Rather than picking a subset of the features, we can new features that are combinations of existing features

$z_7 = 2.5x_1 - 2.9x_{2d} + 3.4x_3 \ldots$

$k << d$

X → DZ → model.

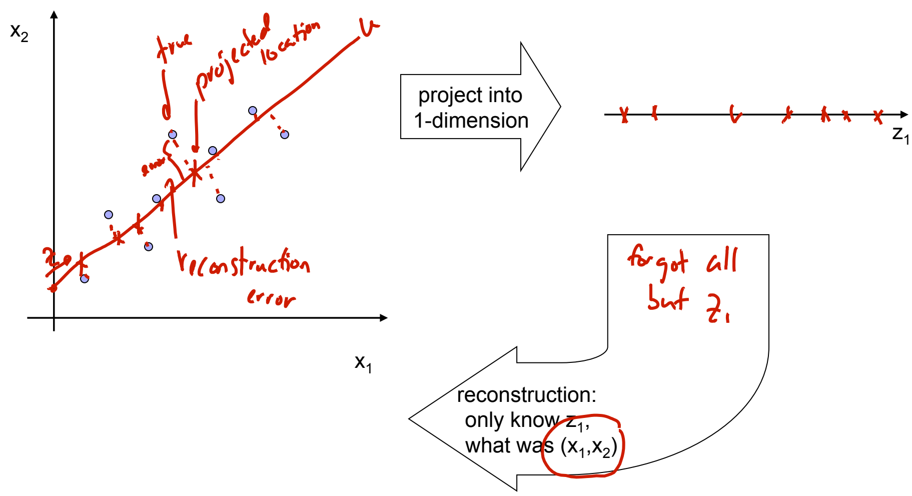
$z = {}_kA\,x$

learn A from data

min reconstruction error

- Let's see this in the unsupervised setting
  - just **X**, but no Y

Linear projection and reconstruction



Principal component analysis – basic idea

- Project d-dimensional data into k-dimensional space while preserving information:
  - e.g., project space of 10000 words into 3-dimensions
  - e.g., project 3-d into 2-d
- Choose projection with minimum reconstruction error

Linear projections, a review

mean $\bar{x}$

- Project a point into a (lower dimensional) space:
  - **point**: $\mathbf{x} = (x_1,\ldots,x_d)$ 1000000
  - **select a basis** – set of basis vectors – $(\mathbf{u}_1,\ldots,\mathbf{u}_k)$ d dimension, learn from data
    - we consider orthonormal basis:
      - $\mathbf{u}_i \bullet \mathbf{u}_i = 1$, and $\mathbf{u}_i \bullet \mathbf{u}_j = 0$ for $i \neq j$
  - **select a center** – $\bar{\mathbf{x}}$, defines offset of space
  - **best coordinates** in lower dimensional space defined by dot-products: $(z_1,\ldots,z_k)$, $z_i = (\mathbf{x}-\bar{\mathbf{x}}) \bullet \mathbf{u}_i$ get z? known
    - minimum squared error

$z_i = (x-\bar{x}) \cdot u_i$? $z_i = \underset{z}{\operatorname{argmin}} \left((x-\bar{x}) - z u_i\right)^2$

 19

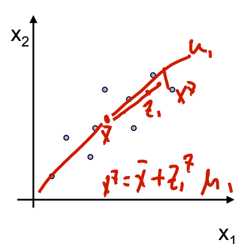
PCA finds projection that minimizes reconstruction error

- Given N data points: $\mathbf{x}^i = (x_1^i,\ldots,x_d^i)$, $i=1\ldots N$
- Will represent each point as a projection: coeffs of projection
  - $\hat{\mathbf{x}}^i = \bar{\mathbf{x}} + \sum_{j=1}^{k} z_j^i \mathbf{u}_j$ where: $\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}^i$ (mean) and $z_j^i = (\mathbf{x}^i - \bar{\mathbf{x}}) \cdot \mathbf{u}_j$

- PCA: 3 1000000
  - Given k<<d, find $(\mathbf{u}_1,\ldots,\mathbf{u}_k)$ minimizing reconstruction error:

$$error_k = \sum_{i=1}^{N} (\mathbf{x}^i - \hat{\mathbf{x}}^i)^2$$

sum over points, squared, true, projection onto lower dimension



 20

Understanding the reconstruction error

u will be ordered such that u_1 more "important" than u_2 ...

$k < d$

approx $\hat{\mathrm{x}}^i = \bar{\mathrm{x}} + \sum_{j=1}^{k} z_j^i \mathbf{u}_j$

$z_j^i = (\mathrm{x}^i - \bar{\mathrm{x}}) \cdot \mathbf{u}_j$

- Given k<<d, find ($\mathbf{u}_1,\ldots,\mathbf{u}_k$) choose u_i minimizing reconstruction error:

$$error_k = \sum_{i=1}^{N} (\mathrm{x}^i - \hat{\mathrm{x}}^i)^2$$

- Note that $\mathbf{x}^i$ can be represented exactly by d-dimensional projection:

$$\mathrm{x}^i = \bar{\mathrm{x}} + \sum_{j=1}^{d} z_j^i \mathbf{u}_j$$

- Rewriting error:

$$error_k = \sum_{i=1}^{N} (x^i - \hat{x}^i)^2 = \sum_{i=1}^{N} \left(\bar{x} + \sum_{j=1}^{d} z_j^i u_j - \left[\bar{x} + \sum_{j=1}^{k} z_j^i u_j\right]\right)^2$$

$$= \sum_{i=1}^{N} \left(\sum_{j=k+1}^{d} z_j^i u_j\right)^2 \leftarrow \text{error is part ignored}$$

$$= \sum_{i=1}^{N} \left[\sum_{j=k+1}^{d} z_j^i \underbrace{u_j \cdot u_j}_{1} z_j^i + \sum_{j=k+1}^{d} \sum_{\ell \neq j} z_j^i \underbrace{u_j \cdot u_\ell}_{0} z_\ell^i \right]$$

$$= \sum_{i=1}^{N} \sum_{j=k+1}^{d} (z_j^i)^2 \leftarrow \text{min error} \equiv \text{min projection onto ignored directions}$$

$z_j^i = (x^i - \bar{x}) \cdot u_j$

Reconstruction error and covariance matrix

$$error_k = \sum_{i=1}^{N} \sum_{j=k+1}^{d} [\underbrace{\mathbf{u}_j \cdot (\mathrm{x}^i - \bar{\mathrm{x}})}_{z_j^i}]^2$$

$$= \sum_{i=1}^{N} \sum_{j=k+1}^{d} u_j^T (x^i - \bar{x}) (x^i - \bar{x})^T u_j$$

$$= \sum_{j=k+1}^{d} u_j^T \underbrace{\left[\sum_{i=1}^{N} (x^i - \bar{x}) (x^i - \bar{x})^T \right]}_{N \Sigma} u_j$$

min error:

$$\min_{u} N \sum_{j=k+1}^{d} u_j^T \Sigma u_j$$

↑ find u_j that minimize this error

$$\Sigma = \frac{1}{\mathrm{N}} \sum_{i=1}^{\mathrm{N}} (\mathrm{x}^i - \bar{\mathrm{x}})(\mathrm{x}^i - \bar{\mathrm{x}})^T$$

$$\Sigma = \begin{pmatrix} \sigma_i^2 & \sigma_{ij} \\ & \end{pmatrix}$$

$$\sigma_{uv} \stackrel{MLE}{=} \frac{1}{N} \sum_{i=1}^{N} (x_u^i - \bar{x}_u)(x_v^i - \bar{x}_v)$$

in vector format

$$\Sigma \stackrel{MLE}{=} \frac{1}{N} \sum_{i=1}^{N} (x^i - \bar{x})(x^i - \bar{x})^T$$