CS 391L: Machine Learning: Bayesian Learning: Beyond Naïve Bayes

Raymond J. Mooney
University of Texas at Austin

Logistic Regression

- Assumes a parametric form for directly estimating P(Y | X). For binary concepts, this is:

$$P(Y=1\,|\,X)=\frac{1}{1+\exp(w_0+\sum_{i=1}^{n}w_iX_i)}$$

$$P(Y=0\,|\,X)=1-P(Y=1\,|\,X)$$

$$=\frac{\exp(w_0+\sum_{i=1}^{n}w_iX_i)}{1+\exp(w_0+\sum_{i=1}^{n}w_iX_i)}$$

- Equivalent to a one-layer backpropagation neural net.
  - Logistic regression is the source of the sigmoid function used in backpropagation.
  - Objective function for training is somewhat different.

Logistic Regression as a Log-Linear Model

- Logistic regression is basically a linear model, which is demonstrated by taking logs.

$$\text{Assign label } Y=0 \text{ iff } 1<\frac{P(Y=0\,|\,X)}{P(Y=1\,|\,X)}$$

$$1<\exp(w_0+\sum_{i=1}^{n}w_iX_i)$$

$$0<w_0+\sum_{i=1}^{n}w_iX_i$$

$$\text{or equivalently }\quad w_0>\sum_{i=1}^{n}-w_iX_i$$

- Also called a **maximum entropy model (MaxEnt)** because it can be shown that standard training for logistic regression gives the distribution with maximum entropy that is consistent with the training data.

Logistic Regression Training

- Weights are set during training to maximize the **conditional data likelihood** :

$$W\leftarrow \underset{W}{\operatorname{argmax}}\prod_{d\in D}P(Y^d\,|\,X^d,W)$$

where D is the set of training examples and Y^d and X^d denote, respectively, the values of Y and X for example d.

- Equivalently viewed as maximizing the **conditional log likelihood** (CLL)

$$W\leftarrow \underset{W}{\operatorname{argmax}}\sum_{d\in D}\ln P(Y^d\,|\,X^d,W)$$

Logistic Regression Training

- Like neural-nets, can use standard gradient descent to find the parameters (weights) that optimize the CLL objective function.
- Many other more advanced training methods are possible to speed convergence.
  - Conjugate gradient
  - Generalized Iterative Scaling (GIS)
  - Improved Iterative Scaling (IIS)
  - Limited-memory quasi-Newton (L-BFGS)

Preventing Overfitting in Logistic Regression

- To prevent overfitting, one can use **regularization** (a.k.a. smoothing) by penalizing large weights by changing the training objective:

$$W\leftarrow \underset{W}{\operatorname{argmax}}\sum_{d\in D}\ln P(Y^d\,|\,X^d,W)-\frac{\lambda}{2}\|W\|^2$$

Where λ is a constant that determines the amount of smoothing

- This can be shown to be equivalent to assuming a Guassian prior for W with zero mean and a variance related to $1/\lambda$.

Multinomial Logistic Regression

- Logistic regression can be generalized to multi-class problems (where *Y* has a multinomial distribution).
- Effectively constructs a linear classifier for each category.

Relation Between Naïve Bayes and Logistic Regression

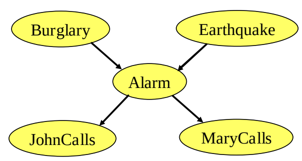
- Naïve Bayes with Gaussian distributions for features (GNB), can be shown to given the same functional form for the conditional distribution P(*Y*|*X*).
  - But converse is not true, so Logistic Regression makes a weaker assumption.
- Logistic regression is a **discriminative** rather than generative model, since it models the conditional distribution P(*Y*|*X*) and directly attempts to fit the training data for predicting *Y* from *X*. Does not specify a full joint distribution.
- When conditional independence is violated, logistic regression gives better generalization if it is given sufficient training data.
- GNB converges to accurate parameter estimates faster (O(log *n*) examples for *n* features) compared to Logistic Regression (O(*n*) examples).
  - Experimentally, GNB is better when training data is scarce, logistic regression is better when it is plentiful.

Graphical Models

- If no assumption of independence is made, then an exponential number of parameters must be estimated for sound probabilistic inference.
- No realistic amount of training data is sufficient to estimate so many parameters.
- If a blanket assumption of conditional independence is made, efficient training and inference is possible, but such a strong assumption is rarely warranted.
- **Graphical models** use directed or undirected graphs over a set of random variables to explicitly specify variable dependencies and allow for less restrictive independence assumptions while limiting the number of parameters that must be estimated.
  - **Bayesian Networks**: Directed acyclic graphs that indicate causal structure.
  - **Markov Networks**: Undirected graphs that capture general dependencies.

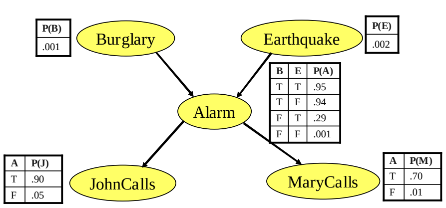
Bayesian Networks

- Directed Acyclic Graph (DAG)
  - Nodes are random variables
  - Edges indicate causal influences



Conditional Probability Tables

- Each node has a **conditional probability table** (**CPT**) that gives the probability of each of its values given every possible combination of values for its parents (conditioning case).
  - Roots (sources) of the DAG that have no parents are given prior probabilities.



CPT Comments

- Probability of false not given since rows must add to 1.
- Example requires 10 parameters rather than $2^5 – 1 = 31$ for specifying the full joint distribution.
- Number of parameters in the CPT for a node is exponential in the number of parents (fan-in).

Joint Distributions for Bayes Nets

- A Bayesian Network implicitly defines a joint distribution.

$$P(x_1, x_2, \ldots x_n) = \prod_{i=1}^{n} P(x_i \mid \text{Parents}(X_i))$$

- Example

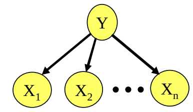
$$P(J \wedge M \wedge A \wedge \neg B \wedge \neg E)$$
$$= P(J \mid A)P(M \mid A)P(A \mid \neg B \wedge \neg E)P(\neg B)P(\neg E)$$
$$= 0.9 \times 0.7 \times 0.001 \times 0.999 \times 0.998 = 0.00062$$

- Therefore an inefficient approach to inference is:
  - 1) Compute the joint distribution using this equation.
  - 2) Compute any desired conditional probability using the joint distribution.

13

Naïve Bayes as a Bayes Net

- Naïve Bayes is a simple Bayes Net

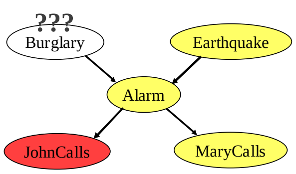


- Priors P(Y) and conditionals P(X_i|Y) for Naïve Bayes provide CPTs for the network.

14

Bayes Net Inference

- Given known values for some **evidence variables**, determine the posterior probability of some **query variables**.
- Example: Given that John calls, what is the probability that there is a Burglary?



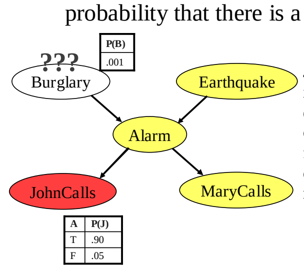
John calls 90% of the time there is an Alarm and the Alarm detects 94% of Burglaries so people generally think it should be fairly high.

However, this ignores the prior probability of John calling.

15

Bayes Net Inference

- Example: Given that John calls, what is the probability that there is a Burglary?

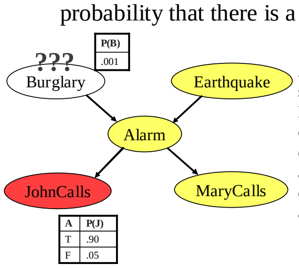


John also calls 5% of the time when there is no Alarm. So over 1,000 days we expect 1 Burglary and John will probably call. However, he will also call with a false report 50 times on average. So the call is about 50 times more likely a false report: P(Burglary | JohnCalls) ≈ 0.02

16

Bayes Net Inference

- Example: Given that John calls, what is the probability that there is a Burglary?



Actual probability of Burglary is 0.016 since the alarm is not perfect (an Earthquake could have set it off or it could have gone off on its own). On the other side, even if there was not an alarm and John called incorrectly, there could have been an undetected Burglary anyway, but this is unlikely.

17

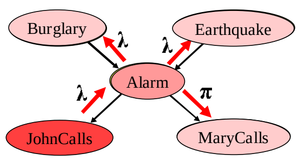
Complexity of Bayes Net Inference

- In general, the problem of Bayes Net inference is NP-hard (exponential in the size of the graph).
- For **singly-connected networks** or **polytrees** in which there are no undirected loops, there are linear-time algorithms based on **belief propagation**.
  - Each node sends local evidence messages to their children and parents.
  - Each node updates belief in each of its possible values based on incoming messages from it neighbors and propagates evidence on to its neighbors.
- There are approximations to inference for general networks based on **loopy belief propagation** that iteratively refines probabilities that converge to accurate values in the limit.

18

Belief Propagation Example

- λ messages are sent from children to parents representing abductive evidence for a node.
- π messages are sent from parents to children representing causal evidence for a node.



19

Markov Networks

- Undirected graph over a set of random variables, where an edge represents a dependency.
- The **Markov blanket** of a node, *X*, in a Markov Net is the set of its neighbors in the graph (nodes that have an edge connecting to *X*).
- Every node in a Markov Net is conditionally independent of every other node given its Markov blanket.

20

Distribution for a Markov Network

- The distribution of a Markov net is most compactly described in terms of a set of **potential functions**, φ_k, for each clique, *k*, in the graph.
- For each joint assignment of values to the variables in clique *k*, φ_k assigns a non-negative real value that represents the compatibility of these values.
- The joint distribution of a Markov is then defined by:

$$P(x_1, x_2, \ldots x_n) = \frac{1}{Z} \prod_k \phi_k(x_{\{k\}})$$

Where $x_{\{k\}}$ represents the joint assignment of the variables in clique *k*, and *Z* is a normalizing constant that makes a joint distribution that sums to 1.

$$Z = \sum_x \prod_k \phi_k(x_{\{k\}})$$

21

Inference in Markov Networks

- Inference in general Markov nets is #P complete.
- Approximation algorithms include:
  - Markov Chain Monte Carlo (MCMC)
  - Loopy belief propagation

22

Bayes Nets vs. Markov Nets

- Bayes nets represent a subclass of joint distributions that capture non-cyclic causal dependencies between variables.
- A Markov net can represent any joint distribution.
  - If network is fully connected then there is one clique that is includes all of the variables and whose potential function directly encodes the full joint distribution.

23

Learning Graphical Models

- **Structure Learning**: Learn the graphical structure of the network.
- **Parameter Learning**: Learn the real-valued parameters of the network
  - CPTs for Bayes Nets
  - Potential functions for Markov Nets

24

Structure Learning

- Use greedy top-down search through the space of networks, considering adding each possible edge one at a time and picking the one that maximizes a statistical evaluation metric that measures fit to the training data.
- Alternative is to test all pairs of nodes to find ones that are statistically correlated and adding edges accordingly.
- Bayes net learning requires determining the direction of causal influences.
- Special algorithms for limited graph topologies.
  - TAN (Tree Augmented Naïve-Bayes) for learning Bayes nets that are trees.

25

Parameter Learning

- If values for all variables are available during training, then parameter estimates can be directly estimated using frequency counts over the training data.
  - Must smooth estimates to compensate for limited training data.
- If there are hidden variables, some form of gradient descent or Expectation Maximization (EM) must be used to estimate distributions for hidden variables.
  - Like setting the weights feeding hidden units in backpropagation neural nets.

26

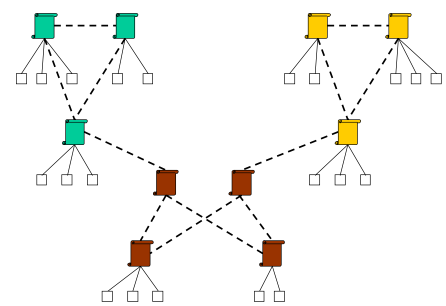
Statistical Relational Learning

- Expand graphical model learning approach to handle instances more expressive than feature vectors that include arbitrary numbers of objects with properties and relations between them.
  - Probabilistic Relational Models (PRMs)
  - Stochastic Logic Programs (SLPs)
  - Bayesian Logic Programs (BLPs)
  - Relational Markov Networks (RMNs)
  - Markov Logic Networks (MLNs)
  - Other TLAs
- **Collective classification**: Classify multiple *dependent* objects based on both and object's properties as well as the class of other related objects.
  - Get beyond IID assumption for instances

27

Collective Classification of Web Pages using RMNs

[Taskar, Abbeel & Koller 2002]



28

Conclusions

- Bayesian learning methods are firmly based on probability theory and exploit advanced methods developed in statistics.
- Naïve Bayes is a simple generative model that works fairly well in practice.
- Logistic Regression is a discriminative classifier that directly models the conditional distribution P(*Y*|*X*).
- Graphical models allow specifying limited dependencies using graphs.
  - Bayes Nets: DAG
  - Markov Nets: Undirected Graph

29