

MLnet Summer School on Machine Learning
and Knowledge Acquisition:
LEARNING AND PROBABILITIES

Wray Buntine, RIACS
NASA Ames Research Center, MS 269-2
Moffet Field, CA 94035-1000
`wray@kronos.arc.nasa.gov`

Thanks to Padhraic Smyth and Peter Cheeseman for input.

Not all slides will be covered in the presentation.

OVERVIEW OF TUTORIAL

I. Introduction: 20 minutes

- motivation and context;
- a brief history

II. Foundations of learning: 50 minutes

- issues in learning;
- basic principles.

III. Learning representations and methods: 50 minutes

- probabilistic graphical models;
- mixture models.

VI. Specialist topics: 30 minutes

- missing values;
- knowledge discovery and refinement;
- connections between theories (wont be covered).

VII. Essentials: 5 minutes

- check list of representations;
- check list of methods;
- check list of theory;
- check list of fields.

SECTION Ia.

Introduction:

- **motivation and context:**

learning algorithms can be engineered from well understood principles, and the engineering process can be partly automated;

- a brief history.

SS #1

EXAMPLES: TEXT CATEGORIZATION

Text categorization: assigning documents to subject categories.

Example: Associated Press newswire with 100,000's short items in approx. 90 categories using some 11,000 words.

PRECIOUS METALS CLIMATE IMPROVING, SAYS MONTAGU
LONDON, April 1 – The climate for precious metals is improving with prices benefiting from renewed inflation fears and the switching of funds from dollar and stock markets ... Silver prices in March gained some 15 pct in dlr terms due to a weak dollar and silver is felt to be fairly cheap relative to gold ... The report said the firmness in oil prices was likely to continue in the short term ...
REUTER

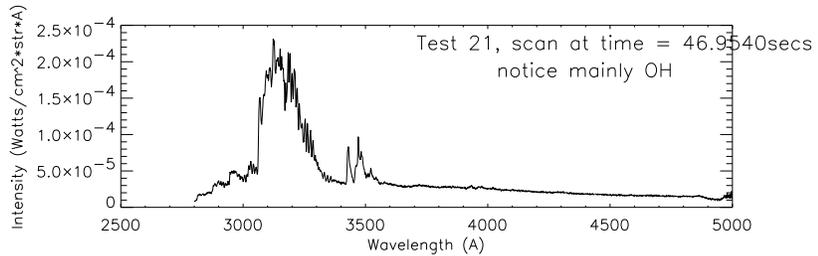
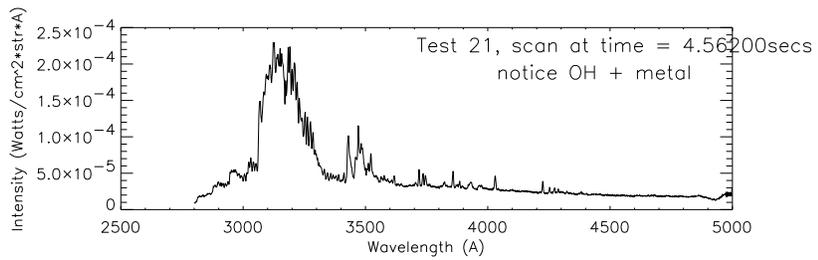
- Do you use a monolithic classifier with 11,000 inputs and 90 output classes?
- What subset of features do you use?
- How do you decompose the problem into sub-problems?

SS #1

IDENTIFYING METALS IN SPECTRA

Data: Scans are measurement of visible & near-infrared light from exhaust plume of rocket.

Task: for new Spectrum predict parts per billion (PPB) of different metals.



Spectrum = OH(temperature,pressure) + Metal(PPB,temperature,pressure) .

Unknowns: temperature, pressure, PPB (=parts per billion), and the OH model.
SS #1

LEARNING

Theme: Machine Learning \equiv intelligent software for data analysis

What: finding *models* in data

Why: to do prediction, to “understand”, to find “opportunities”

How: *search* for “good” models

Restrictions: not considering dynamic decision problems, e.g., control
only considering learning from random samples, although probability applies to other cases like learning from queries

SS #2

BACKGROUND FOR LEARNING SOFTWARE

Learning is often an embedded task :

- rarely more than 10% of the time is spent on learning!

Time on a project is spent :

- massaging the data, interfacing to other software and data sources,
- modeling, elicitation, refining the model,
- visualization, assessment,
- embedding the data analysis in the real task (diagnosis, etc.).

Current software tools for data analysis support :

Mathematical processing: Lapack, Matlab, IMSL.

Data manipulation and display: Perl, IDL, Khoros, Matlab, S-Plus.

Component integration: Khoros, Apple's Scientist's Workbench, TCL/Tk.

Specific learning tasks: Autoclass, S-plus, CART, Back-prop, etc.

SS #2

ON THE DESIGN OF ALGORITHMS

Data & Representation

LEARNING = + Prior knowledge

+ Learning principle

+ Search/Optimization

- Learning principle = Bayesian probability and decision theory.
- Many other principles (PAC in some cases, MDL, cross validation with empirical Bayes, etc.) differ mainly in philosophy and areas of applicability, not too much when implemented on the same problem.

SS #3

EXAMPLE DESIGNS

Algorithm	Derivation
basic backprop w. weight-decay	feed-forward nets + MAP+batch updates
learning Bayes nets	Bayes nets + exact Bayes factors + MAP + local search
class probability trees	trees + exact Bayes factors + MAP + local search
Autoclass (unsupervised learning)	Bayes net with one central latent variable + EM

- generative theory of learning rather than a descriptive theory
- works for many classes of learning problems except for local methods (Kernel density and nearest neighbor)

SS #4

DESIGN COMPONENTS

Principled design of data analysis software can be partly automated.

Specification of learning problems using graphical models.

Learning principle: Bayesian theory subsumes all useful theories.

That I'm aware of. On the other hand, non-Bayesian theories provide complementary perspectives and are based on different intuitions. No one theory is uniformly superior in all aspects.

Search/Optimization: broad algorithm schemas and strategies exist for many classes of problems.

SS #5

BAYESIAN THEORY: ASIDE

Probability	Usage	Field
frequencies	learning models (trees, networks) give frequencies for i.i.d. events	Orthodox statistics
belief	expert's opinions about one-off events	Decision analysis
belief about frequencies	expert's/user's opinions about learning models	Bayesian statistics

- Beliefs about frequencies called *second-order probabilities*. Notice they are first-order beliefs because frequencies are model parameters.
- Beliefs about frequencies *before* seeing any data are known as *prior* beliefs.
- Priors are a major source of contention. On real problem, reasonable priors can usually be developed, although this may be difficult.

SS #6

BASIC CONTEXT FOR DATA ANALYSIS

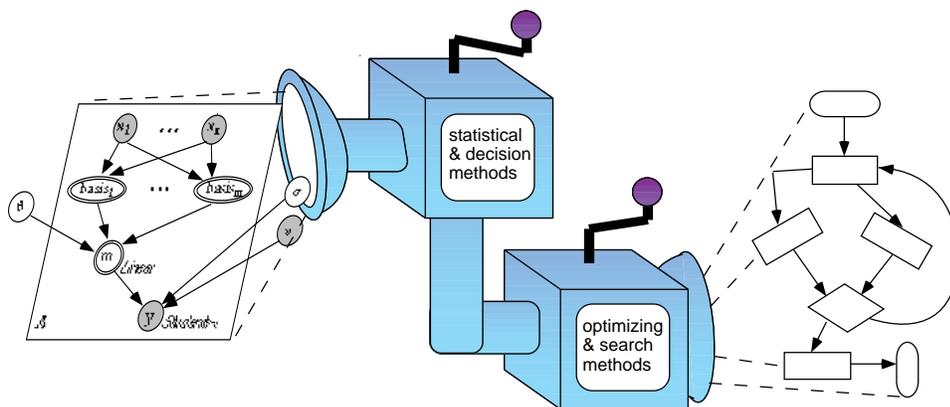
- In applications, no one algorithm (for trees, nets, linear regression, unsupervised learning, etc.) ever quite fits the task.
- Analysis of large data sets involves:
domain expertise + statistical knowledge + computational skills.
- Scientists/engineers rarely skilled in both statistics and computing.
- Our model:
 1. Scientist interacts with data analyst and designs prototype model.
 2. Data analyst encodes model and uses software generator and some fine-tuning to create software for IDL/C/parallel computer.
 3. Scientist uses software and discovers problems in model.
 4. Scientist (with help from data analyst) extends model and refines software as prototype develops.
- Require rapid prototyping of data analysis software.

SS #7

BASIC SOFTWARE GENERATOR

Input: graphical specification of the data analysis problem in terms of the basic inference model, prior knowledge, data, missing values, parallel computing specs., etc.

Output: optimized code and library calls to perform the learning/refinement or discovery for us, e.g., as a stand-alone program, Khoros or IDL module, or C* for the CM-5.



SS #8

ADVANTAGES OF SOFTWARE GENERATORS

- Software generators and software libraries place state-of-the-art technology in the hands of developers, e.g., TCL/Tk (window interfaces), Lapack (matrix manipulation), etc.
- Allow code to be embedded in larger, custom applications using a companies own data-base, graphics environments, etc.
- Allow fast prototyping of new applications.
- Is an ideal teaching aid: embodies key principles and methods in learning.
- Allow effective parallelization of algorithms from high-level specification. **NB.** Parallelization of source code is generally inefficient.
- We already generate data analysis software (of a kind): GLM [60], BUGS [36], feed-forward networks, etc.

SS #9

SECTION Ib.

Introduction:

- motivation;
- **a brief history:**

different theories and methods for learning have blossomed in the last three decades; each have their own strengths and weaknesses.

SS #10

BACKGROUND OF EARLY STATISTICS

late 1700's to late 1800's: mathematical physicists develop basic techniques, Laplace develops Bayesian methods (see [91]);

1890's on: non-physicists begin to develop statistics;

1920's to 1950's: **Orthodox Statistics** developed by Fisher, Neyman, etc.;

- need to overcome *ad hocery* in experimental sciences;
- standard recipes for common medical/sociological/biological questions;
- Bayesian statistics dormant, deemed “subjective” and by implication “poor scientific practice”;
- computers and computational methods primitive;

SS #11

THE TURNING POINT OF STATISTICS (1940-60s): LEARNING? HOW?

- Orthodox Statistics back then was **poor in methods and theory for learning** (because of inability to handle search, overfitting, etc.).
- Advent of **computers** mean computational methods for learning now feasible.
- **Theory of rationality** developed by Wald, Cox, de Finetti, etc.:
 - axioms of rationality [26, 47] imply computational inference decomposed into decisions and beliefs, i.e., Bayesian methods [3];
 - “non-rational” methods have inconsistencies (e.g., parts of Orthodox Statistics, see [4, Section 1.6]);
 - but most *practical* methods have a rational counterpart (e.g., fuzzy logic, one-sided hypothesis testing, MDL, etc.).
- Although essential to the rational theory, **priors are widely perceived as a problem**, and in some circles “unscientific”, because they can be *subjective*.

SS #12

MODERN LEARNING METHODS: FILLING THE VACUUM

Inductive Inference: asymptotic results for noise free learning, 1960-70’s (see [1]).
(asymptotic so ignore priors)

PAC, PAB, and Uniform Convergence: sample independent, worst-case, large-sample bounds, grew out of pattern recognition and computer science, late 1980’s on (see [40, 41, 28, 98, 43]). (large sample so ignore priors)

Statistical Physics: adapting mathematical techniques from statistical physics, late 1980’s on (see [86, 87]). (assume “truth” is known so no prior needed)

Stochastic Complexity, Minimum Description Length (MDL), etc.:
techniques from coding theory to learning, late 1970’s on [81, 99, 59]). (rename priors to be “code-lengths” and claim they are objective)

Bayesian Statistics: applies pure probability and decision theory to learning, focusing on constructive theory [5, 26, 92, 14, 54, 73, 8]. (accept priors as unavoidable in the smaller sample case)

SS #13

MORE MODERN LEARNING METHODS: FILLING THE VACUUM

Computational Learning Theory: generally contains a mix of the previous five; focuses on complexity rather than constructive theory (see annual COLT conferences).

Modern Orthodox Statistics: asymptotic, approximate large sample (order N), etc.; hypothesis testing, resampling schemes [16, 85, 30, 38, 18]. (stick to asymptotics, or treat priors as “complexity measures” or “smoothing terms”)

Pattern Recognition: Emphasis on applied problems and methods, nearest neighbor, Kernel methods, unsupervised methods [29, 37, 61]. (methods similar to modern statistics)

Neural Networks and Machine Learning: Emphasis on applied problems and methods, the former in numeric, incremental algorithms, the latter in symbolic.

SS #14

LEARNING THEORIES AND METHODS: SUMMARY

- Theories can be interpreted and compared with language of probability and decision theory [42, 12, 2].
- When addressing the same question, most practical theories say to do (in a crude sense) the same thing, but differ in their justification and philosophy. Most theories differ on how they address/side-step/ignore the problem of **priors**.
- More recently, experts are conversant with several disciplines and theories, i.e., the boundaries are disappearing but the complementary perspectives remain important.
- Each field has its own strengths and weaknesses. Good cross-field comparisons exist [80, 77, 22, 64, 101]
- Classification tree algorithms make good inter-disciplinary studies because of their widespread development [83, 25, 75, 76, 100, 11].

SS #15

SECTION IIa.

Foundations of learning:

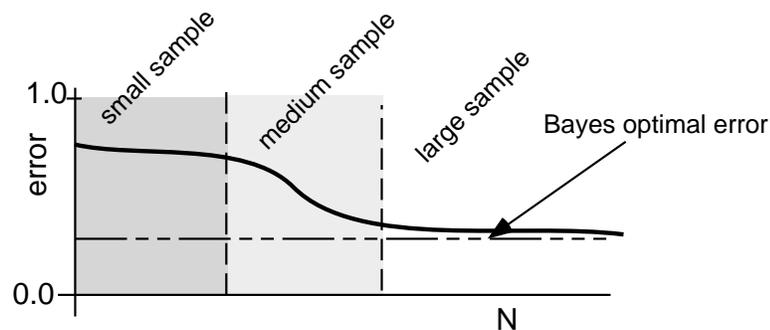
- **issues in learning;**

basic issues arise no matter what your representation, and we can understand them without recourse to theory;

- **basic principles.**

SS #16

LEARNING CURVES



Small sample: inadequate data, little better than random.

Medium sample: some data, along with prior knowledge yields reasonable results.
Typically occurs for $N = O(\text{number of parameters})$.

Large sample: “enough” data to yield near optimal results. Asymptotes to Bayes optimal error (lowest error achievable by any classifier) as $O(\frac{1}{N})$, or similar.

Where these occur is relative to the hypothesis space e.g., number of parameters, e.g., learning a tree of depth 2 requires much less data than a tree of depth 20.

SS #17

SUBSECTION: BASIC ISSUES IN LEARNING

The following set of examples illustrate key concepts in learning.

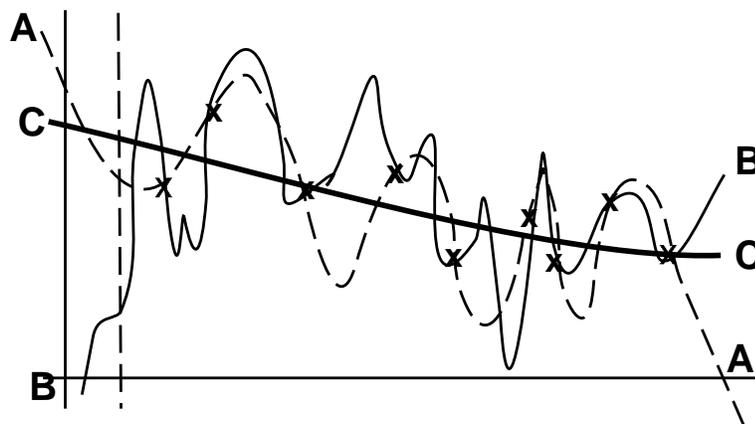
2-dimensional curve fitting is used because its easy to visualize.

The same concepts arise in any non-trivial learning problem: trees, rules, neural networks, regression, logistic regression, unsupervised learning, knowledge refinement, etc.

SS #18

EXAMPLE: SMOOTHING & OVERFITTING

Consider the plot given in the figure below. The x 's are data points. Curves A , B and C are potential fits.



SS #19

SMOOTHING, cont.

- Questions:
 - Which of the 3 curves do you prefer?
 - If you had to predict where the “true” curve crosses the vertical dotted line, where would you guess?
- The standard/popular view is that:
 - The best prediction would more or less follow the *smoother* curve C .
 - B is a poor curve because it takes too many unnecessary twists and turns. It **overfits** the data.

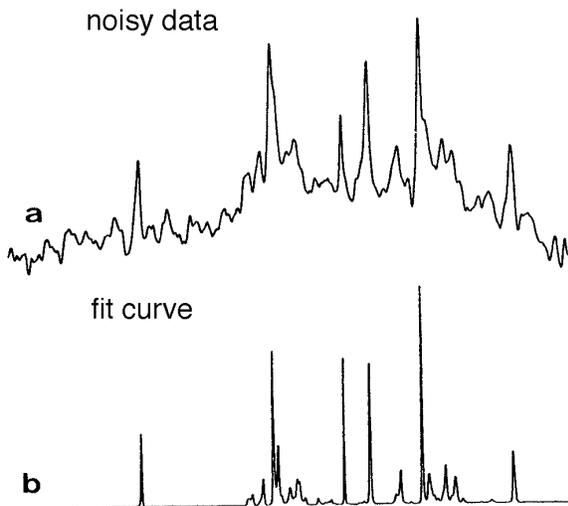
NOTE: we can change the representation (rescale X,Y axes) so that B becomes smooth and C becomes wiggly; the preference for the smoother curve must be *vocabulary dependent*.

- Choice of language/vocabulary is important in learning.

SS #20

EXAMPLE: LESS SMOOTHING

- Figure (a) shows some equally spaced data, a line of measurements with noise. What is the “good” underlying curve?
- Figure (b) shows a curve rated as “good” by the experts. It is *not* smoother because it is a line-spectrum.



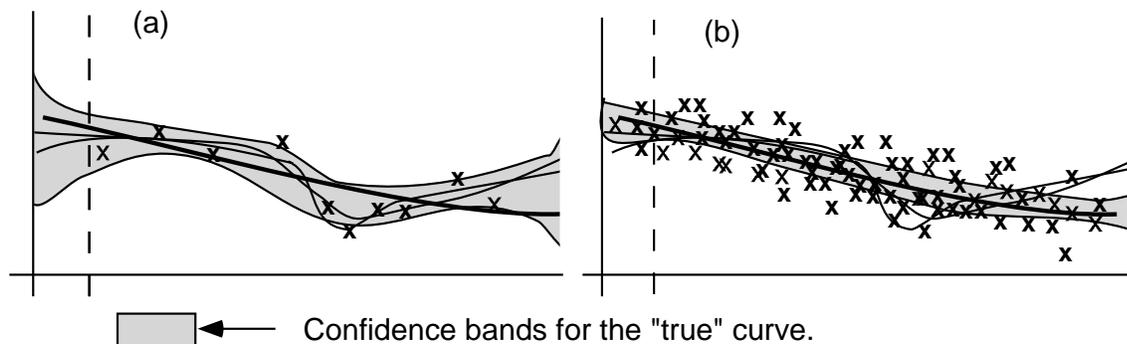
Smoother/simpler isn't inherently better.

In real problems, we have prior beliefs about the “truth”. These can and should effect the learning process.

SS #21

EXAMPLE: MODEL UNCERTAINTY

Consider the two plots given in the figure below. The x 's are data points. *Confidence bands* indicate the region wherein the “true” curve might reasonably lie.



Plot (b) represents the situation of plot (a) extended with a lot more data.

SS #22

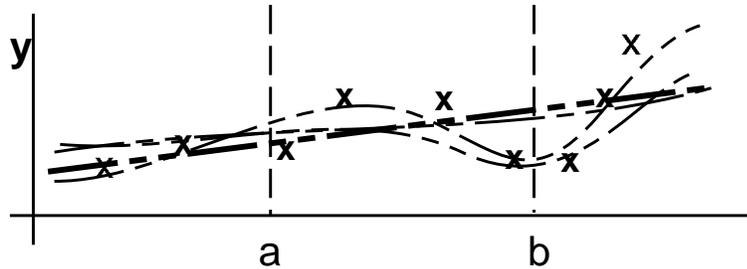
MODEL UNCERTAINTY, cont.

- Notice:
 - In plot (a) all three curves seem reasonable, and the confidence band is much broader.
 - In plot (b) only the thick curve now seems reasonable.
 - Notice many more curves seem reasonable for plot (a), and hence your uncertainty is much larger.
- In general, the number of different “reasonable fits” to data is an indication of uncertainty in the “best fit”, and any predictions made.
- **The less data there is, the more “reasonable fits” there are.**

SS #23

EXAMPLE: MULTIPLE MODELS

Consider the plot given in the figure below. The \mathbf{x} 's are data points. What do we say to the question: is $\hat{y}_a \leq \hat{y}_b$?



- If we look at the single “smooth best fit”, the answer is *yes*.
- If we look at several of the other “reasonable fits”, then a more accurate answer would be *usually*.
- **multiple models are useful when assessing questions of confidence and derived quantities.**

SS #24

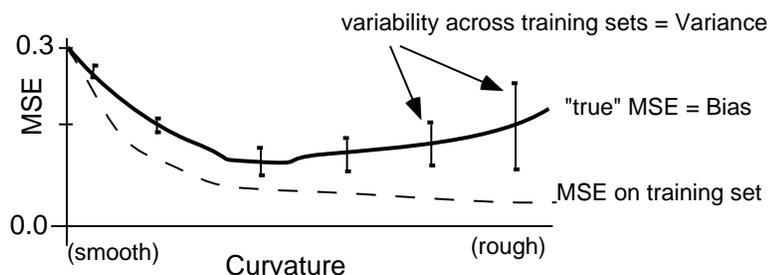
EXAMPLE: YOU HAVE INFINITE DATA!

(Or, you have “enough” data to be in the large sample case.)

- The data acts as an *oracle* for the “truth”.
 - What disease might this patient have? Select patients with similar symptoms and history and use them as a guide.
 - What model fits this problem best? Each model can be evaluated reliably by seeing how well it fits the data.
- Learning is now reduced to *search* through the space of models.
 - e.g. find the model with the highest sample likelihood, or best “fit” according to some other measure.
- This question is in general answered by PAC (Probably Approximately Correct), uniform convergence methods and their various generalizations.

SS #25

EXAMPLE: BIAS VERSUS VARIANCE



The graph shows the typical *error/complexity trade-off* for learning [33]. If we fit a curve with a given degree of curvature, C ($C = 0$ for a straight line) to our data:

- What is the *variance*?
 - What typical mean square error (MSE) can we get after learning?
 - How does this vary from training sample to sample of the same size?
- What is the *bias*?
 - What is the least MSE we could get for curves of curvature C ?
 - How does this differ from the least possible MSE (Bayes error)?

SS #26

BIAS VERSUS VARIANCE, cont.

- Variance and bias are in general negatively correlated.
 - There are “less” potential smoother curves to fit the same training sample.
 - Hence, the smoother the curve, the more confident we are that we have learned the “best” curve of that fixed curvature.
 - Equivalently, rougher curves have higher *variance* in mean-square error after training.
- Small sample learning implies that either high bias or high variance.

$$\text{Expected training error} = \text{bias} + \text{variance} .$$

- Picking an *appropriate* strong bias (i.e., an appropriate informative prior) is important for learning [33]. This is a complex issue [84].

SS #27

OBJECTIVITY vs. SUBJECTIVITY

”Are sardines packed in olive oil better for heart problems than sardines packed in natural fish oils?”

- You’re a consumer with a heart problem:
 - use whatever subjective opinions you trust to get the best results;
 - if you have a family history of heart problems, then be cautious;
 - subjectivity is preferred by the single agent!
- The Surgeon General wishes to make a claim, one of: *yes, no, maybe, we’re not sure yet*.
 - the Surgeon General needs to justify the claim to a broad range of people coming from different backgrounds;
 - the *consensus* of scientific opinion is that ...
 - objectivity is required here and can be implemented as: the same opinion is reached for a range of different people (priors).

SS #28

SUMMARY OF LESSONS

- With large samples, *learning is search* with the data as an oracle.
- Without much data, our expectations and biases, represented as *prior beliefs*, can and should effect the learning process.
- On any real problem I have seen, the domain expert *always* has some prior beliefs. Knowledge refinement is the norm.
- The less data there is, the more “reasonable fitting” models there are.
- Multiple models are useful when assessing questions of confidence and derived quantities.
- Occam’s razor (i.e., smoother/simpler models are nice) is relative to the vocabulary used.
- Objectivity/Subjectivity relative to the decision context.

SS #29

QUESTIONS FOR A THEORY OF LEARNING

1. Can learning *identify* the “truth” with finite/infinite data? Is this identification *exact* or does it *converge with probability one asymptotically*?
2. What *sample size* is required so that learning with high probability can achieve accuracy/cost within ϵ of the optimal?
3. What is a typical *learning curve*?
4. What is an *approximate/optimal algorithm* for learning with the current sample in hand, given the current *context*?
5. How *computationally complex* is this approximate/optimal algorithm.
6. What should be the *result* of learning? A set of alternatives?
7. How do you learn when the “truth” doesn’t appear in the hypothesis space?
8. How can learning *modify/refine* existing or previously learned knowledge?

SS #30

SECTION IIb.

Foundations of learning:

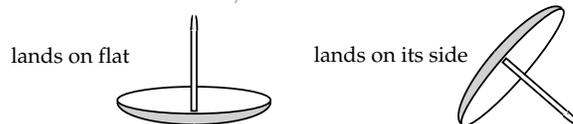
- issues in learning;
- **basic principles:**

Bayesian theory addresses each of the above issues, and shows how to do knowledge refinement; basic steps demonstrated here on simple problems.

SS #31

EXAMPLE, BERNOULLI

You have a thumb tack. When tossed, how does it land?



- θ = probability it lands on its flat; is probability as frequency.
- $p(\theta)$ = probability density as belief about frequency θ ; represents your prior on θ . What's your prior here?
- Suppose θ was probability that "a US citizen has at least 50% German descent." What's your prior $p(\theta)$?
- Suppose θ was probability that "an Australian citizen has at least 12.5% aboriginal descent." What's your prior $p(\theta)$?

SS #32

EXAMPLE, BERNOULLI

Model: Binomial, probability of success $p(r^i = 1) = \theta$, $0 \leq \theta \leq 1$. See [48].

Data: observe r successes in N trials.

Knowns: r , N .

Unknowns: θ , and success of new case s .

Action: prediction \hat{s} for new case.

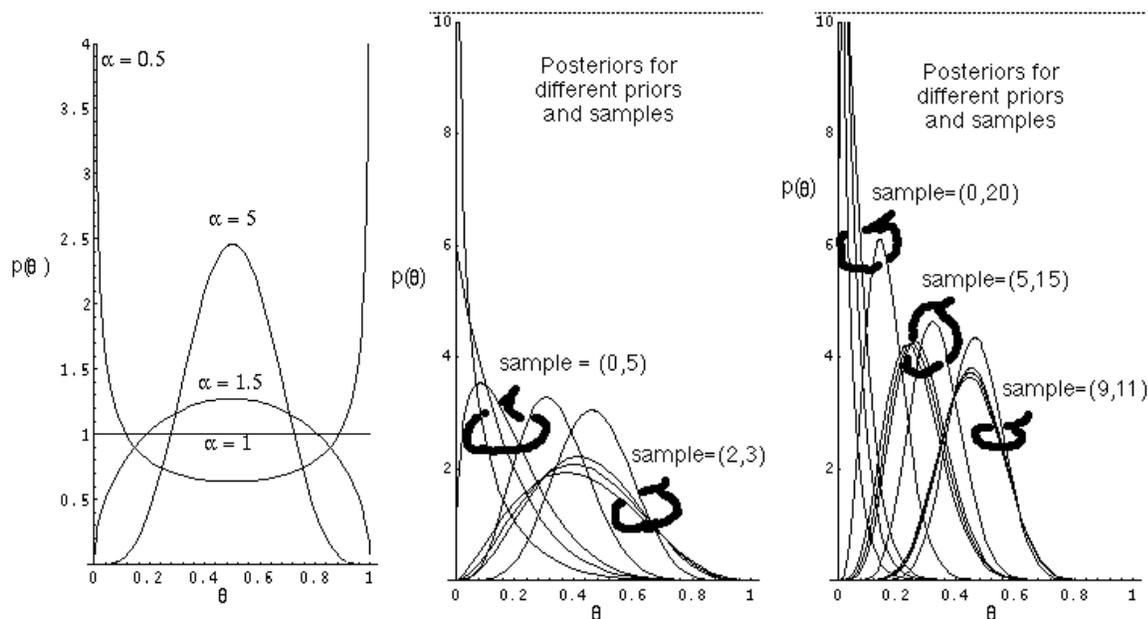
Utility: \$25 if prediction correct, $s = \hat{s}$; -\$25, otherwise.

Probability model:

$$p(\theta, r, N, s) = \begin{cases} p(\theta) \theta^r (1 - \theta)^{N-r} & \text{new case is success,} \\ p(\theta) (1 - \theta)^r (1 - \theta)^{N-r} & \text{new case is failure.} \end{cases}$$

SS #33

EXAMPLE, BERNOULLI, continued



SS #34

EXAMPLE, BERNOULLI, continued

Prior: assume θ is in the Beta(α, β) distribution:

$$p(\theta) = \frac{1}{\text{Beta}(\alpha, \beta)} \theta^{\alpha-1} (1-\theta)^{\beta-1}$$

Likelihood: as before.

Posterior: thus,

$$p(\theta|r, N) \propto p(r, N|\theta)p(\theta) = \frac{1}{\text{Beta}(r+\alpha, N-r+\beta)} \theta^{r+\alpha-1} (1-\theta)^{N-r+\beta-1}$$

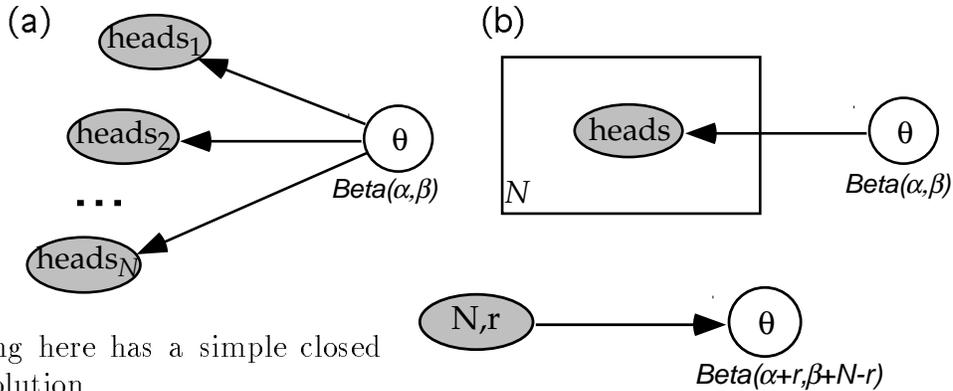
The MAP (maximum a posterior) value $\hat{\theta}$ is $\frac{r+\alpha-1}{N-r+\beta-1}$, and the posterior expected value of $\bar{\theta}$ is $\frac{r+\alpha}{N-r+\beta}$.

Best prediction: for minimum errors utility on binary choice, best prediction is success if $\bar{\theta} > 0.5$, and failure otherwise.

SS #35

EXAMPLE, BERNOULLI, continued

A Bayesian network for this problem is given in (a), assuming a $\text{Beta}(\alpha, \beta)$ prior. Equivalent form using “plates” is given in (b).



Learning here has a simple closed form solution.

This kind of simplification occurs for a well known class of problems, including simple Bayes classifiers.

SS #36

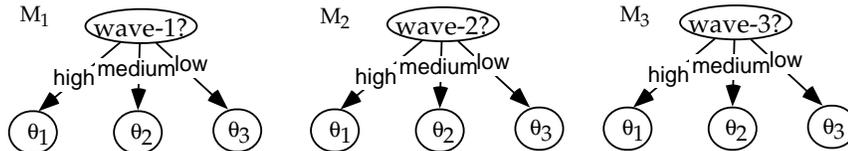
DECISION STUMPS AND BAYES NETS

Sample Wavelength Database

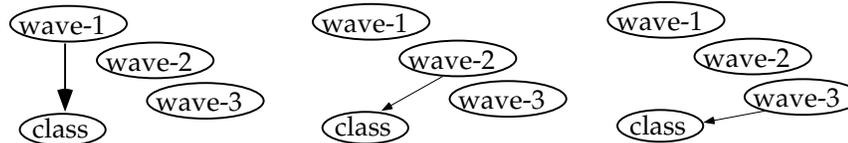
Wavelength-1	Wavelength-2	Wavelength-3	Class
High	Low	High	type A
Medium	Medium	High	type B
Low	Medium	Low	type A
...

Given uniform database with 3 variables $Wave-1, 2, 3$ and $Class$.

How might we learn decision stumps? i.e., choose between:



How might we learn Bayesian networks with one arc? i.e., choose between:



SS #37

LEARNING DECISION STUMPS, cont.

Model: Alternative set of decision stumps M_i with test on feature $Wave-i$ at root node, probability of $Class = type-A$ at the i -th leaf node is θ_i , $0 \leq \theta_i \leq 1$.

Data: observe N cases of $Wave$ and $Class$, denoted $Data = (Waves, Classes)$.

Knowns: $Data$, and a new case $Wave'$.

Unknowns: which decision stump $Stump \in \{M_1, M_2, M_3\}$, leaf probabilities $\theta_1, \theta_2, \theta_3$, and type of new case $Class' \in \{type-A, type-B\}$.

Action: prediction \widehat{Class} for new case.

Utility: \$25 if prediction correct, -\$25, otherwise.

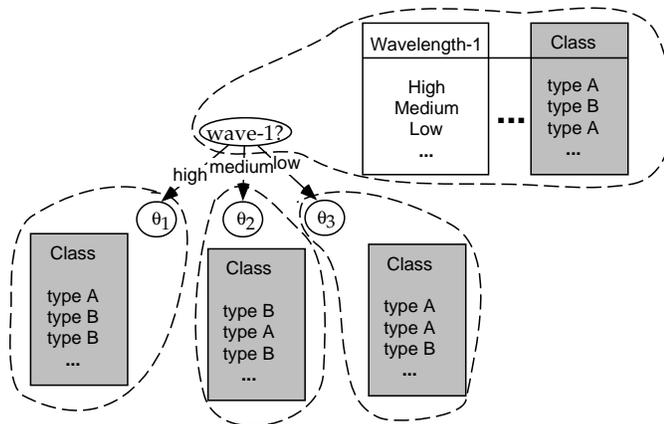
Probability model:

$$\begin{aligned}
 p(\theta, Data, Wave', Class', Stump|M) = & \\
 & p(Stump|M) p(\theta|Stump, M) p(Waves, Wave'|M) \\
 & p(Classes|Waves, \theta, Stump, M) p(Class'|Wave', \theta, Stump, M)
 \end{aligned}$$

SS #38

LEARNING DECISION STUMPS, cont.

If we knew which decision stump where “true”, we would have a Bernoulli sampling problem at each leaf.



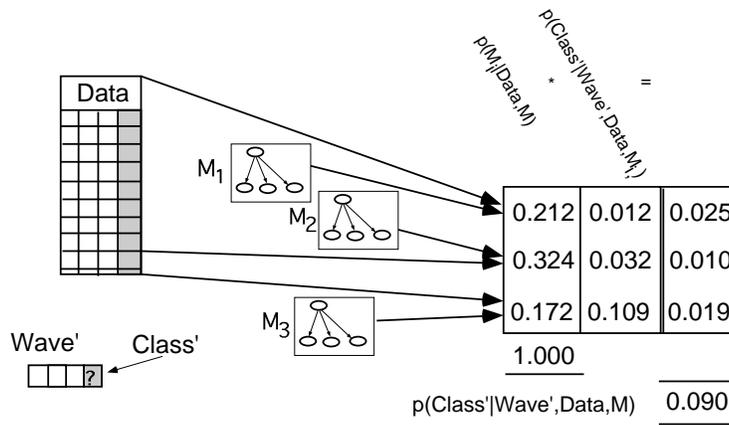
In this case, the problem is solved by the above methods. Find the leaf that the new case belongs in, and then apply the Bernoulli sampling theory at that leaf [17, 11]. But, which decision stump is “true?”

SS #39

LEARNING DECISION STUMPS, cont.

Since we don't know which decision stump is "true" we have to average over the classification probabilities from each of the 3 possible decision stumps.

$$p(Class'|Wave', Data, M) = \sum_{i=1}^3 p(Class'|Wave', Data, M_i)p(M_i|Data, M)$$



SS #40

LEARNING DECISION STUMPS, cont.

$$p(Class'|Wave', Data, M) = \sum_{i=1}^3 p(Class'|Wave', Data, M_i)p(M_i|Data, M)$$

where $p(Class'|Wave', Data, M_i)$ is the class probability calculated assuming we know the "true" decision stump is M_i .

$$p(M_i|Data, M) = \frac{p(Data, M_i|M)}{\sum_{i=1}^3 p(Data, M_i|M)} = \frac{p(M_i|M)p(Data|M_i, M)}{\sum_{i=1}^3 p(M_i|M)p(Data|M_i, M)}$$

$p(Data|M_i, M)$ is the *evidence* for model M_i :

$$p(Data|M_i, M) = \int_{\theta_1, \theta_2, \theta_3} p(Data|\theta_1, \theta_2, \theta_3, M_i, M) p(\theta_1, \theta_2, \theta_3|M_i, M) d(\theta_1, \theta_2, \theta_3) .$$

This integral can be calculated exactly and is a ratio of Gamma functions [11].

SS #41

LEARNING DECISION STUMPS, cont.

- Which prior shall we use for the Bernoulli problems at each leaf node?
- Which prior shall we use for each tree stump, $p(M_i|M)$?
- We can approximate:

$$p(\text{Class}'|\text{Wave}', \text{Data}, M) \approx p(\text{Class}'|\text{Wave}', \text{Data}, \widehat{M}_i, M)$$

for \widehat{M}_i the MAP model M_i chosen to maximize $p(M_i|\text{Data}, M)$. This will be good as long as M_i is significantly better than the other models.

- How do we scale this for decision trees of arbitrary depth?

This general approach scales to provide a competitive algorithm for learning classification trees. A lot of this detail appears in [11].

SS #42

LEARNING DECISION STUMPS, cont.

- Bayesian methods automatically lead to *averaging over multiple models* (e.g., see Perrone's workshop at NIPS'93 and [103]). This automatically accounts for overfitting, does smoothing, etc.
- Alternatively, the MAP approximation leads to *model selection*, choosing the "best" model [21].
- Same principles extend for selecting
 - the number of classes in unsupervised learning [20],
 - the right sized neural networks [93],
 - "good" Bayesian networks learned from data [13, 23].
- How do we set priors? In general, this requires care but reasonable defaults exist in some cases, for instance, for trees [11].
- Learning Bayesian networks from data is analogous to learning class probability trees [10].

SS #43

SUMMARY OUTLINE: BELIEF

Bayesian probability theory:

- Write down the variables in the domain, keeping knowns, K , unknowns, U , and actions A , separate.
 - The learning sample will be known,
 - the prediction for the class of the new case will be an action,
 - the parameters of your model, and the “true” class of the new case will be unknown.
- Write down a probabilistic model for all the variables U and K , $p(U, K)$.
- Construct $p(U|K)$ to represent what you’ve learned about U from knowing K .

SS #44

SUMMARY OUTLINE: ACTIONS

Bayesian decision theory:

- Write down a value/cost/loss/utility function to express the goal of learning, $u(K, U, A)$.
- Make the action A which will maximize the utility/value (or minimize cost/loss).

$$A = \operatorname{Argmax}_A \mathcal{E}_{U|K} (u(K, U, A)) = \operatorname{Argmax}_A \int_U p(U|K) u(K, U, A) dU . \quad (1)$$

SS #45

SECTION IIIa.

Learning representations and methods:

- **probabilistic graphical models;**

graphical models offer a unified way to model and represent learning problems; methods exist for generating learning algorithms from these graphical models [14, 36];

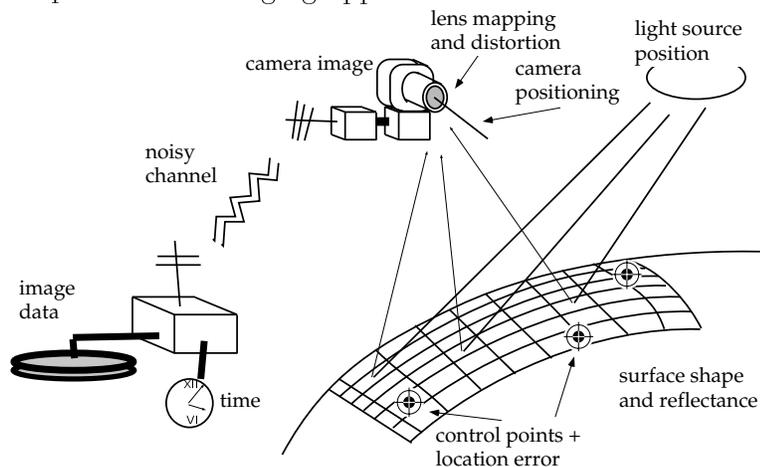
- **mixture models;**

SS #46

AN IMAGE PROBLEM

For our purposes: A *model* is a mathematical description of the processes generating the data, including all sources of uncertainty. It is by necessity an *assumption* since truth is rarely that simple.

A pictorial description of an imaging application:



SS #47

THE IMAGE MODEL

A model would be mathematical equations relating the various knowns and unknowns of the problem:

Knowns: image data, time, light source position, camera positioning, precise location of control points, lens distortion function, control point matching function.

Unknowns: camera positioning error, bit noise from noisy channel, relative location of control points in image, “true” reflectance of each point on surface.

Equations: Mapping unknowns and knowns probabilistically:

- image data = camera image data + bit noise from noisy channel
- bit noise = random flipping of bits with probability ϵ
- camera image data = convolution(“true” reflectance, lens mapping)
- location of control points in image =

Model yields a *joint probability distribution* for all known and unknown variables.

SS #48

ON MODELS

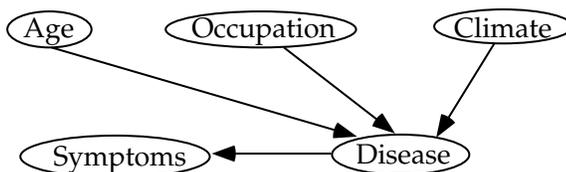
- Model is an *assumption* about the domain.
- Model yields a joint probability distribution for all known and unknown variables.
- Model often based on causal understanding of the domain.
- Model includes all noise processes and sources of uncertainty.
- Model may make approximations in order to be computationally tractible.
- A decision tree is **not** a model but a set of instructions on how to make predictions (it has no probabilities). A class probability tree can be a model. Likewise, a feed-forward neural network needs an additional, explicit error model to become a model in our sense (e.g., Gaussian error).
- Probabilistic graphs such as Bayesian networks, Markov networks, and chain graphs are a language for expressing models. They are rather like data flow graphs.

SS #49

INTRODUCTION TO BAYESIAN NETS

Bayesian networks on boolean variables, consist of a structure and its associated conditional probability tables. For tutorials see [89, 19, 44].

Structure, S : represented by a Directed Acyclic Graphs (DAGs) like below.



Conditional probability tables, θ : represented as below.

$Age < 45$	0.46
$Age \leq 45$	0.31

$p(Age)$

	<i>Disease</i>	
<i>Symptoms</i>	stomach ulcer	angina
stomach pain	0.23	0.17
chest pain	0.31	0.45

$p(Symptoms|Disease)$

SS #50

INTRODUCTION TO BAYESIAN NETWORKS, cont

- The Bayesian network (S) alone implies the joint probability density of the variables takes the form:

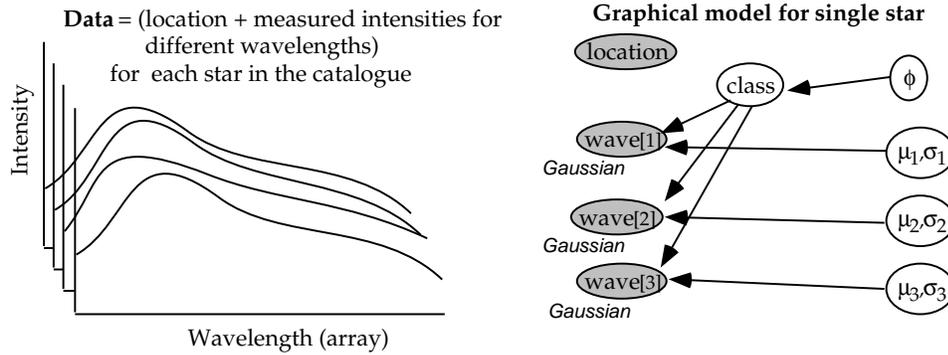
$$p(Age, Occ, Clim, Dis, Symp) = p(Age)p(Occ)p(Clim)p(Dis|Age, Occ, Clim)p(Symp|Dis).$$

- Functional forms (Gaussian, Poisson, etc.) or tables associated with the network give the probability distribution at each node.
- Algorithms for manipulating these models are well developed (see recent UAI conferences).

SS #51

THE BAYESIAN NETWORK FOR AUTOCLASS III

The Autoclass III [20] application to the IRAS star catalogue looked at intensities measured for different wavelengths in the infra-red region.

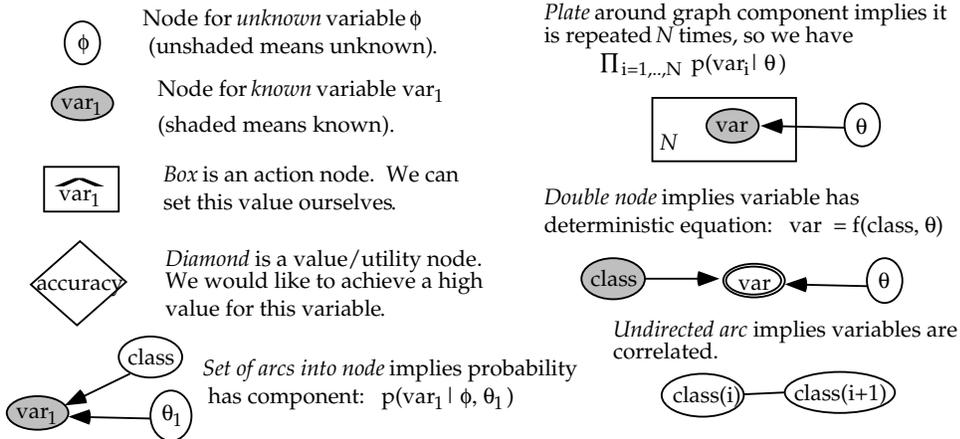


This simple model assumes independence between intensities. A more complete model would assume correlation between neighboring wavelengths, and dependence between the hidden class and the star position.

SS #52

LEGEND FOR GRAPHICAL MODELS

For graphical models in general:

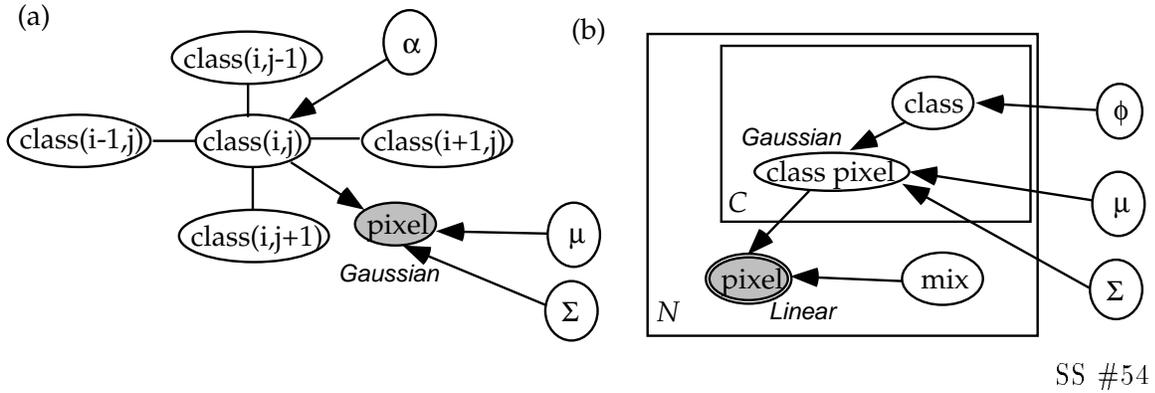


SS #53

UNSUPERVISED LEARNING FOR IMAGES

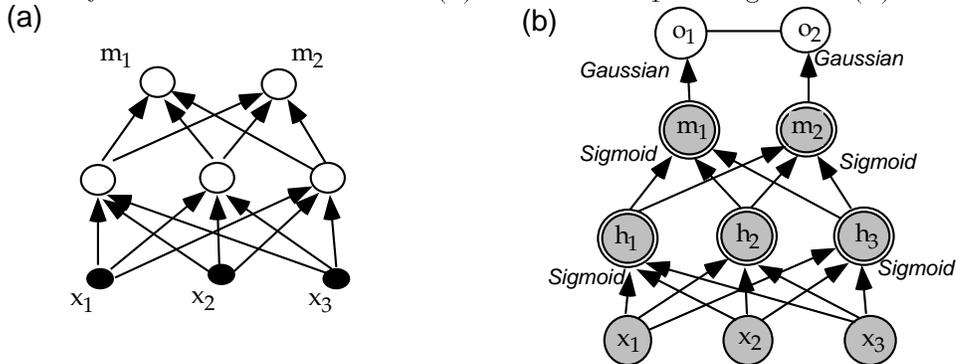
Spatial coherence models: Figure (a) shows a popular Markov model used for creating spatially coherent classifications [6].

Mixed pixels: Pixels in an image represent a mixture of class types (30% grassland, 70% woodland); figure (b) shows the popular linear mixture model used for this.



FEED-FORWARD NETWORKS (FFNNS)

A 3-hidden layer feed-forward network (a) and a corresponding DAG (b).



- The neural nodes are represented as deterministic nodes (double circles).
- A output layer denotes the response (or output) variables being predicted, o_1, o_2 .
- The inputs are shaded indicating their value is known.
- Both network predictions (m_i) and the actual responses (o_i) are represented. The DAG also models the error as a *Gaussian*.

GRAPHICAL MODELS

Other problems can be represented in graphical models using:

- known and unknown variables,
- deterministic nodes [88],
- standard probabilities functions at nodes (Gaussian, multinomial, Dirichlet, logistic, etc.),
- mixed directed and undirected arcs [32],
- optional arcs (indicating alternative models) [14].
- plates representing samples [14].

Graphical models offer a unified framework for representing a problem (prior knowledge and data), performing problem decomposition, specifying a knowledge refinement task, etc.

SS #56

GRAPHICAL MODELS, cont.

Probabilistic graphical models can represent models for different applications and from different disciplines:

Connectionism: stochastic Hopfield networks, feed-forward networks, mixtures of experts.

Artificial intelligence: (machine learning and knowledge discovery) Bayesian networks for expert systems, dynamic models for planning and control, some rule-based systems.

Statistics and Pattern Recognition: Hidden Markov models, Kalman filters, Generalized linear models and additive models, i.e., linear regression.

SS #57

SECTION IIIb.

Learning representations and methods:

- probabilistic graphical models;
- **mixture models;**

mixture models and well understood algorithms for handling them are ubiquitous in learning;

SS #58

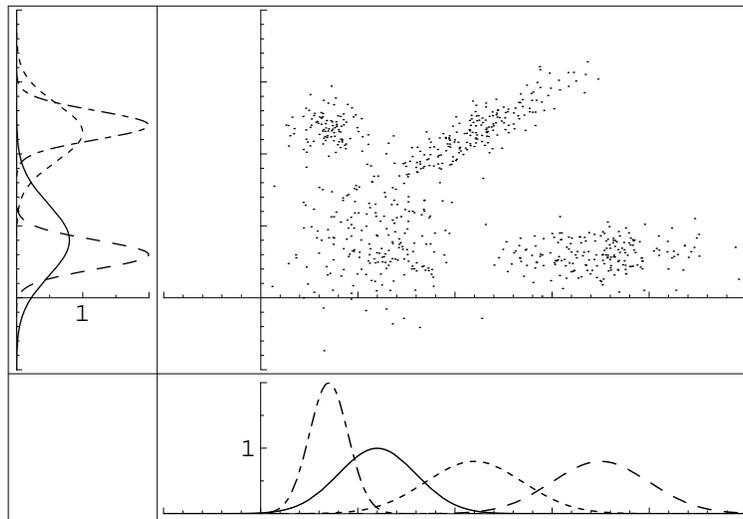
MIXTURE MODELS: EXAMPLE

This sample is a *mixture* of 4 2-dimensional Gaussians. The marginal distributions for each Gaussian is shown on the axes.

Observed data: is N pairs of real values $x_i = (x_i^1, x_i^2)$.

hidden variable: is the number $h = 1, 2, 3, 4$ indicating which Gaussian a point comes from.

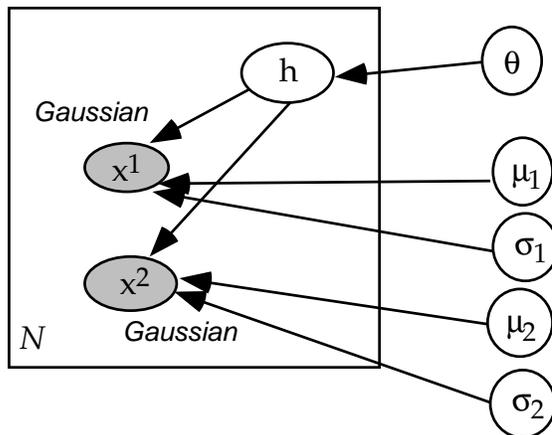
$\theta, \mu, \sigma:$ h frequency and means, variances of 4 Gaussians.



SS #59

MIXTURE MODELS: EXAMPLE, cont.

The graphical model representing this learning problem, assuming Gaussians independent between x^1 and x^2 , is given below.



SS #60

MIXTURE MODELS: MOTIVATION

Mixture models are **ubiquitous** in data analysis [96, 62]. They model:

- Missing values in other problems (trees, feed-forward networks, etc.) [15, 92].
- Latent or hidden variables, e.g., medical *syndromes*.
- Unsupervised learning and clustering, e.g., Autoclass [20].
- Supervised learning and multivariate splits in trees [53].
- Robust regression [57].
- Non-parametric density estimation (i.e., equivalent to Kernel density estimation and nearest neighbor).
- Rule-based systems with multi-firing probabilistic rules.
- Related to hidden Markov models.

SS #61

MIXTURE MODELS

- Data has observed variables x , and *hidden/latent* (unobserved) variable h .
- Likelihood for x, h is parametric model M with parameters θ :

$$p(x, h|\theta, M) = f_M(x, h, \theta) .$$

for some simple parametric distribution f_M . e.g., multivariate Gaussian, Bayesian network with known structure, etc.

- The likelihood for the observed data (for discrete h) is

$$p(x|\theta, M) = \sum_h f_M(x, h, \theta) .$$

- This likelihood is used by most learning theories, maximum likelihood and applied statistical methods, Bayesian methods, minimum description length, etc., when devising an algorithm.

SS #62

ALGORITHMS ON MIXTURES

Many general purpose algorithms exist for learning with mixtures [96, 92]. Incremental versions of each of these algorithms also exist (e.g., [95]). These are:

1. k -means clustering and related algorithms [29],
2. The Expectation-Maximization (EM) algorithm [27].
3. Gibbs sampling [49, 35], and more general purpose Markov chain Monte Carlo algorithms [39, 68, 79].

These are listed in terms of:

- increasing computational complexity,
- increasing statistical sophistication,
- increasing accuracy,
- decreasing bias.

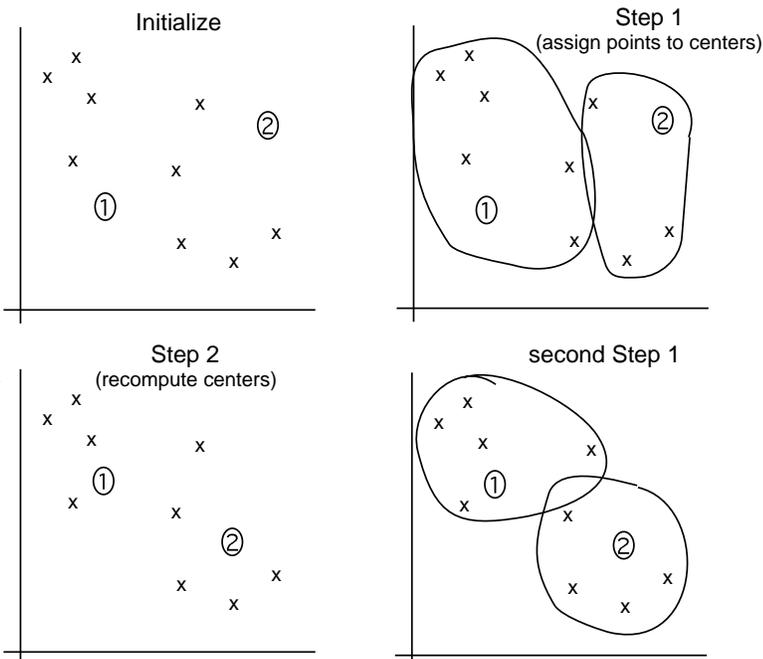
SS #63

K-MEANS ALGORITHM

Initialize: class centers.

Repeat: until converges,

1. Assign cases to their most likely class.
2. Recompute class (h_i) centers to their maximum *a posteriori*.



SS #64

K-MEANS ALGORITHM: ANALYSIS

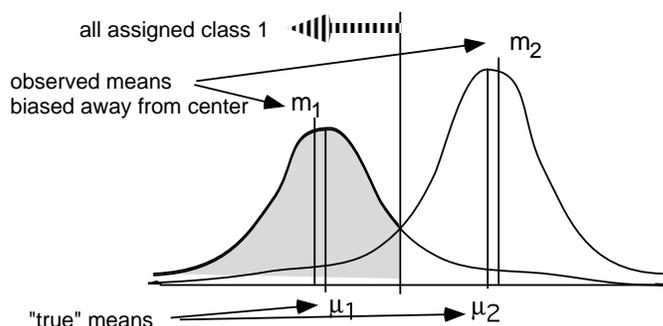
- Step 1 assigns classes h_i for $i = 1, \dots, N$ to maximize

$$p(\theta, x_1, h_1, \dots, x_N, h_N, M) \propto \prod_{i=1}^N p(h_i | x_i, \theta, M) .$$

- Step 2 assigns θ to maximize

$$p(\theta, x_1, h_1, \dots, x_N, h_N, M) \propto p(\theta | x_1, h_1, \dots, x_N, h_N, M) .$$

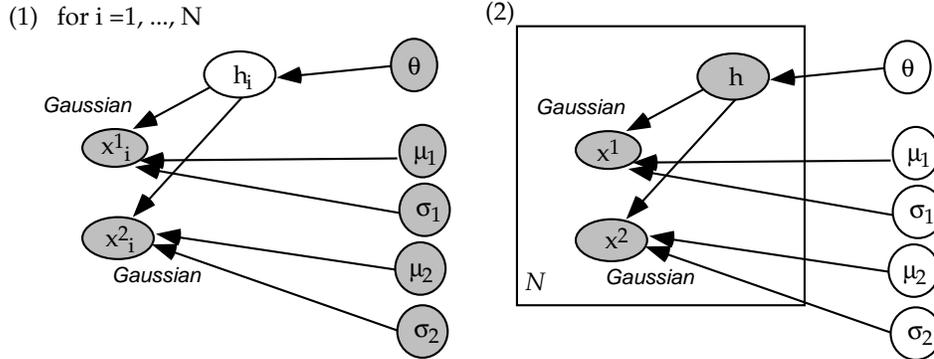
- On convergence, hidden variables h_i and θ are at a joint maximum. This is known to be biased, i.e., estimates for θ differ on average from the “truth”.



SS #65

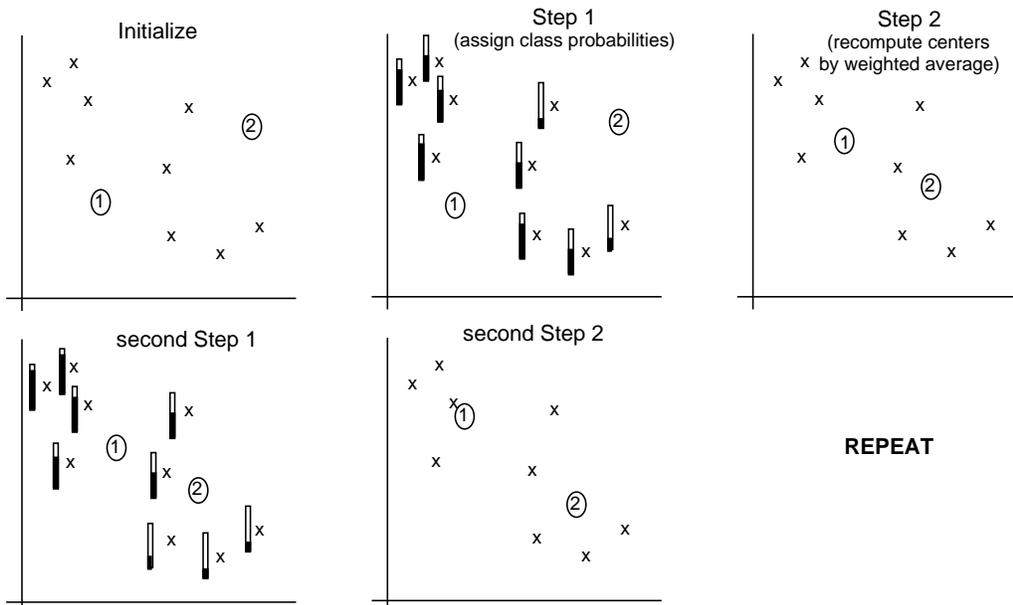
GRAPHICAL MODEL

The graphical model representing this and the subsequent two algorithms is given below.



SS #66

EM ALGORITHM



SS #67

EM ALGORITHM: ANALYSIS

- Adds to the k -means algorithm by making probabilistic assignment to hidden classes.

Intuition: since we don't know the hidden classes, assigning them a single fixed class is biasing us, especially if they are borderline,

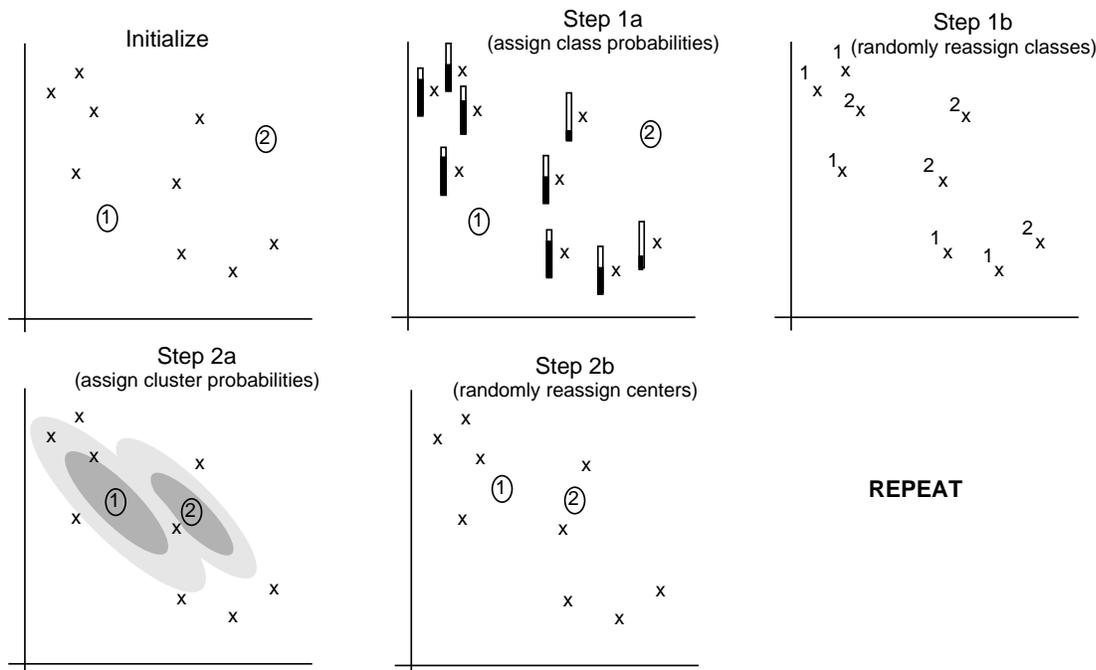
- Analysis is more complex than k -means but EM is asymptotically unbiased. It computes the MAP for θ .

$$p(\theta|x_1, \dots, x_N, M) .$$

- Convergence slow near a local maxima so some implementations switch to conjugate gradient or other methods [63] when near solution.
- To understand the general approach, you need to consider the *exponential family* of distributions [16, 14], although it applies more generally (e.g., [53]).

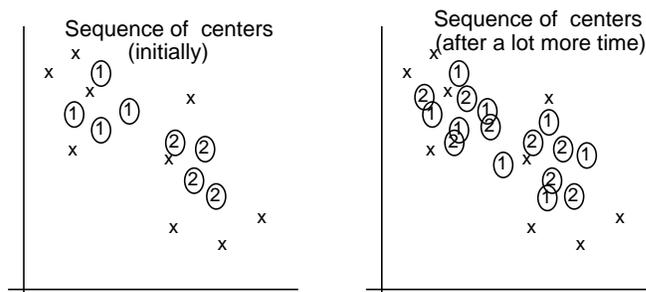
SS #68

GIBBS ALGORITHM



SS #69

GIBBS ALGORITHM: INTERPRETATION



- There is nothing inherent in the problem that says the top left cluster must be cluster 1, and the bottom right cluster 2.
- After a long time, labeling will be identical.
- This is because the problem has *symmetry*.
- Happens with any hidden/latent variable problem, but **not** missing values.
- Use symmetry breaking to handle this.

SS #70

GIBBS ALGORITHM: ANALYSIS

- Step 1 samples each h_i for $i = 1, \dots, N$ from the distribution

$$p(h_1, \dots, h_N | x_1, \dots, x_N, \theta, M) \propto \prod_{i=1}^N p(h_i | x_i, \theta, M) .$$

- Step 2 samples θ from the distribution

$$p(\theta | x_1, h_1, \dots, x_N, h_N, M) .$$

- Sampling generates *dependent* samples of θ that nevertheless have large sample properties similar to the posterior distribution $p(\theta | x_1, \dots, x_N, M)$ [68, 79].
- Gibbs sampling corresponds to simulated annealing with fixed temperature [97].
- Local repair/search methods [52, 65] correspond to simulated annealing with zero temperature, so Gibbs sampling is *probabilistic local search*.
- Gibbs sampling can be applied more generally to *many* learning and data analysis problems [35].

SS #71

K-MEANS vs. EM vs. GIBBS

The three algorithms differ in how they perform Steps 1 and 2.

Algorithm	Step 1	Step 2
<i>k</i> -means	MAP or ML	MAP or ML
EM	posterior mean	MAP or mean
Gibbs	sample	sample

SS #72

SECTION VIa.

Specialist topics:

- **missing values;**

well known methods for addressing missing values exist;

- knowledge discovery and refinement;
- connections to other theory.

SS #73

MISSING VALUES

The problem of **missing values**, or **unknown values**, **incomplete data** is endemic in data analysis. Here's the situation for supervised learning.

Sample Medical Records Database

ID	Temperature (F)	Age	Weight	Blood Pressure	Sex	Blood Test	Diagnosis
35267	?	25	180	?	Male	?	flu A
83877	98.1	62	132	145	Female	negative	flu B
23414	99.4	44	211	161	Male	?	healthy
09372	103.2	?	141	131	Female	negative	flu A
62253	97.1	58	130	141	Female	negative	healthy
07631	101.1	75	178	153	Female	positive	?
..
..

- Missing values are denoted by “?”.
- Some fields (rows) and some cases (columns) may be free of missing values.
- Missing values in the diagnosis column can, surprisingly, be useful (in supervised mixture models).

SS #74

OTHER TYPES OF MISSING VALUES

There are (at least) two other types of missing values (in classification) we will ignore.

Don't care: In supplying the data, the expert has not supplied these values because they believe them irrelevant for this case.

- The missing value corresponds to *irrelevance information*.
- Implies something about the form of the “true” concept.

Informative Missing: The value is missing but its absence represents important information. **e.g.** a missing telephone number may indicate the person has no telephone.

- The missing value could correspond to an additional field value. **e.g.** *Telephone = maybe-none*.
- In general the nature of the process causing the missing value needs to be modeled, where it is relevant for classification.

SS #75

BASIC OUTLINE OF MISSING VALUES

- Assume the missing value was not collected when the case was observed. Collection or not *independent* of the case's classification.
- In Bayesian, maximum likelihood, applied statistical, MDL, and many other learning theories, analysis of missing values is the same.

Why? they all have a common model of the data likelihood at their core.

- In general, missing data is handled with *mixture models* [96].
- We have a case with class *class*, known data, *given-fields*, and missing data *missing-fields*.

$$fields = (given-fields, missing-fields) .$$

- How do we modify the learning algorithm? For most classes of algorithms, well understood alternatives exist [95, 74, 15, 90].

SS #76

MISSING VALUES IN DISCRIMINATIVE LEARNING

- Discriminative classification models provide a direct model.

$$p(class|fields, Model) = \text{probability of } class \text{ given inputs} .$$

where *Model* could be “a class probability tree”, or “a feed-forward network”, or “linear regression”.

- The conditional likelihood of a single case with missing data becomes:

$$p(class|given-fields, Model) = \sum_{missing-fields} p(class|fields, Model) p(missing-fields|given-fields)$$

- We need to augment the discriminative model with *Another-Model*, a model predicting what values the *missing – fields* might take. This can be learned too.

$$p(missing-fields|given-fields, Another-Model) .$$

SS #77

EXAMPLE: DISCRIMINATIVE METHODS

ID	Temperature (F)	Age	Weight	Blood Pressure	Sex	Blood Test	Diagnosis
35267	99.0?	25	180	150?	Male	negative?	flu A
83877	98.1	62	132	145	Female	negative	flu B
23414	99.4	44	211	161	Male	negative?	healthy
09372	103.2	50?	141	131	Female	negative	flu A
62253	97.1	58	130	141	Female	negative	healthy
07634	101.1	75	170	155	Female	positive	?
..
..

Requires models of how to predict/sample/fill-in missing values from the other values:

- Model F_1 shows how to predict Blood Test result for first case $ID = 35267$.
- Model F_2 shows how to predict Blood pressure for first case $ID = 35267$, etc.
- The case with unknown diagnosis is ignored without loss of information. **Why?** it doesn't occur in the conditional likelihood for the sample.

SS #78

MISSING VALUES IN GENERATIVE LEARNING

- Generative classification models provide an indirect model of the full data set,

$$p(\text{class}, \text{fields} | \text{Model}) = \text{probability of full case } (\text{class}, \text{fields}).$$

where Model could be simple or “idiot’s” Bayesian classifier.

- Classification is done using Bayes theorem to compute $p(\text{class} | \text{fields}, \text{Model})$ from $p(\text{class}, \text{fields} | \text{Model})$.
- The likelihood of a single case becomes:

$$p(\text{class}, \text{given-fields} | \text{Model}) = \sum_{\text{missing-fields}} p(\text{class}, \text{fields} | \text{Model})$$

This sum makes many standard algorithms unuseable in their direct form.

- The case with unknown diagnosis can now be used because it provides some information about the unknown Model . **Why?** It does occur in the conditional likelihood for the sample.

SS #79

EXAMPLE: GENERATIVE METHODS

ID	Temperature (F)	Age	Weight	Blood Pressure	Sex	Blood Test	Diagnosis
35267	99.0?	25	180	150?	Male	negative?	flu A
83877	98.1	62	132	145	Female	negative	flu B
23414	99.4	44	211	161	Male	negative?	healthy
09372	103.2	50?	141	131	Female	negative	flu A
62253	97.1	58	130	141	Female	negative	healthy
07631	101.1	75	178	153	Female	positive	healthy?
..
..

Uses existing model of the data to predict/sample/fill-in what the missing values might be. But, we have a Catch-22:

- We need to have learned the model *Model* to predict/sample/fill-in.
- We need to predict/sample/fill-in the missing data to learn the model *Model*.

SS #80

METHODS: IGNORE THEM

Sample Medical Records Database

ID	Temperature (F)	Age	Weight	Blood Pressure	Sex	Blood Test	Diagnosis
35267	?	25	180	?	Male	?	flu A
83877	98.1	62	132	145	Female	negative	flu B
23414	99.4	44	211	161	Male	?	healthy
09372	103.2	?	141	131	Female	negative	flu A
62253	97.1	58	130	141	Female	negative	healthy
07631	101.1	75	178	153	Female	positive	?
..
..

- Ignore the missing values, so leads to inefficient use of data.
- Some problems have many missing values, so we get nothing to learn on!
- But is simple and computationally efficient. Use as an *initialization routine* for more complex algorithms.

SS #81

DISCRIMINATIVE METHODS: FILL-IN

- For the discriminative model, e.g. trees, networks, fill in the missing value with its most likely posterior value (or something similar, e.g., *sample* according to posterior probability).

$$\widehat{missing-fields} = \text{Argmax}_{missing-fields} p(missing-fields|given-fields) .$$

- Then we approximate the likelihood for the case by that with the filled in value.
- This gives us a set of complete data with:

$$\begin{aligned} p(class|given-fields, Model) \\ &= \sum_{missing-fields} p(class|fields, Model) p(missing-fields|given-fields) \\ &\approx p(class|\widehat{missing-fields}, given-fields, Model) . \end{aligned}$$

- Leads to biased results if fill-in is done deterministically. Can sometimes be unbiased if fill-in is done probabilistically, i.e., as per usual mixture models, EM, Gibbs, etc.

SS #82

GENERATIVE METHODS: FILL-IN

- For the generative model,

$$\begin{aligned} \widehat{missing-fields} &= \text{Argmax}_{missing-fields} p(class, missing-fields, given-fields|Model) \\ &= \text{Argmax}_{missing-fields} p(missing-fields|class, given-fields, Model) \end{aligned}$$

- This gives us a set of complete data with:

$$p(class, given-fields|Model) \approx p(class, \widehat{missing-fields}, given-fields|Model) .$$

- This needs to be done based on the “true” model, so we can first do a rough approximation to the “true” model (e.g., using the subset of complete data) and then do the fill in.
- Once we’ve learned *Model* a bit better, we can then fill in the missing values again, and relearn. This extension corresponds to the EM algorithm or Gibb’s sampling, depending on whether the fill-in is deterministic or probabilistic.

SS #83

METHODS: FRACTIONAL EXAMPLES

ID	Temperature (F)	Age	Weight	Blood Pressure	Sex	Blood Test	Diagnosis	$p(\text{case})$
35267	99.3?	25	180	150?	Male	negative?	flu A	0.180
35267	101.6?	25	180	137?	Male	positive?	flu A	0.340
35267	98.7?	25	180	155?	Male	negative?	flu A	0.480
83877	98.1	62	132	145	Female	negative	flu B	1
23414	99.4	44	211	161	Male	negative?	healthy	0.392
23414	99.4	44	211	161	Male	positive?	healthy	0.608
09372	103.2	41?	141	131	Female	negative	flu A	0.723
09372	103.2	65?	141	131	Female	negative	flu A	0.277
62253	97.1	58	130	141	Female	negative	healthy	1
67631	101.1	75	178	153	Female	positive	?	1
..
..

- Like the fill-in methods but do multiple fill-ins, and assign each a probability (denoted $p(\text{case})$ in the figure).
- Better approximates the sum.
- See [74, 15, 90].

SS #84

OTHER METHODS

All the usual methods for handling mixture models.

- EM algorithm.
- Gibbs sampling.
- Incremental versions of both [95].

SS #85

SECTION VIb.

Specialist topics:

- missing values;
- **knowledge discovery and refinement;**

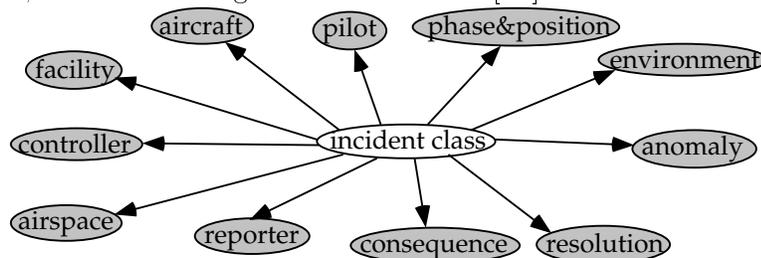
knowledge discovery requires prototyping and refinement of probabilistic models;

- connections between theories.

SS #86

HYBRID CLUSTERING AND KNOWLEDGE DISCOVERY

The US Aviation Safety Reporting System is a national resource for aircraft safety maintaining a database of aircraft incidents. Initial clustering/unsupervised learning of the database, was done using the model below [56].



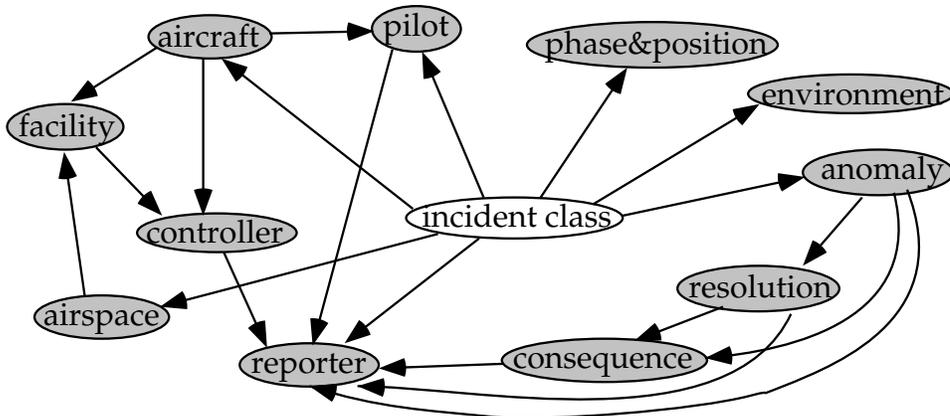
Aviation psychologists, reviewing the results, in many cases said, “so what?”

- We already know wide-body aircraft don't go on unscheduled flights (joy rides).
- We already know that if an aircraft has four pilots it must be military.

SS #87

KNOWLEDGE REFINEMENT

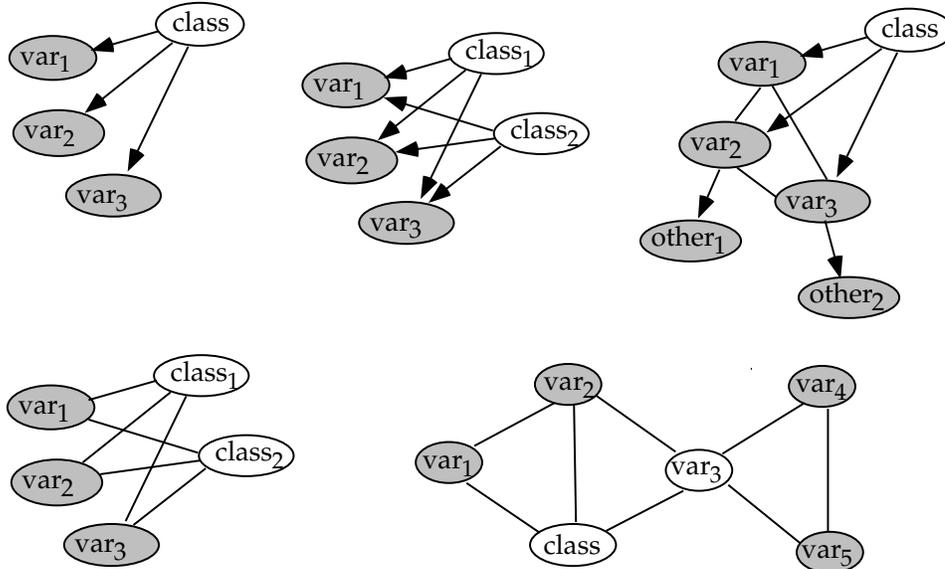
- Include this “known” information in the model, as arcs, probabilities, and informative priors on probabilities.
- Do clustering/unsupervised learning on the unexplained parts of the data.
- Following hybrid model partly specifies this.



SS #88

UNSUPERVISED LEARNING: STYLES

Other styles of unsupervised learning include single or multiple hidden (latent) classes, correlational models, etc.



SS #89

KNOWLEDGE REFINEMENT/DISCOVERY CYCLE

1. Learn with standard unsupervised model to get a feel for domain.
2. Have domain expert critique, extend and modify model.
 - Look, the system discovered that “only military planes have 4 pilots”. In fact, also, “military aircraft don’t fly in commercial airspace unless its an mistake”, so lets add all these to the initial model.
3. With improved model, encoding experts elicited knowledge, perform knowledge refinement and learning again.
4. Repeat the process.

NB. Requires prototyping software for knowledge discovery.

SS #90

SECTION VIc.

Specialist topics:

- missing values;
- knowledge discovery and refinement;
- **connections between theories.**

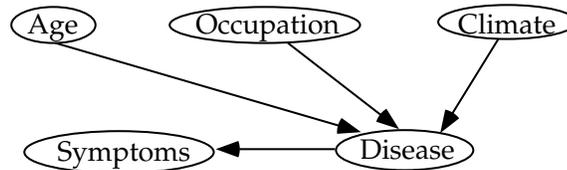
learning theories can be interpreted in the common framework of probability and decision theory;

SS #91

LEARNING BAYESIAN NETWORKS

To learn Bayesian networks on boolean variables, we must learn:

Structure, S : represented by a Directed Acyclic Graphs (DAGs) like below.



Conditional probability tables, θ : represented as below.

$Age < 45$	0.46
$Age \leq 45$	0.31

$p(Age)$

	<i>Disease</i>	
<i>Symptoms</i>	stomach ulcer	angina
stomach pain	0.23	0.17
chest pain	0.31	0.45

$p(Symptoms|Disease)$

SS #92

PAC THEORY

Consider Bayesian networks where there are n boolean variables and each variable is restricted to have at most k parents.

PAC theory generalized to the noisy case provides results on [46]:

Sample size bounds: To learn with *confidence* $1 - \delta$ that your network has Kullback-Leibler distance that is within ϵ of the optimum for a network with k parents, then use a sample of size

$$N = 2^{2k+13} \frac{n^2}{\epsilon^2} \left(k + 4 + \log \frac{n}{\epsilon} \right)^2 \left((k+1) \log 2(n+1) - \log \delta \right) .$$

These bounds are not tight, and probably can be improved.

Complexity of the search: For $k \geq 2$, the problem of finding a network with minimum Kullback-Leibler distance from the observed sample is NP-hard. Various search algorithms exist for these problems.

SS #93

WHAT DOES PAC *CONFIDENCE* MEAN?

OPT = the minimum Kullback-Leibler distance for a network with k parents from the “true” network (S, θ) .
 $S_{PAC}(sample)$ = the computed network with minimum Kullback-Leibler distance from the observed sample $sample$.
 $KL_{PAC}(sample)$ = its Kullback-Leibler distance from the “true” network (S, θ) .
 N = the size of the sample $sample$.
Confidence of $1 - \delta$ means that no matter what the “true” network (S, θ) is:

$$p(KL_{PAC}(sample) < OPT + \epsilon \mid N, S, \theta) \geq 1 - \delta .$$

This is equivalent to, no matter what prior over (S, θ) is used,

$$p(KL_{PAC}(sample) < OPT + \epsilon \mid N) \geq 1 - \delta .$$

Because this **ignores** the details of the sample, it answers the question [9, 42]: *if I obtain a sample of size N , how confident can I expect to be about learning (irrespective of the prior/truth)? NOT how confident can I be with my current sample?*

SS #94

APPROXIMATE BAYESIAN THEORY

The Bayesian **Maximum A Posteriori (MAP)** approximation is a constructive, asymptotically optimal scheme for creating an algorithm.

It provides [13, 21]:

An improved (over maximum likelihood) measure to optimize: Choose the structure S that maximizes the posterior log. probability $p(S|sample)$

$$p(S|sample) \propto p(S, sample) = p(S) \int_{\theta} p(sample|S, \theta) \cdot p(\theta|S) d\theta ,$$

Domain specific information can be used to tune the measure through priors to achieve **knowledge refinement**.

Techniques for computing it: For Bayesian networks on boolean variables, $p(S, sample)$ can be computed exactly in time $O(n2^{k+1}) + O(|sample|)$ where k is the maximum number of parents. The search problem to optimize this is in general equivalent to the PAC case.

SS #95

MDL THEORY

One version says to choose the structure S that minimizes the binary encoding of the sample *sample* together with the Bayesian network (structure S and probability tables θ). This **description length (DL)** is made up of:

$$DL(S) + DL(\textit{precision}(\theta)|S) + DL(\theta|S, \textit{precision}(\theta)) + DL(\textit{sample}|S, \theta) ,$$

where:

$DL(S)$ = number of bits needed to encode the structure, S ,

$DL(\textit{precision}(\theta)|S)$ = number of bits needed to encode the precision of the real values θ , etc.

This is **approximately equivalent** to the Bayesian MAP approach if we make the transformation [99]:

$$\begin{aligned} DL(S) &= -\log p(S) , \\ DL(\textit{sample}|S, \theta) &= -\log p(\textit{sample}|S, \theta) , \\ DL(\textit{precision}(\theta)|S) &\approx -\frac{1}{2} \log \det \textit{Variance}(\theta|S, \textit{sample}) \end{aligned}$$

SS #96

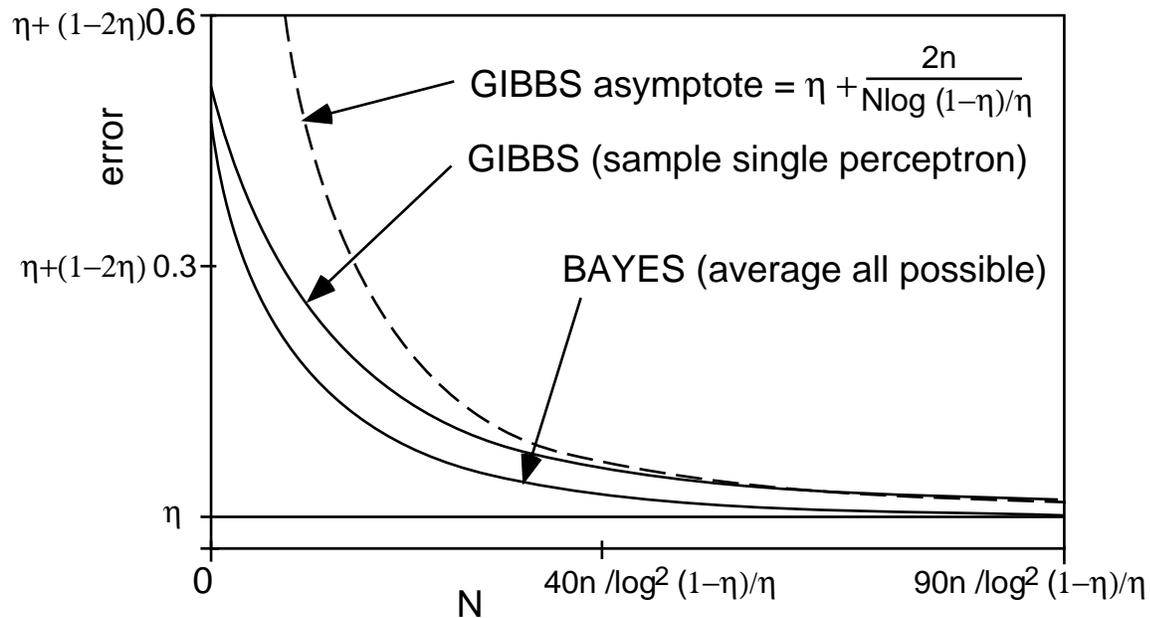
STATISTICAL PHYSICS APPLIED TO PERCEPTRONS

(nothing on Bayesian networks available!)

- Input is n binary variables and the output is a binary variable.
- In truth the output is classified according to a Perceptron rule with noise η added.
- We have a sample *sample* of size N .
- GIBBS algorithm randomly chooses the weights w for a perceptron according to its posterior probability $p(w|\textit{sample})$.
- BAYES algorithm does the full Bayesian approach: making a new prediction by doing a posterior weighted average of *all* weights w .
- Average learning curve for GIBBS (upper solid) and BAYES (lower) reproduced from [71].

SS #97

PERCEPTRONS, cont.



SS #98

LEARNING THEORIES: SUMMARY

- Theories can be interpreted and compared with language of probability and decision theory [42, 12].
 - PAC/PAB addresses the question: “how much data should I obtain?” in a prior independent way, not, “how do I learn with the current sample?”
 - Bayesian MAP and MDL methods related.
 - Statistical physics offers advanced techniques for implementing Bayesian methods.
- When addressing the same question, most practical theories say to do (in a crude sense) the same thing, but differ in their justification and philosophy.
- Most theories differ on how they say to address/side-step/ignore the problem of **priors**.
- Many researchers are now conversant with multiple theories and their correspondence.

SS #99

SECTION VII.

Essentials:

- check list of representations;
- check list of methods;
- check list of theory;
- check list of fields (see Section Ib).

SS #100

CHECK LIST OF REPRESENTATIONS

- Basic distributions: Gaussians, uniform, multinomial.
- Conditional multivariate distributions: various linear models [38, 60] trees [75, 7, 11], rules, and graph models [69, 55], feed-forward networks [82, 77].
- Undirected graphical models, i.e., Markov random fields, for vision, etc. [34, 78, 102, 45]
- Directed models, i.e., influence diagrams and Bayesian networks, including deterministic nodes [72, 70]
- Mixture models [96, 62, 50].

SS #101

CHECK LIST OF METHODS

- Maximum A Posterior (MAP) [73, 21].
- Exact Bayes factors [11, 14].
- Laplace's method, approximate Bayes factors, marginals and expected values [92, 54, 94].
- EM, ICM and other deterministic Gibbs models [92, 58, 27].
- Gibbs sampling and other Monte Carlo methods [68, 79, 35, 36]
- Differentiation, i.e., back-prop, (for Laplace's method and MAP).
- Methods for making the above parallel or on-line.
- Cross validation, bootstrap and empirical Bayes [30, 31].
- Methods for handling priors [5].
- Subsampling to handle large datasets [67].

SS #102

CHECK LIST OF THEORY

- Asymptotic, and large sample results (convergence, order of magnitude, etc).
- Transformations between MDL, Bayesian methods [99], and others.
- PAC and its Bayesian interpretation [12].
- Sample complexity, VC dimension, and other measures of problem complexity (see recent COLT work).
- Monte Carlo sampling theory [68].
- Interpretations of probability and rationality, standard "paradoxes" and their resolution.
- Broader issues such as Occam's razor, subjectivity vs. objectivity, the principle of indifference [3, 51].

SS #103

NEW ALGORITHMS FROM OLD

- use Gibbs to handle missing values when learning Bayesian networks; initialize using the “ignore them” strategy;
- use EM to handle missing values when learning class probability trees;
- devise a prior for decision lists and devise a MAP algorithm using local search to learn decision lists;
- do Kernel density or nearest neighbor type density estimation with a probabilistic mixture model;

SS #104

SOME RESEARCH QUESTIONS

Adapt the standard generic algorithms to:

- handle large samples more efficiently e.g., by subsampling [67];
- run on parallel computers, e.g., easy for Gibbs;
- incorporate and adjust for missing values efficiently;
- approximate/estimate sample and run-time complexity for a given problem specification relative to a particular algorithm (EM, Gibbs, etc.).

SS #105

References

- [1] D. Angluin and C.H. Smith. Inductive inference: Theories and methods. *Computing Surveys*, 15(3):237–269, 1983.
- [2] S. Ben-David and M. Jacovi. On learning in the limit and non-uniform (ϵ, δ) -learning. In L. Pitt, editor, *COLT'93: Workshop on Computational Learning Theory*, pages 209–217. Morgan Kaufmann, 1993.
- [3] J. O. Berger. Statistical analysis and the illusion of objectivity. *American Scientist*, 76(March-April):159–165, 1988.
- [4] J.O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer-Verlag, New York, 1985.
- [5] J.M. Bernardo and A.F.M. Smith. *Bayesian Theory*. John Wiley, Chichester, 1994.
- [6] J. Besag, J. York, and A. Mollie. Bayesian image restoration with two applications in spatial statistics. *Ann. Inst. Statist. Math.*, 43(1):1–59, 1991.
- [7] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees*. Wadsworth, Belmont, 1984.
- [8] G.L. Bretthorst. An introduction to model selection using probability theory as logic. In G. Heidbreder, editor, *Maximum Entropy and Bayesian Methods*. Kluwer Academic, 1994. Proceedings, at Santa Barbara, 1993.
- [9] W.L. Buntine. A critique of the Valiant model. In *International Joint Conference on Artificial Intelligence*, pages 837–842, Detroit, 1989. Morgan Kaufmann.
- [10] W.L. Buntine. Classifiers: A theoretical and empirical study. In *International Joint Conference on Artificial Intelligence*, Sydney, 1991. Morgan Kaufmann.
- [11] W.L. Buntine. Learning classification trees. In D.J. Hand, editor, *Artificial Intelligence Frontiers in Statistics*, pages 182–201. Chapman & Hall, London, 1991.
- [12] W.L. Buntine. *A Theory of Learning Classification Rules*. PhD thesis, University of Technology, Sydney, 1991. Written in 1990, awarded in 1992.
- [13] W.L. Buntine. Theory refinement of Bayesian networks. In D'Ambrosio et al. [24].
- [14] W.L. Buntine. Learning with graphical models. Technical Report FIA-94-02, Artificial Intelligence Research Branch, NASA Ames Research Center, 1994.
- [15] W.L. Buntine and A.S. Weigend. Bayesian back-propagation. *Complex Systems*, 5(1):603–643, 1991.
- [16] G. Casella and R.L. Berger. *Statistical Inference*. Wadsworth & Brooks/Cole, Belmont, CA, 1990.

- [17] B. Cestnik and I. Bratko. On estimating probabilities in tree pruning. In *Proceedings of the Sixth European Working Session on Learning*, Porto, Portugal, 1991. Pitman Publishing.
- [18] J.M. Chambers and T.J. Hastie, editors. *Statistical Models in S*. Wadsworth & Brooks/Cole, Pacific Grove, California, 1992.
- [19] E. Charniak. Bayesian networks without tears. *AI Magazine*, 12(4):50–63, 1991.
- [20] P. Cheeseman, M. Self, J. Kelly, W. Taylor, D. Freeman, and J. Stutz. Bayesian classification. In *Seventh National Conference on Artificial Intelligence*, pages 607–611, Saint Paul, Minnesota, 1988. American Association for Artificial Intelligence.
- [21] P.C. Cheeseman. On finding the most probable model. In J. Shrager and P. Langley, editors, *Computational Models of Discovery and Theory Formation*. Morgan Kaufmann, 1990.
- [22] B. Cheng and D.M. Titterton. Neural networks: A review from a statistical perspective. *Statistical Science*, 9:2–54, 1994. with comments and rejoinder.
- [23] G.F. Cooper and E.H. Herskovits. A Bayesian method for the induction of probabilistic networks from data. In D’Ambrosio et al. [24], pages 86–94.
- [24] B.D. D’Ambrosio, P. Smets, and P.P. Bonissone, editors. *Uncertainty in Artificial Intelligence: Proceedings of the Seventh Conference*, Los Angeles, CA, 1991.
- [25] G.R. Dattatreya and L.N. Kanal. Decision trees in pattern recognition. In L.N. Kanal and A. Rosenfeld, editors, *Progress in Pattern Recognition 2*, pages 189–239. Elsevier Science Publishers B.V., North Holland, 1985.
- [26] M.H. DeGroot. *Optimal Statistical Decisions*. McGraw-Hill, 1970.
- [27] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39:1–38, 1977.
- [28] L. Devroye. *A Course in Density Estimation*. Birkhauser, Boston, 1987.
- [29] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis*. John Wiley, New York, 1973.
- [30] B. Efron and R. Tibshirani. Statistical data analysis in the computer age. *Science*, 253:390–395, 1991.
- [31] B. Efron and R.J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, New York, 1993.
- [32] M. Frydenberg. The chain graph Markov property. *Scandinavian Journal of Statistics*, 1991.

- [33] S. Geman, E. Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4:1–58, 1992.
- [34] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian relation of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, 1984.
- [35] W.R. Gilks, D.G. Clayton, D.J. Spiegelhalter, N.G. Best, A.J. McNeil, L.D. Sharples, and A.J. Kirby. Modelling complexity: applications of Gibbs sampling in medicine. *Journal of the Royal Statistical Society B*, 55:39–102, 1993.
- [36] W.R. Gilks, A. Thomas, and D.J. Spiegelhalter. A language and program for complex Bayesian modelling. *The Statistician*, 1993.
- [37] D.J. Hand. *Kernel Discriminant Analysis*. Wiley, Chichester, 1982.
- [38] T. Hastie and R. Tibshirani. Generalized additive models. *Statistical Science*, 1:297–318, 1986.
- [39] W.K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.
- [40] D. Haussler. Quantifying inductive bias: AI learning algorithms and Valiant’s learning framework. *Artificial Intelligence*, 36(2):177–222, 1988.
- [41] D. Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Control*, 100(1):78–150, September 1992.
- [42] D. Haussler, M. Kearns, and R. Schapire. Bounds on the sample complexity of Bayesian learning using information theory and the VC dimension. *Machine Learning*, 14(1):83–113, 1994.
- [43] David Haussler and Manfred Warmuth. The probably approximately correct (pac) and other learning models. In A. Meyrowitz and S. Chipman, editors, *Machine Learning: Induction, Analogy and Discovery*. 1993.
- [44] M. Henrion, J.S. Breese, and E.J. Horvitz. Decision analysis and expert systems. *AI Magazine*, 12(4):64–91, 1991.
- [45] J.A. Hertz, A.S. Krogh, and R.G. Palmer. *Introduction to the Theory of Neural Computation*. Addison-Wesley, 1991.
- [46] K.-U. Höffgen. Learning and robust learning of product distributions. Research Report Nr. 464, revised May 1993, Fachbereich Informatik, Universität Dortmund, 1993.
- [47] E.J. Horvitz, D.E. Heckerman, and C.P. Langlotz. A framework for comparing alternative formalisms for plausible reasoning. In *Fifth National Conference on Artificial Intelligence*, pages 210–214, Philadelphia, 1986. American Association for Artificial Intelligence.

- [48] R.A. Howard. Decision analysis: perspectives on inference, decision, and experimentation. *Proceedings of the IEEE*, 58(5), 1970.
- [49] T. Hrycej. Gibbs sampling in Bayesian networks. *Artificial Intelligence*, 46:351–363, 1990.
- [50] R.A. Jacobs, M.I. Jordan, S.J. Nowlan, and G.E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1), 1991.
- [51] W.H. Jefferys and J.O. Berger. Ockham’s razor and Bayesian analysis. *American Scientist*, 80(Jan-Feb):64–72, 1992.
- [52] D.S. Johnson, C.H. Papdimitriou, and M. Yannakakis. How easy is local search? In *FOCS’85*, pages 39–42, 1985.
- [53] M.I. Jordan and R.I. Jacobs. Supervised learning and divide-and-conquer: A statistical approach. In ML10 [66], pages 159–166.
- [54] R.E. Kass and A.E. Raftery. Bayes factors and model uncertainty. Technical Report #571, Department of Statistics, Carnegie Mellon University, PA, 1993. Submitted to Jnl. of American Statistical Association.
- [55] Ron Kohavi. Bottom-up induction of oblivious, read-once decision graphs : Strengths and limitations. In *Twelfth National Conference on Artificial Intelligence*, 1994. Paper available by anonymous ftp from `Starry.Stanford.EDU:pub/ronnyk/aaai94.ps`.
- [56] R. Kraft and W.L. Buntine. Initial exploration of the ASRS database. In *Seventh International Symposium on Aviation Psychology*, Columbus, Ohio, 1993.
- [57] K. Lange and J.S. Sinsheimer. Normal/independent distributions and their applications in robust regression. *Journal of Computational and Graphical Statistics*, 2(2), 1993.
- [58] S.L. Lauritzen. The EM algorithm for graphical association models with missing data. *Computational Statistics and Data Analysis*, 1993. To appear; available as Tech. Rep. R 91-05, Institute for Electronic Systems, Aalborg University.
- [59] M. Li and P. Vitányi. Inductive reasoning and Kolmogorov complexity. *Journal of Computer and Systems Science*, 44(2):343–384, 1992.
- [60] P. McCullagh and J.A. Nelder. *Generalized Linear Models*. Chapman and Hall, London, second edition, 1989.
- [61] G. McLachlan. *Discriminant Analysis and Statistical Pattern Recognition*. Wiley, New York, 1992.
- [62] G. J. McLachlan and K. E. Basford. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, New York, 1988.
- [63] I. Meilijson. A fast improvement to the EM algorithm on its own terms. *J. Roy. Statist. Soc. B*, 51(1):127–138, 1989.

- [64] D. Michie, D.J. Spiegelhalter, and C.C. Taylor, editors. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, Hertfordshire, England, 1994.
- [65] S. Minton, M.D. Johnson, A.B. Philips, and P. Laird. Solving large-scale constraint-satisfaction and scheduling problems using a heuristic repair method. In *Eighth National Conference on Artificial Intelligence*, pages 17–24, Boston, Massachusetts, 1990. American Association for Artificial Intelligence.
- [66] *Machine Learning: Proc. of the Tenth International Conference*, Amherst, Massachusetts, 1993. Morgan Kaufmann.
- [67] R. Musick, J. Catlett, and S. Russell. Decision theoretic subsampling for induction on large databases. In ML10 [66].
- [68] R.M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, Dept. of Computer Science, University of Toronto, 1993.
- [69] J.J. Oliver. Decision graphs - an extension of decision trees. In *Proceedings of the Fourth International Workshop on Artificial Intelligence and Statistics*, pages 343–350, 1993. Extended version available as TR 173, Department of Computer Science, Monash University, Clayton, Victoria 3168, AUSTRALIA.
- [70] R.M. Oliver and J.Q. Smith, editors. *Influence Diagrams, Belief Nets and Decision Analysis*. Wiley, 1990.
- [71] M. Opper and D. Haussler. Calculation of the learning curve of Bayes optimal classification algorithm for learning a perceptron with noise. In *COLT'91: 1991 Workshop on Computational Learning Theory*, pages 75–87. Morgan Kaufmann, 1991.
- [72] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.
- [73] S.J. Press. *Bayesian Statistics*. Wiley, New York, 1989.
- [74] J.R. Quinlan. Unknown attribute values in induction. In A.M. Segre, editor, *Proceedings of the Sixth International Machine Learning Workshop*, Cornell, New York, 1989. Morgan Kaufmann.
- [75] J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1992.
- [76] J.R. Quinlan and R.L. Rivest. Inferring decision trees using the minimum description length principle. *Information and Computation*, 80:227–248, 1989.
- [77] B. D. Ripley. Neural networks and related methods for classification. *Journal of the Royal Statistical Society B*, 56(3), 1994.
- [78] B.D. Ripley. *Spatial Statistics*. Wiley, New York, 1981.
- [79] B.D. Ripley. *Stochastic Simulation*. John Wiley & Sons, 1987.

- [80] B.D. Ripley. Statistical aspects of neural networks. In *Invited lectures for Sem-Stat (Séminaire Européen de Statistique)*, Sandbjerg, Denmark, 1993. Chapman and Hall.
- [81] J. Rissanen. Stochastic complexity. *Journal of the Royal Statistical Society B*, 49(3):223–239, 1987.
- [82] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning internal representations by error propagation. In David E. Rumelhart, James L. McClelland, and the PDP Research Group, editors, *Parallel Distributed Processing*, page 318. MIT Press, 1986.
- [83] S.R. Safavian and D. Landgrebe. A survey of decision tree classification methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(3), 1991.
- [84] C. Schaffer. Overfitting avoidance as bias. *Machine Learning*, 13(1), 1993.
- [85] D. W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley and Sons, New York, 1992.
- [86] H.S. Seung, H. Sompolinsky, and N. Tishby. Learning curves in large neural networks. In *COLT'91: Workshop on Computational Learning Theory*. Morgan Kaufmann, 1991.
- [87] H.S. Seung, H. Sompolinsky, and N. Tishby. Statistical mechanics of learning from examples. *Physical Review A*, 45:6056–6091, 1992.
- [88] R.D. Shachter. An ordered examination of influence diagrams. *Networks*, 20:535–563, 1990.
- [89] R.D. Shachter and D. Heckerman. Thinking backwards for knowledge acquisition. *AI Magazine*, 8(Fall):55–61, 1987.
- [90] D.J. Spiegelhalter and S.L. Lauritzen. Sequential updating of conditional probabilities on directed graphical structures. *Networks*, 20:579–605, 1990.
- [91] S.M. Stigler. *The History of Statistics: The Measurement of Uncertainty before 1900*. Belknap Press of Harvard Uni. Press, Cambridge, Massachusetts, 1986.
- [92] M.A. Tanner. *Tools for Statistical Inference*. Springer-Verlag, New York, second edition, 1993.
- [93] H.H. Thodberg. Bayesian backprop in action: Pruning, ensembles, error bars and applications to spectroscopy. In *Advances in Neural Information Processing Systems 5 (NIPS'93)*. Morgan Kaufmann, 1994.
- [94] L. Tierney and J.B. Kadane. Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association*, 81(393):82–86, 1986.
- [95] D.M. Titterton. Recursive parameter estimation using incomplete data. *Journal of the Royal Statistical Society B*, 46(2):257–267, 1984.

- [96] D.M. Titterton, A.F.M. Smith, and U.E. Makov. *Statistical Analysis of Finite Mixture Distributions*. John Wiley & Sons, Chichester, 1985.
- [97] P.J.M. van Laarhoven and E.H.L. Aarts. *Simulated Annealing: Theory and Applications*. D. Reidel, Dordrecht, 1987.
- [98] V. Vapnik. *Estimation of Dependencies Based on Empirical Data*. Springer-Verlag, New York, 1982.
- [99] C.S. Wallace and P.R. Freeman. Estimation and inference by compact encoding. *Journal of the Royal Statistical Society B*, 49(3):240–265, 1987.
- [100] C.S. Wallace and J.D. Patrick. Coding decision trees. *Machine Learning*, 1993.
- [101] S.M. Weiss and C.A. Kulikowski. *Computer Systems That Learn*. Morgan-Kaufmann, 1991.
- [102] J. Whittaker. *Graphical Models in Applied Multivariate Statistics*. Wiley, 1990.
- [103] D. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–260, 1992.