Lecture 21: April 8, 2015
cs 573: Probabilistic Reasoning
Professor Nevatia
Spring 2015

Review

- HW #6B posted, due 4/13/15
- Previous Lecture
  - Inference in temporal models
- Today's objective
  - Intro to learning

Learning

- How to construct a graphical model for a problem domain?
  - Structure
    - What are the variables of interest? If they are continuous valued, should they be discretized?
    - What should be the links between the variable nodes?
      - Should the edges be directed or not?
  - Parameters
    - Specify CPDs or affinity factors (for MNs), log linear models for continuous variables
- Can the structure and parameters both be learned simply from examples?

Motivations for Learning

- Problem domain is too complex to design by hand
  - Consider examples of medical diagnosis, computer vision, language understanding…
  - Medical knowledge, language grammar rules exist but may not be completely correct or precise
  - Human vision (and speech) understanding is not fully open to introspection…
- Distributions change over time or from site to site
- Much training data may be available in age of "big data"
  - Evidence-based medicine
- Note: this is NOT a course in Machine Learning, we confine to learning of PGMs
  - Possible overlap with cs567 and some EE courses

Learning Topics

- Is structure given or to be learnt? If former, task is that of parameter learning
- Is any prior knowledge of the model, such as parameter distributions given?
- Complete data (each sample includes values of all the variables in the model) *vs* incomplete data (values of some variables are missing (*hidden*), in each instance.
- Bayesian networks vs Markov networks
  - Parameter estimation in MNs is significantly harder as the partition function entails all variables as opposed to the local normalizing constants in factored CPDs for BNs.
- We will first focus on parameter learning in BNs

Goals of Learning

- Let P^* be the underlying probability distribution and M^* be the corresponding graphical model (BN, MN, template model…)
- Given M samples from P^* , our goal is to estimate M^* ;
  - Find $M^{\sim}$ that best approximates M^* .
- Several notions of approximations are possible:
  - Probability Density (distribution) estimation
  - Prediction/classification tasks
  - Knowledge discovery (e.g. learning a physical law)

Density Estimation

- Find $P^{\sim}$ that best approximates the *generating distribution* P^{*} .
- Use KL-divergence to measure similarity (as in variational approximations)
- $D(P^{*}||P^{\sim}) = E_{\xi \sim P^{*}} [\log ((P^{*}(\xi)/ P^{\sim}(\xi))]$

  $= - H_{P}(X) - E_{\xi \sim P^{*}} [\log P^{\sim}(\xi)]$

  (X is the set of variables over which the distributions are defined)
- Only the second term is a function of $P^{\sim}$, and is called *expected log-likelihood* . We want to maximize this term.
  - Assigns high probabilities to observed instances

Likelihood

- *likelihood* of the data, given a model M is P(D : M)
  - Note D is a set of data, probability of set D when the model is given by M
  - ":" is not the same as "|" in conditonal probability
- Let D be set of M independent, identically distributed (iid) samples: $D = \{\xi[1], \xi[2], \ldots, \xi[M]\}$

  then, $P(D : M) = \prod_{m=1,M} P(\xi[m] : M)$
- *log-likelihood* $l(D : M) = log\ P(D : M)$

  $log\ P(D : M) = \sum_{m=1,M} P(\log \xi[m] : M)$
- *log-loss* is negative of log-likelihood = $-l(D : M)$
- loss (ξ : M) is loss associated with sample ξ
  - Loss = 0 if P (ξ : M) = 1
- Expected loss (or risk) is $E_{\xi \sim P^*}$ [loss (ξ : M)]

Prediction Task

- We may care about a specific aspect of the distribution only, say predicting P(Y|X); X and Y are specific subsets of the variables
- Most common is the task of classification; find the most likely assignment for Y.
  - Loss function could be binary: how many samples are classified correctly?
- This can be considered to be *discriminative* training
- We will focus on density estimation in this course which may be considered to be *generative* training
- Discriminative learning has shown better performance in recent years (for specific tasks) but typically requires much more training data (harder to generalize)

Overfitting etc

- Overfitting and generalization
  - Learned model fits the training data very well but does not generalize to unseen data (*e.g.* new patients, new images….)
- Bias-variance
  - If we choose a simple model for approximation, it will remain sub-optimal even with large training data: this introduces a *bias*.
  - If model is highly expressive (has many parameters), it can fit P* well but requires more training data: with limited data, results will have high *variance*.
- Regularization
  - Model that has terms that reward good fit to the data but also terms that penalize addition of parameters

Evaluating Performance

- *Holdout testing*: split data into training and test sets: D_{train} and D_{test} .
  - Can be used to compare performance of different learning methods
  - For one method, difference between performance on the training and test sets should not be too large; otherwise *overfitting* is indicated
  - How to split data between two sets? Trade-off between not enough data for training vs not enough for testing
- Cross-validation
  - Divide data into *k* sets; use one for testing, rest for training
  - Repeat *k*-times; *k-fold cross-validation.*
  - Leave one out: test on only one, use rest for testing.
  - Can also measure variance in performance
- Validation dataset
  - In cross-validation, test data is also used to select the best algorithm (or parameters)
  - Should test on a set not used in the optimization procedure
- Independent Test Set: use only once!

Parameter Estimation (ch. 17 KF)

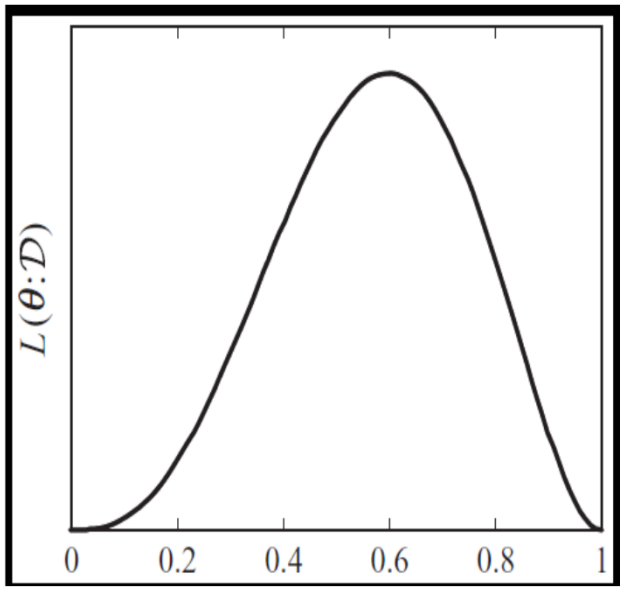
- Dataset D = {ξ[1], ξ[2],…. ξ[M])
- Two approaches
  - Maximum likelihood estimator: model giving highest probability to data
  - Bayesian estimator: takes prior probabilities of distributions into account
- Example: Toss a thumbtack, heads or tails

  - Assume trials are iid (independent and identically distributed)
  - θ is the probability of heads
  - Θ: hypothesis space (set of all parameters θ in [0,1])

Parameter Estimation

- $P(<H,T,T,H,H>) = \theta\,(1-\theta)(1-\theta)\theta*\theta = \theta^3(1-\theta)^2$
- *Likelihood function*, likelihood as a function of parameters (θ); note reversal of order of data and θ in the following notation:

  $L(\theta : <H,T,T,H,H>) = P(<H,T,T,H,H> : \theta) = \theta^3(1-\theta)^2$



Log L = 3 log θ + 2 log (1- θ)

Take derivative w.r.t. θ and set = 0 to maximize; gives θ = .6

Agrees with "frequency" interpretation

General Case

- General likelihood for M[1] heads, M[0] tails is
  $L(\theta : D) = \theta^{M[1]}(1-\theta)^{M[0]}$

- log-likelihood is $l(\theta : D) = M[1]\log\theta + M[0]\log(1-\theta)$
- Differentiate w.r.t. θ , set equal to zero, we get

$$\hat{\theta} = \frac{M[1]}{M[1] + M[0]}$$

- Answer is consistent with our intuition, just measure the frequency
- Note that MLE estimate may be incorrect if number of trials is small (e.g. 1). *Confidence interval* is an estimate of the variance.

MLE Principle

- $\boldsymbol{X}$ is the set of random variables
- Dataset D = $\{\xi[1], \xi[2], \ldots. \xi[M])$
- *Parametric* model: $P(\xi ; \theta)$; ξ is an instance, $\sum_{\xi} P(\xi : \theta) = 1$
- $L(\boldsymbol{\theta} : D) = \prod_m P(\xi[m] : \theta)$
- MLE gives $\hat{\boldsymbol{\theta}} = \max_{\boldsymbol{\theta} \text{ in } \Theta} L(\boldsymbol{\theta} : D)$
- Various Cases
  - $\Theta_{\text{thumbtack}} = [0,1]$
  - Let X take one of *k* values $x^1, \ldots.. x^k$; then,

    $P_{\text{multinomial}}(x ; \boldsymbol{\theta}) = \theta_k$ if $x = x^k$; note $\boldsymbol{\theta}$ is a k-dimensional vector; θ_k is the k^{th} element.
    - $\Theta_{\text{multinomial}} = \{\boldsymbol{\theta} \text{ in } [0,1]^k : \sum_i \theta_i = 1\}$
  - $\Theta_{\text{Gaussian}} = R \times R^{+}$; $\boldsymbol{\theta} = \langle \mu, \sigma \rangle$ (note σ must be positive)

Sufficient Statistics

- Informally: captures all the needed info in the samples;
- Formal Definition 17.1

> *A function $\tau(\xi)$ from instances of $\mathcal{X}$ to $\mathbb{R}^\ell$ (for some ℓ) is a* sufficient statistic *if, for any two data sets $\mathcal{D}$ and $\mathcal{D}'$ and any $\boldsymbol{\theta} \in \Theta$, we have that*
>
> $$\sum_{\xi[m] \in \mathcal{D}} \tau(\xi[m]) = \sum_{\xi'[m] \in \mathcal{D}'} \tau(\xi'[m]) \implies L(\boldsymbol{\theta} : \mathcal{D}) = L(\boldsymbol{\theta} : \mathcal{D}').$$
>
> We often refer to the tuple $\sum_{\xi[m] \in \mathcal{D}} \tau(\xi[m])$ as the sufficient statistics of the data set $\mathcal{D}$.

- For thumbtack, it is counts of M[1] and M[0]
- For multinomial, it is <M[1]…. M[K]>
  - $\tau(x^k)$ is k-dimensional vector with value 1 where $x = x^k$;
  - $L(\boldsymbol{\theta} : D) = \prod_k \theta_k^{M[k]}$
  - Max is given by $\hat{\theta_k} = M[k]/M$ (easy to derive by taking derivative of the log of the L function above and setting equal to zero)
- For Gaussian distribution, $\tau(x) = <1, x, x^2>$
  - μ is given by average of sample values
  - σ given by square terms (subtract x terms)
  - Max of likelihood is given by empirical mean and variance

MLE for Bayesian Networks

- Structure of BN allows estimation to reduce to a set of independent parameters.
- First consider network of two binary variables $X \rightarrow Y$
- CPD: prior probability of X given by θ_{x1} and θ_{x0}

  Conditional probabilities $\theta_{Y|X} = \theta_{Y|x1} \cup \theta_{Y|x0}$

  $\theta_{Y|x1} = \{\theta_{y1|x1}, \theta_{y0|x1}\}$; $\theta_{Y|x0} = \{\theta_{y1|x0}, \theta_{y0|x0}\}$
- Each training instance is a tuple $< x[m], y[m] >$
- Likelihood function is: $$L(\boldsymbol{\theta} : \mathcal{D}) = \prod_{m=1}^{M} P(x[m], y[m] : \boldsymbol{\theta}).$$
- Factorize: $$L(\boldsymbol{\theta} : \mathcal{D}) = \prod_{m} P(x[m] : \boldsymbol{\theta}) P(y[m] \mid x[m] : \boldsymbol{\theta}).$$
- Distribute multiply: $$L(\boldsymbol{\theta} : \mathcal{D}) = \left(\prod_{m} P(x[m] : \boldsymbol{\theta})\right) \left(\prod_{m} P(y[m] \mid x[m] : \boldsymbol{\theta})\right)$$
- Note that first term is probability of a single variable so can be optimized as before (from counts of 1 and 0 values)

MLE for Bayesian Networks

- Simplify second term (from previous slide)

$$\prod_m P(y[m] \mid x[m] : \boldsymbol{\theta}_{Y|X})$$

$$= \prod_{m:x[m]=x^0} P(y[m] \mid x[m] : \boldsymbol{\theta}_{Y|X}) \cdot \prod_{m:x[m]=x^1} P(y[m] \mid x[m] : \boldsymbol{\theta}_{Y|X})$$

$$= \prod_{m:x[m]=x^0} P(y[m] \mid x[m] : \boldsymbol{\theta}_{Y|x^0}) \cdot \prod_{m:x[m]=x^1} P(y[m] \mid x[m] : \boldsymbol{\theta}_{Y|x^1}).$$

- Consider P(y[m] | x[m] : $\boldsymbol{\theta}_{Y|x0}$), if y[m]=$y^1$, this term is $\theta_{y1|x0}$ otherwise $\theta_{y0|x0}$.
- Let M[x0, y^1] be number of samples where X = x^0, Y = y^1, etc then: $\prod_{m:x[m]=x^0} P(y[m] \mid x[m] : \boldsymbol{\theta}_{Y|x^0}) = \theta_{y1|x0}^{M[x0,y1]} \cdot \theta_{y0|x0}^{M[x0,y0]}$

- Maximizing will give us $\theta_{y1|x0} = M[x^0, y^1] / M[x^0]$
  - Again, what we would have expected
  - Similar expressions for other terms

General likelihood Decomposition

- Bayesian network G, parameters $\boldsymbol{\theta}$
- Given Dataset D = {ξ[1], ξ[2],…. ξ[M]}
- $L(\boldsymbol{\theta} : D) = \prod_m P(\xi[m] : \boldsymbol{\theta})$

$$= \prod_m \prod_i P(x_i[m] \mid pa_{Xi}[m] : \boldsymbol{\theta})$$

$$= \prod_i \{\prod_m P(x_i[m] \mid pa_{Xi}[m] : \boldsymbol{\theta})\}$$ (reverse product order)

$$= \prod_i L_i(\boldsymbol{\theta}_{Xi|paXi} : D)$$; L_i is the local likelihood of X*i* .

$$L_i(\boldsymbol{\theta}_{Xi|paXi} : D) = \prod_m P(x_i[m] \mid pa_{Xi}[m] : \boldsymbol{\theta}_{Xi|paXi})$$

- Likelihood decomposes; we can maximize each L_i independently; Proposition 17.1

Let $\mathcal{D}$ be a complete data set for $X_1, \ldots, X_n$, let $\mathcal{G}$ be a network structure over these variables, and suppose that the parameters $\theta_{X_i|\mathrm{Pa}_{X_i}}$ are disjoint from $\theta_{X_j|\mathrm{Pa}_{X_j}}$ for all $j \neq i$. Let $\hat{\theta}_{X_i|\mathrm{Pa}_{X_i}}$ be the parameters that maximize $L_i(\theta_{X_i|\mathrm{Pa}_{X_i}} : \mathcal{D})$. Then, $\hat{\theta} = \langle \hat{\theta}_{X_1|\mathrm{Pa}_1}, \ldots, \hat{\theta}_{X_n|\mathrm{Pa}_n} \rangle$ maximizes $L(\theta : \mathcal{D})$.

Table CPDs

- Variable X with parents **U**, CPD is P(X|**U**), entries are $\theta_{x|\mathbf{u}}$
- $L_X(\boldsymbol{\theta}_{X|U} : D) = \prod_m \theta_{x[m]|\mathbf{u}[m]}$
  $= \prod_{\mathbf{u}} \prod_x \theta_{x|\mathbf{u}}^{M[\mathbf{u},x]}$
- Maximize for each $\theta_{x|\mathbf{u}}$ independently, subject to $\sum \theta_{x|\mathbf{u}} = 1$
- $\hat{\theta}_{x|\mathbf{u}} = M[\mathbf{u},x]/ \sum_x M[\mathbf{u},x] = M[\mathbf{u},x]/ M[\mathbf{u}]$;
- Note: as data will be divided into several subsets, such as shown above, it may be hard to get enough samples for each subset, possibly resulting in overfitting.

Gaussian BNs

- Linear Gaussian CPDs of the form:
  - $P(X|\mathbf{u}) = N(\beta_0 + \beta_1 u_1 + \ldots \beta_k u_k ; \sigma^2)$
- Task is to learn $(\beta_0 , \beta_1 \ldots \beta_k , \sigma)$
- Can follow s derivation similar to that for discrete case; final result is as expected:
  - Estimate means of X and **U**
  - Estimate covariance matrix of {X} U **U**
  - Can solve for parameters from the above two (Thm 7.4)

Next Class

- Read sections 17.3 and 17.4