PROBABILISTIC MODELS – PART 5: LEARNING BAYESIAN NETWORKS

Gerhard Widmer

Institute of Computational Perception
Johannes Kepler University
Linz, Austria

gerhard.widmer@jku.at
www.cp.jku.at/people/widmer

October 28, 2025

Presentation partly based on and inspired by [Koller & Friedman, 2009] and [Russell & Norvig, 2021], including the use of some figures from their books and/or lecture slides.

Many thanks to Daphne Koller, Nir Friedman, Stuart Russell, and Peter Norvig for making these available (`pgm.stanford.edu`; `aima.cs.berkeley.edu`).

Goals of this Lecture

- Discuss some fundamental issues related to learning from examples
- Introduce a criterion to be optimised: the Likelihood
- Explain the bias-variance trade-off and the need for regularisation

- Preview of next chapters:

  Methods for learning the parameters (CPDs) of a model: ☞ **Part 5a**

  Methods for learning the structure of a model: ☞ **Part 5b**

Outline

Motivation

Problem:

- "Manual" construction of model for a given problem may be impossible

  ... because experts not available (or too expensive)

  ... or problem not well enough understood or too complex

Possible Resource:

- May have a set of **example cases** (atomic events) observed in / collected from the world, e.g.:
- *Medical diagnosis:* database of patient records, patient histories, symptoms, tests performed, diagnoses, treatments, treatment outcomes, ...
- *Speech recognition:* recordings of speech annotated with word labels, sentence markers etc.

Goal:

- Construct a structured model of the (hidden) distribution most likely underlying the observed examples

☞ **Automatic Model Learning**

General Setting

Assumptions:

- World to be modelled is governed by some 'true', but unknown **distribution** P^* corresponding to some 'true', but unknown network $\mathcal{M}^* = (\mathcal{G}^*, \theta^*)$
- Given: a set $\mathcal{D} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_M\}$ of M events coming from P^* ("training set")
- Training instances $\boldsymbol{x}_i$ are all sampled, independently, from the same P^*: they are **independent and identically distributed (i.i.d.)**
- In words: The training examples are **representative** of the world P^*

Task:

- Learn some model $\tilde{\mathcal{M}}$ (from a given family of models) with a **distribution** $P_{\tilde{\mathcal{M}}}$ that is an approximation to P^*, and with a graph structure $\tilde{\mathcal{G}}$ that reflects the true **(in)dependencies** in the world.

May want to learn

- only model parameters $\tilde{\theta}$ for a fixed (given) structure, or both structure $\tilde{\mathcal{G}}$ and suitable parameters $\tilde{\theta}$
- a single model, or a whole spectrum (set) of different possible models (e.g., a probability distribution over models)

Learning as an Optimisation Problem

What model do we want to learn?

- Answer 1: *"The correct one"* — not decidable
- Answer 2: *"One that agrees well with the observed events $\mathcal{D}$"*
- Note: The example observations $\mathcal{D}$ is all the information we have ...

General approach:

- Define an **"objective function"** $F(\mathcal{M}, \mathcal{D})$ – a measure that estimates how 'good' a given model $\mathcal{M}$ is in relation to the given training examples $\mathcal{D}$
- Develop an algorithm to find the model that maximises F:

$$\tilde{\mathcal{M}} = \arg\max_{\mathcal{M}} F(\mathcal{M}, \mathcal{D})$$

Learning is a search / optimisation problem:

- Search for a model $\tilde{\mathcal{M}}$ (in *huge* space of possible model structures and parameter settings) with maximum $F(\mathcal{M}, \mathcal{D})$

A Common Objective Function: The Likelihood

Definition

The **Likelihood** of a model $\mathcal{M}$ relative to a dataset $\mathcal{D}$ is **the probability that the model assigns to the set $\mathcal{D}$**:

$$L(\mathcal{M} : \mathcal{D}) = P_{\mathcal{M}}(\mathcal{D})$$

In Words: The Likelihood is ...

- the probability that one would get exactly the instances in $\mathcal{D}$ if one randomly sampled $k = |\mathcal{D}|$ samples from the distribution $P_{\mathcal{M}}$
- the probability that $\mathcal{M}$ would generate exactly the observations $\mathcal{D}$ if asked to generate k samples
- the degree to which $\mathcal{M}$ "fits" or "explains" the observations $\mathcal{D}$.

☞ **Search for model $\tilde{\mathcal{M}}$ with maximal likelihood relative to the given (fixed) training data $\mathcal{D}$.**

Calculating the Likelihood

Definition

If the examples $\mathcal{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M\}$ are **independent and identically distributed (i.i.d)**, the **likelihood** $L(\mathcal{M} : \mathcal{D})$ is

$$L(\mathcal{M} : \mathcal{D}) = P_{\mathcal{M}}(\mathcal{D}) = \prod_{\boldsymbol{x}_i \in \mathcal{D}} P_{\mathcal{M}}(\boldsymbol{x}_i)$$

In Words:

- The likelihood is the product of the probabilities assigned by the model to the individual training examples
- Easy to calculate for a given model and a training set.

The Log-Likelihood

Numerical problems with the likelihood function:

- Probability $P_{\mathcal{M}}(\mathcal{D})$ will be minuscule for any specific set of observations $\mathcal{D}$
- Trying to calculate this will produce an arithmetic underflow
- Solution: Use logarithm instead.

Definition

The **Log-Likelihood** $\ell(\mathcal{M} : \mathcal{D})$ of a model $\mathcal{M}$ relative to a dataset $\mathcal{D}$ is the logarithm of the likelihood:

$$\ell(\mathcal{M} : \mathcal{D}) = \log L(\mathcal{M} : \mathcal{D}) = \log \prod_{\boldsymbol{x}_i \in \mathcal{D}} P_{\mathcal{M}}(\boldsymbol{x}_i) = \sum_{\boldsymbol{x}_i \in \mathcal{D}} \log P_{\mathcal{M}}(\boldsymbol{x}_i)$$

Note:

- Likelihood and log-likelihood are monotonically related:
  $\ell(\mathcal{M} : \mathcal{D})$ has its maximum where $L(\mathcal{M} : \mathcal{D})$ is maximal

☞ **The log-likelihood will form a part of our objective function for learning structured probabilistic models.**

Overfitting vs. Generalisation

Consider a simple example:

- Want to learn a distribution P^* over a probability space defined by 20 binary variables
- There are $2^{20} > 10^6$ possible distinct atomic events in this space
- Training set $\mathcal{D}$ consists of 1000 (different) instances.

Problem:

- If we permit our model class to contain *any distribution* possible over this space, the model $\mathcal{M}_{ML}$ that maximises the (log-)likelihood would be one that
  - assigns equal probability 0.001 to each of the observed training instances
  - and probability 0 to all $(2^{20} - 1000)$ other events.

Resulting model $\mathcal{M}_{ML}$ ...

- tightly fits and reproduces the training examples with high probability
- but assigns zero probability to (i.e., considers *impossible*) anything else.

☞ **Needed: A model that generalises!**

Overfitting vs. Generalisation

Generalisation:

- Given training set $\mathcal{D}$ will usually not contain *all possible* situations (events) that could ever occur
- Purpose of a learned model is to answer queries about new situations
- Model must be more general than simple summary of training set!

Overfitting:

- Model with highest (log-)likelihood generally is one that exactly fits the data
- Assigns high probability to the seen examples, and low/zero probability to any other event
- Not useful for answering queries about new situations
- If we permit arbitrarily complex models and maximise the (log-)likelihood, we will get a model that overfits

☞ **Need some restrictions on allowed models!**

Overfitting and Model Complexity

Note general relation between Overfitting and Model Complexity:

- Fitting a given training set $\mathcal{D}$ with perfect precision requires complex models
  (Model complexity = number of parameters required to specify the model)
- Overfitting models are usually complex models
- Reducing model complexity reduces ability of model to describe $\mathcal{D}$ precisely
- Reducing model complexity enforces generalisation

For Bayesian Network Models:

- Complexity directly relates to number of edges in the graph
- More parents per variable $\Rightarrow$ more numbers in the CPDs
- Reducing number of parents introduces stronger independencies
- ... and thus makes it impossible to represent distributions where there are no independencies

☞ **Constraining the structure of a BN reduces the space of representable distributions**

Re: “Overfitting Models are Complex”

Consider a simple example:

- World with three binary variables A, B, C
- Given training set $\mathcal{D}$:

	A	B	C
$\boldsymbol{x}_1$	1	0	0
$\boldsymbol{x}_2$	0	1	0
$\boldsymbol{x}_3$	0	0	1

Exercise:

- Construct a Bayesian network over A, B, C that assigns probability $1/3$ to each of the three examples $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3$ and 0.0 to any other event $\in \{0, 1\}^3$
- Can this be done without creating a fully connected network?

Bias and Variance

Goal:

- Avoid learning an overfitting model
- Force learner to generalise

Approach:

- Put constraints on class of models allowed to the learner:
- **Hard constraints:** Strictly restrict the class of models (e.g., only permit certain structures, limit number of parents, etc.)
- **Soft constraints:** Introduce an additional *regularisation term* to the objective function that adds a penalty for complex models.

Consequence:

- Models from a constrained model class are limited in how closely they can approximate the target distribution P^*
- Learned models will not be able to precisely describe the empirical distribution in the training data, and perhaps also the target distribution P^*

Error possibility introduced by restricting expressivity of model class is called Bias.

Bias and Variance

On the other hand:

- A large space of highly expressive (complex) models is more likely to contain a model $\tilde{\mathcal{M}}$ that closely approximates P^* (low bias)

But:

- Limited training set $\mathcal{D}$ may not be able to select the 'right' model among the large number of models in the hypothesis space
- Many candidate models will have similar likelihood relative to $\mathcal{D}$
- Small changes in $\mathcal{D}$ can radically change the properties of the selected model
- Running the learning algorithm several times with different training sets sampled from P^* will produce highly variable overfitting models.

Error possibility introduced by permitting high expressivity of model class is called Variance.

The Bias-Variance Tradeoff

Fundamental Tradeoff:

- A restriction to **simple models** makes hypothesis space smaller and increases the probability of **bias error**
- On the other hand, in a smaller hypothesis space, it is less likely to find an overfitting model.

vs.

- Permitting **complex models** reduces probability of bias error
- but introduces **variance** as a potential source of error:
- There may be many models that fit $\mathcal{D}$ *"by chance"* ...

$\Rightarrow$ **Tradeoff:** generalisation vs. overfitting, bias vs. variance
$\Rightarrow$ Task of system designer/experimenter:
find a good place in this continuum, relative to given problem (data set)
$\Rightarrow$ There is no theoretical decision guide
$\Rightarrow$ Much of Machine Learning is about this ...

Preview: Learning Tasks in Graphical Models

Different learning scenarios:

- **Parameter Learning** (given a fixed model structure)
  - Maximum likelihood parameter estimation
  - Bayesian parameter estimation
- **Structure Learning**
  (learning both the structure and the parameters of a model)

Additional level of difficulty: Learning from Incomplete Observations

In this class:

- Parameter learning in Bayesian networks (☞ Part 5.a)
- Structure learning in Bayesian networks (☞ Part 5.b)
- Learning from incomplete observations in a special class of graphical models: Hidden Markov Models (HMMs) (☞ Part 6.a)

What you should remember of this section

- The General Setting: Model Learning as an Optimisation Problem
- Objective Functions: Likelihood and Log-likelihood
- Generalisation vs. Overfitting
- Bias vs. Variance

Literature

Koller, Daphne and Friedman, Nir (2009).
Probabilistic Graphical Models: Principles and Techniques. Cambridge, MA: MIT Press.