Lecture 9c - Unsupervised Learning under Uncertainty

Jesse Hoey
School of Computer Science
University of Waterloo

June 30, 2022

Readings: Poole & Mackworth (2nd. Ed.) Chapt. 10.2,10.3,10.5

Incomplete Data

- So far:
  - values of all attributes are known
  - learning is easy
- But many real-world problems have hidden variables (aka latent variables)
  - Incomplete data
  - Values of some attributes missing
- Incomplete data $\rightarrow$ unsupervised learning

Maximum Likelihood learning

Recall: ML learning of Bayes net parameters for each variable V with parents $pa(V)$, and each value those parents can take on $pa(V) = \mathrm{v}$:

$$\theta_{V=true,pa(V)=\mathrm{v}} = P(V = true|pa(V) = \mathrm{v})$$

so that the ML learning of θ is:

$$\theta_{V=true,pa(V)=\mathrm{v}} = \frac{\text{number with } (V = true \wedge pa(V) = \mathrm{v})}{\text{number with } pa(V) = \mathrm{v}}$$

Can add pseudocounts as priors
But what if some variable values are missing?

Complete vs. Missing Data

For Cancer diagnosis example:

- Complete data (what we used to learn from in lecture 9a)

id	Malfnction	Cancer	TestB	TestA	Report	Database
1	false	false	true	true	false	false
2	false	true	true	true	true	true
3	false	true	true	true	true	true
4	false	false	false	true	false	false

...

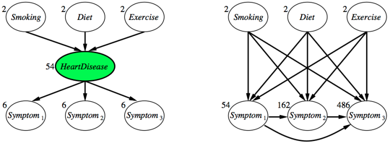
- Incomplete (missing) data (more realistic)

id	Malfnction	Cancer	TestB	TestA	Report	Database
1	?	?	?	true	?	false
2	?	?	?	true	?	false
3	true	?	?	true	?	false
4	?	true	?	true	?	false

...

How to deal with missing data

1. Ignore hidden variables
   number of parameters shown (variables have 3 values):



(a) (b)

2. Ignore records with missing values
   - does not work with true latent variables (e.g. always missing)

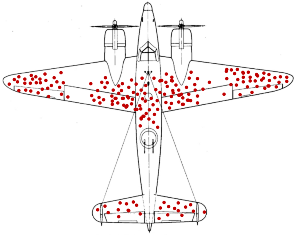
Missing Data

- You cannot ignore missing data unless you know it is missing at random.
- Often data is missing because of something correlated with a variable of interest.
- For example: data in a clinical trial to test a drug may be missing because:
  - the patient dies,
  - the patient dropped out because of severe side effects,
  - they dropped out because they were better, or
  - the patient had to visit a sick relative.

  — ignoring some of these may make the drug look better or worse than it is.
- In general you need to model why data is missing.

Survivorship Bias

Bullet holes on planes returning from battle:
where should the extra armour be installed?



Abraham Wald (WWII)

"Direct" maximum likelihood

3. maximize likelihood directly
   Suppose Z is hidden and E is observable, with values e

$$
\begin{aligned}
h_{ML} &= \arg\max_h P(\mathrm{e}|h) \\
&= \arg\max_h \left[\sum_Z P(\mathrm{e}, \mathrm{Z}|h)\right] \\
&= \arg\max_h \left[\sum_Z \prod_{i=1}^{n} P(X_i|parents(X_i), h)_{\mathrm{E=e}}\right] \\
&= \arg\max_h \left[\log \sum_Z \prod_{i=1}^{n} P(X_i|parents(X_i), h)_{\mathrm{E=e}}\right]
\end{aligned}
$$

Problem: can't push log inside the sum to linearize!

Expectation-Maximization Algorithm

4. If we knew the missing values, computing h_{ML} would be easy again!
   Expectation-Maximization (EM):
   A). Guess h_{ML}
   B). iterate:
   - expectation: based on h_{ML}, compute expectation of missing values $P(Z|h_{ML}, e)$
   - maximization: based on expected missing values, compute new estimate of h_{ML}
5. Really simple version (e.g. K-means algorithm):
   - expectation: based on h_{ML}, compute most likely missing values $\arg\max_Z P(Z|h_{ML}, e)$
   - maximization: based on those missing values, you now have complete data, so compute new estimate of h_{ML} using ML learning as before

k-means algorithm

k-means algorithm can be used for clustering: dataset of observables with input features X generated by one of a set of classes, C (e.g. Naïve Bayes, $C \to X$)

Inputs:

- training examples
- the number of classes, k

Outputs:

- a representative value for each input feature for each class
- an assignment of examples to classes

Algorithm:

1. pick k means in X, one per class, C
2. iterate until means stop changing:
   a assign examples to k classes (e.g. as closest to current means)
   b re-estimate k means based on assignment

Expectation Maximization

- Approximate the maximum likelihood
- Start with a guess h_0
- Iteratively compute:

$$h_{i+1} = \arg\max_h \sum_Z P(Z|h_i, e) \log P(e, Z|h)$$

- expectation: compute $P(Z|h_i, e)$ ("fills in" missing data)
- maximization: find new h that maximizes $\sum_Z P(Z|h_i, e) \log P(e, Z|h)$
- can show that $P(e|h_{i+1}) \geq P(e|h_i)$ when computed like this

Expectation Maximization

Can show that:

$$\log P(e|h) \geq \sum_Z P(Z|e, h) log P(e, Z|h)$$

EM finds a local maximum of right side: lower bound on left side

log inside sum can linearize the product

$$h_{i+1} = \arg\max_h \sum_Z P(Z|h_i, e) \log P(e, Z|h)$$

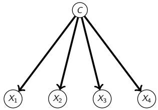
$$= \arg\max_h \sum_Z P(Z|h_i, e) \log \prod_{j=1}^{n} P(X_i|parents(X_i), h)_{E=e}$$

$$= \arg\max_h \sum_Z P(Z|h_i, e) \sum_{j=1}^{n} \log P(X_i|parents(X_i), h)_{E=e}$$

EM monotonically improves the likelihood

$$P(e|h_{i+1}) \geq P(e|h_i)$$

Naive Bayes with 4 input features



- Suppose $k = 3$, and $dom(C) = \{1, 2, 3\}$.
- $P(C|X_1, X_2, X_3, X_4) \propto P(X_1 \dots X_4|C)P(C)$
- can be computed if we know $P(C)$ and $P(X_i|C)$
- EM idea: based on current $P(C)$ and $P(X_i|C)$, compute $P(C|X_1 \dots X_4) \forall C \in \{1, 2, 3\}$
- use $P(C|X_1 \dots X_4)$ as partial data in ML learning

Augmented Data Method — E step – Naive Bayes

$P(C = 1|X_1 = t, X_2 = f, X_3 = t, X_4 = t) = 0.4$
$P(C = 2|X_1 = t, X_2 = f, X_3 = t, X_4 = t) = 0.1$
$P(C = 3|X_1 = t, X_2 = f, X_3 = t, X_4 = t) = 0.5$:

missing data (C) $\longrightarrow$ filled in data

X_1	X_2	X_3	X_4
⋮	⋮	⋮	⋮
t	f	t	t
⋮	⋮	⋮	⋮

$\longrightarrow$

X_1	X_2	X_3	X_4	C	val
⋮	⋮	⋮	⋮	⋮	⋮
t	f	t	t	1	0.4
t	f	t	t	2	0.1
t	f	t	t	3	0.5
⋮	⋮	⋮	⋮	⋮	⋮

call this $A[X_1, \dots, X_4, C]$

M step

Compute the statistics for each feature and class:

$$M_i[X_i, C] = \sum_{X_1,\dots,X_{i-1},X_{i+1},\dots,X_n} A[X_1, \dots, X_n, C]$$

$$M[C] = \sum_{X_i} M_i[X_i, C]$$

$M[C]$ is unnormalized marginal.

M step

Compute the statistics for each feature and class:

$$M_i[X_i, C] = \sum_{X_1,\dots,X_{i-1},X_{i+1},\dots,X_n} A[X_1, \dots, X_n, C]$$

$$M[C] = \sum_{X_i} M_i[X_i, C]$$

$M[C]$ is unnormalized marginal.
Compute probabilities by normalizing:

$$P(X_i|C) = M_i[X_i, C]/M[C]$$

$$P(C) = M[C]/s$$

Pseudo-counts can also be added.

General Bayes Network EM

Complete data : Bayes Net Maximum Likelihood

$$\theta_{V=true,pa(V)=v} = \frac{\text{number in e with } (V = true \wedge pa(V) = v)}{\text{number in e with } pa(V) = v}$$

Incomplete data : Bayes Net Expectation Maximization
observed variables X and missing variables Z
Start with some guess for θ,
E Step : Compute weights for each data x_i and latent variable(s) value(s) z_j (using e.g. variable elimination)

$$w_{ij} = P(z_j|\theta, x_i)$$

M Step : Update parameters:

$$\theta_{V=true,pa(V)=v} = \frac{\sum_{ij} w_{ij}|V = true \wedge pa(V) = v \text{ in } \{x_i, z_j\}}{\sum_{ij} w_{ij}|pa(V) = v \text{ in } \{x_i, z_j\}}$$

Belief network structure learning (I)

$$P(model|data) = \frac{P(data|model) \times P(model)}{P(data).}$$

- A model here is a belief network.
- A bigger network can always fit the data better.
- $P(model)$ lets us encode a preference for smaller networks (e.g., using the description length).
- You can search over network structure looking for the most likely model.

Belief network structure learning (II)

- can do independence tests to determine which features should be the parents
- XOR problem : just because features do not give information individually, does not mean they will not give information in combination
- ideal: Search over total orderings of variables

Next:

- Planning with uncertainty (Poole & Mackworth (2nd. Ed.) chapter 9.1-9.3,9.5)
- Reinforcement Learning (Poole & Mackworth (2nd. Ed.) chapter 12.1,12.3-12.9)