An Introduction to Bayesian Machine Learning

José Miguel Hernández-Lobato

Department of Engineering, Cambridge University

April 8, 2013
What is Machine Learning?

The design of computational systems that discover patterns in a collection of data instances in an automated manner.

The ultimate goal is to use the discovered patterns to make predictions on new data instances not seen before.

Instead of manually encoding patterns in computer programs, we make computers learn these patterns without explicitly programming them.

Figure source [Hinton et al. 2006].
Model-based Machine Learning

We design a probabilistic model which explains how the data is generated. An inference algorithm combines model and data to make predictions. Probability theory is used to deal with uncertainty in the model or the data.
Basics of Probability Theory

The theory of probability can be derived using just two rules:

**Sum rule:**

\[ p(x) = \int p(x, y) \, dy. \]

**Product rule:**

\[ p(x, y) = p(y|x)p(x) = p(x|y)p(y). \]

They can be combined to obtain **Bayes’ rule:**

\[ p(y|x) = \frac{p(x|y)p(y)}{p(x)} = \frac{p(y|x)p(y)}{\int p(x, y) \, dy}. \]

**Independence** of \( X \) and \( Y \): \( p(x, y) = p(x)p(y) \).

**Conditional independence** of \( X \) and \( Y \) given \( Z \): \( p(x, y|z) = p(x|z)p(y|z) \).
The Bayesian Framework

The probabilistic model $\mathcal{M}$ with parameters $\theta$ explains how the data $\mathcal{D}$ is generated by specifying the likelihood function $p(\mathcal{D}|\theta, \mathcal{M})$.

Our initial uncertainty on $\theta$ is encoded in the prior distribution $p(\theta|\mathcal{M})$.

Bayes' rule allows us to update our uncertainty on $\theta$ given $\mathcal{D}$:

$$p(\theta|\mathcal{D}, \mathcal{M}) = \frac{p(\mathcal{D}|\theta, \mathcal{M})p(\theta|\mathcal{M})}{p(\mathcal{D}|\mathcal{M})}.$$ 

We can then generate probabilistic predictions for some quantity $x$ of new data $\mathcal{D}_{\text{new}}$ given $\mathcal{D}$ and $\mathcal{M}$ using

$$p(x|\mathcal{D}_{\text{new}}, \mathcal{D}, \mathcal{M}) = \int p(x|\theta, \mathcal{D}_{\text{new}}, \mathcal{M})p(\theta|\mathcal{D}, \mathcal{M})d\theta.$$ 

These predictions will be approximated using an inference algorithm.
Bayesian Model Comparison

Given a particular $\mathcal{D}$, we can use the model evidence $p(\mathcal{D}|\mathcal{M})$ to reject both overly simple models, and overly complex models.

Figure source [Ghahramani 2012].
Probabilistic Graphical Models

The Bayesian framework requires to specify a high-dimensional distribution \( p(x_1, \ldots, x_k) \) on the data, model parameters and latent variables.

Working with fully flexible joint distributions is intractable!

We will work with structured distributions, in which the random variables interact directly with only few others. These distributions will have many conditional independencies.

This structure will allow us to:
- Obtain a compact representation of the distribution.
- Use computationally efficient inference algorithms.

The framework of probabilistic graphical models allows us to represent and work with such structured distributions in an efficient manner.
Some Examples of Probabilistic Graphical Models

Graphs

Bayesian Network

Markov Network

Independencies

\[(F \perp H|C), (C \perp S|F, H)\]
\[(M \perp H, C|F), (M \perp C|F), \ldots\]
\[(A \perp C|B, D), (B \perp D|A, C)\]

Factorization

\[p(S, F, H, M, C) = p(S)p(F|S)\]
\[p(H|S)p(C|F, H)p(M|F)\]
\[p(A, B, C, D) = \frac{1}{2} \phi_1(A, B)\]
\[\phi_2(B, C)\phi_3(C, D)\phi_4(A, D)\]

Figure source [Koller et al. 2009].
Bayesian Networks

A BN $\mathcal{G}$ is a DAG whose nodes are random variables $X_1, \ldots, X_d$.

Let $\text{PA}_{X_i}^{\mathcal{G}}$ be the parents of $X_i$ in $\mathcal{G}$.

The network is annotated with the conditional distributions $p(X_i|\text{PA}_{X_i}^{\mathcal{G}})$.

**Conditional Independencies:**

Let $\text{ND}_{X_i}^{\mathcal{G}}$ be the variables in $\mathcal{G}$ which are non-descendants of $X_i$ in $\mathcal{G}$.

$\mathcal{G}$ encodes the conditional independencies $(X_i \perp \text{ND}_{X_i}^{\mathcal{G}}|\text{PA}_{X_i}^{\mathcal{G}}), \ i = 1, \ldots, d$.

**Factorization:**

$\mathcal{G}$ encodes the factorization $p(x_1, \ldots, x_d) = \prod_{i=1}^{d} p(x_i|\text{pa}_{X_i}^{\mathcal{G}})$. 
BN Examples: Naive Bayes

We have features $\mathbf{X} = (X_1, \ldots, X_d) \in \mathbb{R}^n$ and a label $Y \in \{1, \ldots, C\}$.

The figure shows the Naive Bayes graphical model with parameters $\theta_1$ and $\theta_2$ for a dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^d$ using plate notation.

Shaded nodes are observed, unshaded nodes are not observed.

The joint distribution for $\mathcal{D}, \theta_1$ and $\theta_2$ is

$$p(\mathcal{D}, \theta_1\theta_2) = \prod_{i=1}^n \left[ \prod_{j=1}^d p(x_{i,j} | y_i, \theta_1) p(y_i | \theta_2) \right] p(\theta_1)p(\theta_2).$$

Figure source [Murphy 2012].
BN Examples: Hidden Markov Model

We have a temporal sequence of measurements \( \mathcal{D} = \{X_1, \ldots, X_T\} \).

Time dependence is explained by a hidden process \( \mathbf{Z} = \{Z_1, \ldots, Z_T\} \), that is modeled by a first-order Markov chain.

The data is a noisy observation of the hidden process.

The joint distribution for \( \mathcal{D}, \mathbf{Z}, \theta_1 \) and \( \theta_2 \) is

\[
p(\mathcal{D}, \mathbf{Z}, \theta_1, \theta_2) = \prod_{t=1}^{T} p(x_t | z_t, \theta_1) \prod_{t=1}^{T} p(z_t | z_{t-1}, \theta_2) \, p(\theta_1) p(\theta_2) p(z_0).
\]

Figure source [Murphy 2012].
BN Examples: Matrix Factorization Model

We have observations $x_{i,j}$ from an $n \times d$ matrix $X$.

The entries of $X$ are generated as a function of the entries of a low rank matrix $UV^T$, $U$ is $n \times k$ and $V$ is $d \times k$ and $k \ll \min(n, d)$.

$$p(D, U, V, \theta_1, \theta_2, \theta_3) = \prod_{i=1}^{n} \prod_{j=1}^{d} p(x_{i,j} | u_i, v_j, \theta_3)$$

$$\prod_{i=1}^{n} p(u_i | \theta_1) \prod_{i=j}^{d} p(v_j | \theta_2) p(\theta_1)p(\theta_2)p(\theta_3).$$
D-separation

Conditional independence properties can be read directly from the graph.

We say that the sets of nodes $A$, $B$ and $C$ satisfy $(A \perp B | C)$ when all of the possible paths from any node in $A$ to any node in $B$ are blocked.

A path will be blocked if it contains a node $x$ with arrows meeting at $x$:
1. i) head-to-tail or ii) tail-to-tail and $x$ is $C$.
2. head-to-head and neither $x$, nor any of its descendants, is in $C$.

$(a \perp b | c)$ does not follow from the graph.

$(a \perp b | f)$ is implied by the graph.

Figure source [Bishop 2006].
Bayesian Networks as Filters

\[ p(x_1, \ldots, x_k) \] must satisfy the CIs implied by d-separation.

\[ p(x_1, \ldots, x_k) \] must factorize as

\[ p(x_1, \ldots, x_d) = \prod_{i=1}^{d} p(x_i|\text{pa}_{G}^G x_i). \]

\[ p(x_1, \ldots, x_k) \] must satisfy the CIs

\[ (X_i \perp_{\text{ND}}^{G} X_i|\text{PA}_{G} x_i), \; i = 1, \ldots, d. \]

The three filters are the same!

Figure source [Bishop 2006].
Markov Blanket

The Markov blanket of a node $x$ is the set of nodes comprising parents, children and co-parents of $x$.

It is the minimal set of nodes that isolates a node from the rest, that is, $x$ is CI of any other node in the graph given its Markov blanket.

Figure source [Bishop 2006].
Markov Networks

A MN is an undirected graph $G$ whose nodes are the r.v. $X_1, \ldots, X_d$. It is annotated with the potential functions $\phi_1(D_1), \ldots, \phi_k(D_k)$, where $D_1, \ldots, D_k$ are sets of variables, each forming a maximal clique of $G$, and $\phi_i, \ldots, \phi_k$ are positive functions.

Conditional Independencies:

$G$ encodes the conditional independencies $(A \perp B \mid C)$ for any sets of nodes $A, B$ and $C$ such that $C$ separates $A$ from $B$ in $G$.

Factorization:

$G$ encodes the factorization $p(X_1, \ldots, X_d) = Z^{-1} \prod_{i=1}^{k} \phi_i(D_i)$, where $Z$ is a normalization constant.
Clique and Maximal Clique

**Clique**: Fully connected subset of nodes.

**Maximal Clique**: A clique in which we cannot include any more nodes without it ceasing to be a clique.

\[(x_1, x_2)\] is a clique but not a maximal clique.

\[(x_2, x_3, x_4)\] is a maximal clique.

\[
p(x_1, \ldots, x_4) = \frac{1}{Z} \phi_1(x_1, x_2, x_3) \phi_2(x_2, x_3, x_4).
\]

Figure source [Bishop 2006].
MN Examples: Potts Model

Let \( x_1, \ldots, x_n \in \{1, \ldots, C\} \),

\[
p(x_1, \ldots, x_n) = \frac{1}{Z} \prod_{i \sim j} \phi_{ij}(x_i, x_j),
\]

where

\[
\log \phi_{ij}(x_i, x_j) = \begin{cases} 
\beta > 0 & \text{if } x_i = x_j \\
0 & \text{otherwise}
\end{cases},
\]

Figure source [Bishop 2006]. Figure source Erik Sudderth.
From Directed Graphs to Undirected Graphs: Moralization

Let \( p(x_1, \ldots, x_4) = p(x_1)p(x_2)p(x_3)p(x_4|x_1, x_2, x_3) \).

How do we obtain the corresponding undirected model?

\( p(x_4|x_1, x_2, x_3) \) implies that \( x_1, \ldots, x_4 \) must be in a maximal clique.

General method:
1. Fully connect all the parents of any node.
2. Eliminate edge directions.

Moralization adds the fewest extra links and so retains the maximum number of CIs.

Figure source [Bishop 2006].
CIs in Directed and Undirected Models
Markov Network in Undirected Models

If all the CIs of $p(x_1, \ldots, x_n)$ are reflected in $\mathcal{G}$, and vice versa, then $\mathcal{G}$ is said to be a perfect map.

Figure source [Bishop 2006].
Basic Distributions: Bernoulli

Distribution for $x \in \{0, 1\}$ governed by $\mu \in [0, 1]$ such that $\mu = p(x = 1)$.

$\text{Bern}(x|\mu) = x\mu + (1 - x)(1 - \mu)$.

$\mathbb{E}(x) = \mu$.

$\text{Var}(x) = \mu(1 - \mu)$. 

![Diagram of Bernoulli distribution with arrows for $\mu$ and $1 - \mu$]
Basic Distributions: Beta

Distribution for $\mu \in [0, 1]$ such as the prob. of a binary event.

\[ \text{Beta}(\mu | a, b) = \frac{\Gamma(a + b)}{\Gamma(a)\Gamma(b)} \mu^{a-1}(1 - \mu)^{b-1}. \]

\[ \mathbb{E}(x) = \frac{a}{a + b}. \]

\[ \text{Var}(x) = \frac{ab}{((a + b)^2(a + b + 1))}. \]
Basic Distributions: Multinomial

We extract with replacement $n$ balls of $k$ different categories from a bag. Let $x_i$ and denote the number of balls extracted and $p_i$ the probability, both of category $i = 1, \ldots, k$.

$$p(x_1, \ldots, x_k|n, p_1, \ldots, p_k) = \begin{cases} \frac{n!}{x_1! \cdots x_k!} p_1^{x_1} \cdots p_k^{x_k} & \text{if } \sum_{i=1}^k x_k = n \\ 0 & \text{otherwise} \end{cases}.$$  

$E(x_i) = np_i.$

$\text{Var}(x_i) = np_i(1 - p_i).$

$\text{Cov}(x_i, x_j) = -np_i p_j (1 - p_i).$
Basic Distributions: Dirichlet

Multivariate distribution over $\mu_1, \ldots, \mu_k \in [0, 1]$, where $\sum_{i=1}^{k} \mu_i = 1$.

Parameterized in terms of $\alpha = (\alpha_1, \ldots, \alpha_k)$ with $\alpha_i > 0$ for $i = 1, \ldots, k$.

$$\text{Dir}(\mu_1, \ldots, \mu_k | \alpha) = \frac{\Gamma \left( \sum_{i=1}^{k} \alpha_k \right)}{\Gamma(\alpha_1) \cdots \Gamma(\alpha_k)} \prod_{i=1}^{k} \mu_i^{\alpha_i}. \quad$$

$$\mathbb{E}(\mu_i) = \frac{a_i}{\sum_{j=1}^{k} a_j}. \quad$$

Figure source [Murphy 2012].
An Introduction to Bayesian Machine Learning

Basic Distributions: Multivariate Gaussian

\[
p(x|\mu, \Sigma) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp \left\{ -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right\}.
\]

\[E(x) = \mu.\]

\[\text{Cov}(x) = \Sigma.\]

Figure source [Murphy 2012].
Basic Distributions: Wishart

Distribution for the precision matrix $\Lambda = \Sigma^{-1}$ of a Multivariate Gaussian.

$$\mathcal{W}(\Lambda | \mathbf{w}, \nu) = B(\mathbf{W}, \nu)|\Lambda|^{(\nu-D-1)} \exp \left\{ -\frac{1}{2} \text{Tr}(\mathbf{W}^{-1} \Lambda) \right\},$$

where

$$B(\mathbf{W}, \nu) \equiv |\mathbf{W}|^{-\nu/2} \left( 2^{\nu D/2} \pi^{D(D-1)/4} \prod_{i=1}^{D} \Gamma \left( \frac{\nu + 1 - i}{2} \right) \right).$$

$$\mathbb{E}(\Lambda) = \nu \mathbf{W}.$$
Summary

With ML computers learn patterns and then use them to make predictions. With ML we avoid to manually encode patterns in computer programs.

Model-based ML separates knowledge about the data generation process (model) from reasoning and prediction (inference algorithm).

The Bayesian framework allows us to do model-based ML using probability distributions which must be structured for tractability.

Probabilistic graphical models encode such structured distributions by specifying several CIs (factorizations) that they must satisfy.

Bayesian Networks and Markov Networks are two different types of graphical models which can express different types of CIs.
References

- Bishop, C. M. Model-based machine learning Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 2013, 371