Probabilistic Graphical Models

Vocabulary

The model is a declarative representation of our understanding of the world.
A graphical model is essentially a way of representing joint probability distribution over a set of random variables in a compact and intuitive form. There are two main types of graphical models, namely directed and undirected. We generally use a directed model, also known as a Bayesian network.

Operations on Factors

A **factor** is a function or a table denoted as ϕ or Φ. It takes a random variable and gives us a value for every assignment of the random variable

$$\phi : Val(X_1, ..., X_k) \to \Re$$

Scope is the set of arguments, $X_1, X_2, ..., X_k$
Factor Product
The Factor Product is the product of two factors:

$$\phi_1(A, B) * \phi_2(B, C) = \phi_3(A, B, C)$$

This results in independent probabilities for the joint distribution $P(A, B, C)$
If we want to obtain the factor product for a specific assignment to the variables in the scope of the factors, such as $P(A = 1, B = 1, C = 1)$, we would perform the following multiplication:

$$P(A^1, B^1) * P(B^1, C^1) = P(A^1, B^1, C^1)$$

Factor Marginalization
If we have a factor whose scope is A, B, C and we want to remove B from its scope, called marginalization of B, we sum the factor across all values of B.

$$\phi(A, C) = \sum_B \phi(A, B, C); \quad P(A, C) = \sum_B P(A, B, C)$$

Factor Reduction
When we want to obtain the values of the factor for a specific value of one of the variables in the factor's scope. for example: $\phi(A, B, C = 1) = \phi(A, B, C^1)$
$P(A, B, C = 1) = P(A, B, C^1)$
the scope of the new factor is now A, B

Bayesian network

Bayesian networks (BN) [1] are directed acyclic graph (DAG) models with nodes and edges.
node - a variable/factor in the model
edge - a connection between nodes in the model, they can be directional or non-directional
Chain rule: The Chain Rule is a Factor Product of all the CPDs of the Bayesian Network.

$$P(X_1, .., X_n) = \prod_i P(X_i | parent_G(X_i))$$

in/out-degree of $v(\Delta_+(v), \Delta_-(v))$: number of parents, children of v

Reasoning Patterns

Causal Reasoning: Reasoning goes in a causal direction $A \to B \to C$, also top-down.
Evidential Reasoning:Reasoning goes in an evidential direction $A \leftarrow B \leftarrow C$, also bottom-up.
Evidential Reasoning:This type of reasoning is present when we have what is called a "V-structure" in the graph. In Intercausal Reasoning, we look at the flow of information between two causes across this v-structure, $A \to B \leftarrow C$.

Flow of Probabilistic Influence I

When can X influence Y

$X \to Y$ Yes	$X \leftarrow Y$ Yes
$X \to W \to Y$ Yes	$X \leftarrow W \leftarrow Y$ Yes
$X \leftarrow W \to Y$ Yes	$X \to W \leftarrow Y$ No

When can X influence Y given evidence about Z

	$W \notin Z$	$W \in Z$
$X \to W \to Y$	Yes	No
$X \leftarrow W \leftarrow Y$	Yes	No
$X \leftarrow W \to Y$	Yes	No
$X \to W \leftarrow Y$	No[a]	Yes[b]

Trail: A sequence of nodes combined by edges in the graph; $X_1 - ... - X_k$.

[a] No if w and all of its descendants $\notin Z$
[b] yes if w and all of its descendants $\in Z$

Flow of Probabilistic Influence II

Active Trail

- A trail is active if no variables are observed ($X_i \notin Z$), and it has no V-structures such as $X_{i-1} \to X_i \leftarrow X_{i+1}$
- For any structure $X_{i-1} \to X_i \leftarrow X_{i+1}$ we have observed X_i (i.e., the variable at the vertex of the V-structure) or one of its descendants, and no other X_i is observed

Independence

$\models$ is the symbol for **satisfies**
$\perp$ is the symbol for **independence**
For random variables $X, Y, P \models X \perp Y$ in other words, "P satisfies X's independence of Y"

- $P(X, Y) = P(X) * P(Y)$
- $P(X \mid Y) = P(X)$
- $P(Y \mid X) = P(Y)$ (symmetry)

Conditional Independence $P \models X \perp Y \mid Z$

- $P(X, Y \mid Z) = P(X \mid Z) * P(Y \mid Z)$
- $P(X \mid Y, Z) = P(X \mid Z)$
- $P(Y \mid X, Z) = P(Y \mid Z)$

Independencies in Bayesian Networks

d-separation: X and Y are d-separated in G given Z if there is no active trail in G between X and Y given Z. In other words, if all paths from a vertex of A to a vertex of B are blocked w.r.t. (with respect to) to C.
d-separation implies independence
Any Node is d-separated from its non-descendants given its parents.
Independence Map (I-maps)
All of the independencies in G are implied by all of the independence statements that correspond to d-separation statements in the graph.

$$I(G) = \{(X \perp Y \mid Z) : d - sep_G(X, Y \mid Z)\}$$

If P satisfies $I(G)$, then G is an I-map of P

Notes

Factorization: G allows P to be represented
I-map: Independencies encoded by G hold in P
If P factorizes over a graph G, we can read from the graph independencies that must hold in P (an I-map)

Naive Bayes

The only independence assumption made in the Naive Bayes model is: given the class variable, all features are independent of one another... $P \models X_i \perp X_j \mid C$
This allows us to encode the chain rule for Naive Bayes Model in a simple way:

$$P(C, X_1, ..., X_n) = P(C) \prod_{i=1}^{n} P(X_i \mid C)$$

Ratio of the probability of two classes

$$\frac{P(C = c^1 \mid x_1, ..., x_n)}{P(C = c^2 \mid x_1, ..., x_n)} = \frac{P(C = c^1)}{P(C = c^2)} \prod_{i=1}^{n} \frac{P(x_i \mid C = c^1)}{P(x_i \mid C = c^2)}$$

Bernoulli Naive Bayes for Text

Each variable is a binary variable subject to a Bernoulli Distribution.
Naive Bayes, because we make very strong independence assumptions that the event of one word appearing is independent of the event of a different word appearing given that we know the class.
Imagine trying to determine the type of document we have given a list of unique words in the document.
For each word in the dictionary, we have a binary random variable:

- 1 - if word in dictionary appears in the doc
- 0 - if word does not

Multinomial Naive Bayes for Text model

the variables that represent the features are words in the document; n is the length of the document.

Template Models I

Image Segmentation
The same model can be shared for every pixel within an image
The same model can be shared between images of the same database

Template Models II

The following are advantages of template models:
Template models can often capture events that occur in a time series.
CPDs in template models can often be copied many times.
Template models can capture parameter sharing within a model.
Distribution over temporal trajectories
discretize time by Δ
$X^{(t)}$ a specific instantiation of variable X at time $t\Delta$
$X^{(t:t')} = \{X^{(t)}, ..., X^{(t')}\}$ where $(t \leq t')$
Out goal is to be able to represent a probability distribution $P(X^{(t:t')})$ for any t, t'.
Markov Assumption
The Markov Assumption allows us to compact the model
The chain rule for probabilities gives the following

$$P(X^{(0:T)}) = P(X^{(0)}) \prod_{t=0}^{T-1} P(X^{(t+1)} \mid P(X^{(0:t)})$$

The Markov Assumption $(X^{(t+1)} \perp X^{(0:t-1)}) \mid X^{(t)}$
The state at T+1 is independent of the past given the present.
Now we can simplify the previous statement to:

$$P(X^{(0:T)}) = P(X^{(0)}) \prod_{t=0}^{T-1} P(X^{(t+1)} \mid P(X^{(t)})$$

Time Invariance
For every $X^{(t)}$ we will have the following *template probability model*: $P(X' \mid X)$
Important notation in template probability models:
$X^{(t+1)} = X'$ denotes the next timpeoint
$X^{(t)} = X$ denotes the current timepoint
For every single timepoint t, this model is replicated such that:

$$P(X^{(t+1)} \mid P(X^{(t)}) = P(X' \mid X)$$

This replication is called **time invariance**
Two-Time-Slice Bayesian Netowrk (2TBN)
A dynamic Bayesian network (DBN) over $X_1, ..., X_n$ is defined by a:
1. 2TBN, defined as $BN_{\rightarrow}$, over $X_1, ..., X_n$
2. a Bayesian network $BN^{(0)}$ over $X_1^{(0)}, ..., X_n^{(0)}$

Template Models III

Ground Baysian Network
for a trajectory over $0, ..., T$ we define a ground (unrolled) network such that:
1. the dependency model for $X_1^{(0)}, ..., X_n^{(0)}$ is copied from $BN^{(0)}$
2. the dependency model for $X_1^{(t)}, ..., X_n^{(t)}$ for all $t > 0$ is copied from $BN_{\rightarrow}$

Markov chain

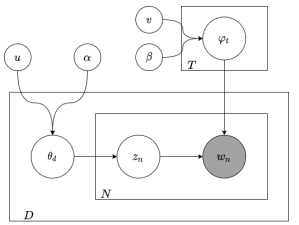
Markov chain is specified by the following components:
$Q = q_1 q_2 \dots q_N$: a set of N states

$A = a_{11} a_{12} \dots a_{n1} \dots a_{nn}$: a transition probability matrix A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{n} a_{ij} = 1 \quad \forall i$

$\pi = \pi_1, \pi_2, \dots, \pi_N$ an initial probability distribution over states. π_i is the probability that the Markov chain will start in state i. Some states j may have $\pi_j = 0$, meaning that they cannot be initial states. Also, $\sum_{i=1}^{n} \pi_i = 1$

Plate notation

Useful for encoding networks that have multiple objects of the same type.
A plate denotes n occurrences of a variable t.
For every instance of the plate variable $t_n = i$, there is a random variable $X(t_i)$.
Nested Plates: Plates within a plate .
Each random variable(s) in the plate is indexed by both the variable of the plate and the variable of the plate within which it is nested.



$p(w, z, \theta, \varphi, u, v, \alpha, \beta) = p(\alpha)p(\beta)p(u)p(v)$
$\prod_{t=1,...,T} p(\varphi_t \mid v, \beta) \prod_{d=1,...,D} p(\theta_d \mid u, \alpha)$
$\prod_{n=1,...,N} p(w_{dn} \mid z_{dn}) p(z_{dn} \mid \theta_d)$

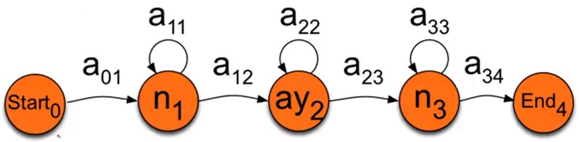
Hidden Markov Models

Hidden Markov Models (HMM) are comprised of a Transition Model, and an Observation Model

Applications

* Robot localization
* Speech Recognition
* Biological sequence analysis
* Text annotation

HMM used to determine the phoneme sequence (nine)



An HMM [2] is specified by the following components:
$Q = q_1 q_2 \dots q_N$ a set of N states
$A = a_{11} \dots a_{ij} \dots a_{NN}$ a transition probability matrix A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{N} a_{ij} = 1$
$\forall i \;\; O = o_1 o_2 \dots o_T$ a sequence of T observations.
$B = b_i(o_t)$ a sequence of observation likelihoods, also called emission probabilities, each expressing the probability of an observation o_t being generated from a state i
$\pi = \pi_1, \pi_2, \dots, \pi_N$ an initial probability distribution over states. π_i is the probability that the Markov chain will start in state i. Some states j may have $\pi_j = 0$ meaning that they cannot be initial states. Also, $\sum_{i=1}^{n} \pi_i = 1$
the first-order hidden Markov model instantiates two simplifying assumptions:

Markov Assumption:

$$P(q_i \mid q_1 \dots q_{i-1}) = P(q_i \mid q_{i-1})$$

Output Independence:

$$P(o_i \mid q_1 \dots q_i, \dots, q_T, o_1, \dots, o_i, \dots, o_T) = P(o_i \mid q_i)$$

Problem 1 (Likelihood): Given an HMM $\lambda = (A, B)$ and an observation sequence O, determine the likelihood $P(O \mid \lambda)$.

Problem 2 (Decoding): Given an observation sequence O and an HMM $\lambda = (A, B)$, discover the best hidden state sequence Q

Problem 3 (Learning): Given an observation sequence O and the set of states in the HMM, learn the HMM parameters A and B.

Hidden Markov Models II

Likelihood Computation: The Forward Algorithm

$$P(O \mid Q) = \prod_{i=1}^{T} P(o_i \mid q_i)$$

$$P(O, Q) = P(O \mid Q) \times P(Q)$$

$$P(O, Q) = \prod_{i=1}^{T} P(o_i \mid q_i) \times \prod_{i=1}^{T} P(q_i \mid q_{i-1})$$

The forward algorithm is a dynamic programming algorithm

HMMs Forward Backward & Viterbi Algo

Recall HMM:

Set of states: $Q = \{q_1, q_2, \dots, q_n\}$

Set of observations: $O = \{o_1, o_2, \dots, o_m\}$

Vector of prior probabilities:
$\Pi = \{\pi_1, \pi_2, \dots, \pi_n\}$, where $\pi_i = P(S_0 = q_i)$
Matrix of transition prob:
$A = \{a_{ij}\}$ where $a_{ij} = P(S_t = q_j \mid S_{t-1} = q_i)$
Matrix of observation prob:
$B = \{b_{ij}\}$ where $b_{ik} = P(O_t = o_k \mid S_t = q_i)$

- Evaluation

Determine the probability of an observation sequence, $O = o_1 o_2 \dots o_T$, given a model, λ, that is, estimating $P(O|\lambda)$.

Direct Method

$$P(O \mid \lambda) = \textstyle\sum_i P(O, Q_i \mid \lambda)$$

To obtain $P(O, Q_i \mid \lambda)$ we multiply the probability of the initial state, q_1 by the transition probabilities for a state sequence, $q_1, q_2, \dots$ and the observation probabilities for an observation sequence, $o_1, o_2, \dots$

$$P(O, Q_i \mid \lambda) = \pi_1 b_1(o_1)\, a_{12} b_2(o_2) \dots a_{(T-1)T} b_T(o_T)$$

For a model with N states and an observation length of T , there are N T possible state sequences. Each term in the summation requires 2T operations. As a result, the evaluation requires a number of operations of the order of 2T × N T .

- Iterative Method (Forward) [3]

Estimate the probabilities of the states/observations per time step
we define an auxiliary variable called forward:

Text classification I

Text classification is the task of classifying documents by their content: that is, by the words of which they are comprised. (email spam filtering)
Often a document is represented as a bag of words.
Consider a document D, whose class is given by C. In the case of email spam filtering there are two classes C = S (spam) and C = H (ham). We classify D as the class which has the highest posterior probability $P(C|D)$, which can be re-expressed using Bayes' Theorem.

Problem statement

- Documents $X \in \mathcal{X}$
- Associated to n_K classes $Y \in [1 \dots n_K]$
- Observing $\boldsymbol{X}$, predict y

Solution

Introduce random variables
Build a model for $\mathrm{P}(\boldsymbol{X}, Y)$ (joint distribution)
Given $X = x$, compute (Bayes rule):

$$\mathrm{P}(Y \mid \boldsymbol{x}) = \frac{\mathrm{P}(\boldsymbol{X}, Y)}{\mathrm{P}(\boldsymbol{x})} = \frac{\mathrm{P}(\boldsymbol{X} = \boldsymbol{x} \mid Y)\mathrm{P}(Y)}{\sum_y \mathrm{P}(\boldsymbol{X} = \boldsymbol{x} \mid Y = y)\mathrm{P}(Y = y)}$$

- Choose $y = \operatorname{argmax} \mathrm{P}(Y \mid \boldsymbol{x})$

Naive Bayes, Bernoulli model

- fixed vocabulary of n_W types. Document $= n_W$ variables : $X \in \mathcal{X} = \{0, 1\}^{nW}$
- X_w tests w in doc.: $X_w \sim \mathcal{B}(\beta_w) \Leftrightarrow \mathrm{P}(X_w = 1) = \beta_w (\epsilon[0, 1])$ (Bernoulli)
- independence between X_w and $X_v (\forall w \neq v)$

$$\mathrm{P}(\boldsymbol{X} = \boldsymbol{x}) = \mathrm{P}(X_w = x_w)$$

$$\mathrm{P}_\beta(\boldsymbol{X} = \boldsymbol{x}) = \textstyle\prod_{w=1}^{n_W} \beta_w^{x_w} (1 - \beta_w)^{(1 - x_w)}$$

With uniform class distribution, the optimal class for $x^\star$

$$\begin{aligned} y^\star &= \operatorname*{argmax}_{y=1\dots n_K} \mathrm{P}_\theta(\boldsymbol{x}^\star, y) \propto \prod_{w=1}^{w=n_W} \beta_{yw}^{x_{w^\star}} (1 - \beta_{yw})^{(1 - x_{w^\star})} \\ &= \operatorname*{argmax}_{y=1\dots n_K} \log \mathrm{P}_\theta(\boldsymbol{x}^\star, y) \\ &\propto \sum_{w=1}^{w=n_W} x_{w^\star} \log \beta_{yw} + (1 - x_{w^\star}) \log(1 - \beta_{yw}) \end{aligned}$$

HMMs Forward Backward & Viterbi Algo

$$\alpha_t(i) = P\left(o_1, o_2, \ldots, o_t, S_t = q_i \mid \lambda\right)$$

That is, the probability of the partial sequence of observations until time t, being in state q_i at time t.

Algorithm 1: `The Forward Algorithm`

Require: HMM, λ; Observation sequence, O; Number of states, N; Number of observations, T

for $i \leftarrow 1$ ***To*** N **do**

  Initialization

$$\alpha_1(i) = P\left(O_1, S_1 = q_i\right) = \pi_i b_i\left(O_1\right)$$

end

for $t \leftarrow 2$ ***To*** T **do**

  for $j \leftarrow 1$ ***To*** N **do**

    Induction

$$\alpha_t(j) = \left[\sum_i \alpha_{t-1}(i) a_{ij}\right] b_j\left(O_t\right)$$

  end

end

$P(O) = \sum_i \alpha_T(i)$ (Termination)

return $P(O)$

N-grams and Probabilities

N-gram An N-gram is a sequence of N words
Corpus: I am happy pecause I am learning
Unigrams: {I, am, happy, because, learning }
Bigrams: {I am, am happy, happy because ...}
Trigrams: { I am happy am happy because, ... }

Probability of N-gram:$P\left(w_N \mid w_1^{N-1}\right) = \dfrac{C\left(w_1^{N-1} w_N\right)}{C\left(w_1^{N-1}\right)}$

Corpus: This is great ... teacher drinks tea.

$w_1 \quad w_2 \quad w_3 \quad w_{498} \quad w_{499} \quad w_{500}$

$w_1^m = w_1 w_2 \ldots w_m$

Reference

References

[1] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009. Google-Books-ID: 7dzpHCHzNQ4C.

[2] Speech and language processing, 2020. https://web.stanford.edu/ jurafsky/slp3/.

[3] Luis Enrique Sucar. Hidden markov models. In Luis Enrique Sucar, editor, *Probabilistic Graphical Models: Principles and Applications*, Advances in Computer Vision and Pattern Recognition, pages 63–82. Springer.