Gaussian Networks

CS B553
Spring 2013

Announcements

• Final projects
  – Get started!
  – Come see me or Alex if you want to talk, otherwise we’ll assume everything is going great…

Central Limit Theorem intuition

• Galton’s board: Balls dropped on triangular array of nails
  – After dropping many balls, what distribution should the bins at bottom obey?

Karl Sims

plot(X)
plot(conv(X,X))
plot(conv(X,conv(X,X)))
plot($\text{conv}(X,\text{conv}(X,\text{conv}(X,\text{conv}(X,\text{conv}(X,X)))))))$

plot($\text{conv}(X,X))$

plot($\text{conv}(X,\text{conv}(X,X)))$

plot($\text{conv}(X,\text{conv}(X,\text{conv}(X,\text{conv}(X,\text{conv}(X,X))))))$

**Multivariate Gaussian distributions**

- Generalization to n dimensional spaces

$$P(x) = \mathcal{N}(x; \mu, \Sigma) = \frac{1}{Z} \exp \left( -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right)$$

$$Z = \sqrt{(2\pi)^n |\Sigma|}$$
Multivariate Gaussian distributions

\[ P(x) = \mathcal{N}(x; \mu, \Sigma) = \frac{1}{2} \exp \left( -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right) \]

\[ Z = \sqrt{(2\pi)^n |\Sigma|} \]

\[
\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 1.1429 & -0.2857 \\ -0.2857 & 0.5714 \end{bmatrix}
\]

**Important properties of covariance matrices**

- Must not be singular
- Must be positive definite
- Easy to check if two variables are independent

**Important properties of multivariate normal distributions**

- Convolution of two multivariate normal distributions is ... ?
- Product of two multivariate normal distributions is ... ?
- Marginals of a normal distribution are ... ?
- Conditional distributions of a normal distribution are ... ?

**Gaussian Bayesian Networks**

- Bayes net where all variables are continuous and all CPDs are linear Gaussians,

\[ P(Y|X = x) = \mathcal{N}(Y; \beta_0 + \beta^T x, \sigma^2) \]

- The joint probability distribution \( P(X_1,..,X_n) \) of a Gaussian Bayes net is a multivariate Gaussian!
  -- So inference on a Gaussian Bayes net is easy. (Why?)
- Any multivariate Gaussian can be represented as a Gaussian Bayes net!

**Gaussian Markov Random Fields**

- Any multivariate Gaussian distribution can be represented as a GMRF
- But, not every GMRF has a multivariate Gaussian as its joint probability distribution
Gaussian Networks

- Gaussian Networks are highly restricted in their modeling power
  - And are not the natural choice for most applications
- But often they are reasonable *approximations* to networks of interest
- Key advantage: Faster inference and fewer parameters!
  - In a discrete Bayes net, the number of parameters and worst-case inference time increases *exponentially* with number of variables
  - In a Gaussian Bayes net, it increases ... ?

Inference on GNs

- Approach 1: Simply construct the multivariate Gaussian distribution, and compute marginals or max marginals directly using above identities.
  - Disadvantages?
- Approach 2: Adapt graphical model inference algorithms to special case of GNs
  - E.g. in sum-product, the messages we compute are always (possibly degenerate) Gaussians
  - Running time is linear in the number of cliques and cubic in the size of the largest clique

Beyond bag-of-words

- Adding geometric structure to an object model is (relatively) easy for specific object recognition
  - And if the object is rigid
  - e.g. [Lowe2003], [Ferrari04], [Rothganger04], [Moreels05]...
- Still not an easy problem: variation due to viewpoint, scale, illumination, occlusion, deformation...

But most (?) objects are flexible

Object class recognition

- Detect instances of broad object categories
  - airplanes, bicycles, cars, faces, motorbikes, ...

Flexible part-based models for object recognition

- SKll not an easy problem: variation due to viewpoint, scale, illumination, occlusion, deformation...
Flexible object models

- Decompose object into parts
  - Individual parts are assumed to be locally rigid
  - Geometric layout of parts can change

Part-based deformable models

- Idea: Model local part appearances, and spatial relationships between parts

- Model $M$ with parts $V = \{v_1, v_2, ..., v_m\}$

- Object configuration $L = (l_{i_1}, ..., l_{i_m})$

- Goal: given image $I$, find configurations that maximize $P(I|L, M)$

  - If part appearances are independent, from Baye’s law,
    $$P(I|L, M) \propto P(I|L) \prod_{v_i \in V} P(I|v_i, M)$$

  - How well does configuration fit spatial model?
  - How well does part location fit local image data?

Form of spatial prior

$$P(L|I, M) \propto P(L|M) \prod_{v_i \in V} P(I|v_i, M)$$

- Usually Gaussians between some pairs of parts

- Several approaches:
  - Bag-of-parts (e.g. [Csurka03], [Dorko04]):
    - Fast inference, but very weak models
  - Trees (e.g. [FH05]):
    - Fast exact inference, but weak models
  - Constellation models [Fergus03]:
    - $2m$-dimensional joint Gaussian
    - Stronger models, but exact inference intractable

$k$-fans

- A family of spatial priors
  - Parameterized on the degree of spatial structure
  - A set $R$ of $k$ reference parts that form a complete subgraph
  - Each non-reference part connected to all parts in $R$

  - $0$-fan (bag-of-parts)
  - $1$-fan
  - $2$-fan
  - $3$-fan
  - $(m-1)$-fan (full joint Gaussian)


$k$-fan inference

- Overall factorization:
  $$P(L|I, M) \propto P(L|M) \prod_{v_i \in L} P(I|v_i, M)$$

  - $0$-fan (bag-of-parts)
  - $1$-fan
  - $2$-fan
  - $3$-fan
  - $(m-1)$-fan (full joint Gaussian)

Bottom-up inference

- A popular heuristic for approximate inference
  - Uses feature detection to identify a small set of salient image points
    (e.g. Harris corners, SIFT, Kadir & Brady)

  - Then maximize $P(L|I, M)$ over that small subset of the configuration space

  - Failure during feature detection can be catastrophic
    - e.g. if occlusions cause parts to be missed
    - Moral: hard intermediate classification decisions are problematic
Avoiding feature detection

- Use feature "operators" that give a likelihood at every pixel location

Back to Assignment 3

- In A3, your program computed messages of the form:
  \[ \delta_{ij}(L_i) = \sum_{l,j} \phi(L_i) \mathcal{N}(L_i - L_j; \Sigma_{ij}, \mu_{ij}) \]
  - This was very slow – quadratic in the number of pixels
- We can speed this up by observing that the message computation is just a convolution, i.e. \( f * g \) where
  \[ f(x) = \phi(x) \quad g(x) = \mathcal{N}(L_i - L_j; \Sigma_{ij}, \mu_{ij}) \]
- Convolutions can be performed with FFT in \( O(n \log n) \) time

Overall Computation for 2 Parts

- Image and model (translation)
- Match cost of each part \( m_1(l_1), m_2(l_2) \)
- Convolution of \( m_1(l_1) \) and log Gaussian
- Minimize \( m_1(l_1) + DT_m_2(T_m_1(l_1)) \)

Our approach: unified inference

- We find exact maximizations of the posterior
  - use a feature operator that computes a probability at every location
  - convolutions combine this "soft" evidence efficiently with the prior

Part appearance models

- Can use any part model for which \( P(I|l_i, M) \) can be computed efficiently
  - Or multiple appearance models within the same object model
- Simple template-based operators
  - First run a preprocessor to label each pixel (e.g. edge detector)
  - Templates record probability of observing each label
  - Simple background model: assume pixel independence