# Statistics

vishvAs vAsuki

August 28, 2014

## Contents

# Part I

# Introduction

## 1 Themes

Statistics is the study of the collection, organization, and interpretation of data.

### 1.1 Statistical inference

Aka inductive statistics, inferential statistics.

### 1.1.1 Inference from data subject to randomness

This is the process of drawing conclusions from data that are subject to random variation, for example, observational errors or sampling variation. Probability theory provides us a way of handling this uncertainty.
Also see the Inference survey for other scenarios where random variation is lesser.

### 1.1.2 Modeling problems

Some problems are very closely tied to particular applications; eg: many vision problems, etc.. Here, the art is often in picking the right model; ie in proper abstraction. Eg: Is it better to use linear dimensionality reduction or manifold learning? Is it better to cast it as a regression problem or a classification problem? Often, simple models work surprisingly well. Also see comments in the 'probabilistic models' survey.
The decision about what you model and what you don't is important. Often, prior work is advanced by designing a model which modeled more aspects of the problem than previous models.
Once the problem is modeled, one must find a suitable estimation algorithm.
See also comments

### 1.1.3 Solving Modeled problems

This is algorithm design and analysis. See algorithms survey too.
Here, you take already-modeled well-specified problems, and try to find upper and lower bounds on resources (time, sample points, memory, randomness .. ) required for various estimation tasks.
Note that finding upper bounds on resources often involves designing new estimation algorithms to solve the problem. This is often done by modifying an algorithm for a closely related model.
Eg: Pattern recognition in a Stream: very limited memory and processing time. Eg: sensors in bridges.

#### 1.1.3.1 Degrees of abstractness

**Domain specific models** Usually solutions tied to particular domains tend to have short term, shallow impact. But, sometimes the general modeling technique and the estimation algorithm in them can be generalized to other domains, leading to greater impact. Also, the problem solved itself may be very important from a non-data-mining point of view; like the astrophysics or biology points of view.

**Abstract models** Impact is often intangible: it improves the intuition/ understanding of problems by practitioners. Good solutions to important abstract problems tend to have long-term, deep impact.

## 1.2 Research effort

Read many papers; Critique the problems posed and the solutions presented; alter problems and argue why it is interesting; criticize solutions, develop better solutions for original problem.

### 1.2.1 Analysis of efficiency and complexity

For learning theory research, see colt ref. Statistical learning theory: find convergence rate in estimating parameter.

### 1.2.2 Modeling and Experimentation

#### 1.2.2.1 Purpose

Modelling the problem often requires exploration and experimentation.
Show that the techniques, even if mathematically well motivated, works well in practice. If it fails, in what cases does it fail?

#### 1.2.2.2 Avoiding unnecessary work

One must approach the problem in a mathematically principled, disciplined manner. Doing so can avoid expending much unnecessary effort. One must take advantage of prior research and predetermined thought patterns.

#### 1.2.2.3 Degree of experimentation

This depends highly on abstractness of the problem. If the problem is very specific to some domain, eg: vision or social network analysis etc; experimentation is much. One has to check the goodness of one's model with the ground truth.

#### 1.2.2.4 Data preparation

Relevant mainly in case of domain-specific research; Usually not present in the case you try to answer more abstract questions.
Huge effort often involved in data-preparation.

### 1.2.3 Following research

ICML, KDD, NIPS. Application conferences: CVPR etc.. Theory: archiv, colt. Data mining conferences tend to be more about applications and clustering than machine learning conferences.

## 1.3 Software

Rapidminer, R, Matlab are frequently used.

Keep up with developments in efficient software to solve common problems: for example, by joining the R package mailing list.

# 2 Decision theory

Certain elements common to probability density estimation, distribution structure learning (including classification) etc.. can be studied within the abstract framework of decision theory.

## 2.1 Agents: Actions, policies

How should an agent act in the face of uncertainty, given some observations? Our objective is to find good decision procedures for the agent.
Like a game against nature. See game theory reference for adversarial games.

### 2.1.1 State and parameters

#### 2.1.1.1 State space

The state of nature changes, possibly in response to actions made by an agent. The agent must act optimally in some sense in the presence of uncertainty.

#### 2.1.1.2 Parameter space T

Some parameters $\theta \in T$ describe the state of nature (the ground truth), especially as it relates to the action-space of an agent; and they are unknown.

#### 2.1.1.3 State transitions

Some parameters describe state transitions. These are aka population parameters. In some problems, we are to estimate some properties of a population by drawing random samples from it. This unknown parameter/ pattern of the population, $t = g(\theta)$, is a constant. Eg: mean, variance of heights.

### 2.1.2 Action space A

Given any change in its knowledge, the agent can act. Its action is chosen from the action space $A = \{a\}$. Actions can change the state of nature and the parameters which specify it.

#### 2.1.2.1 Common examples

In case of density estimation, given the data point $x$, $A$ could be the choice of the density function $f_{\hat{t}}(x)$ as a function of the estimated density $\hat{t}$. In this case, the state of nature is usually not affected by our guess about its parameters.
In hypothesis testing, it could be the probability of the observation $x$, which is a function of the hypothesis chosen earlier.

In case of classification, $A$ would correspond to the various possible labellings of a data point $x$.

### 2.1.3 Goodness of actions

#### 2.1.3.1 Loss function L

$L : T \times A \to \Re$. You pick the loss function to best model the situation faced by the agent, and also for mathematical tractability - this is often the modeler's job.

#### 2.1.3.2 Examples

Squared error between prediction and label in case of classification.
$-f_{\hat{t}}(x)$ in case of probability distribution modeling.

### 2.1.4 Decision procedure d

#### 2.1.4.1 Mapping observation D to actions

Denote an observation by the random variable $D$. Decision procedures take as input $o \in ran(D)$ and return an action $a \in A$. Their pre-image can be often be replaced with sufficient statistics drawn from observations.
Note that the observation $D$ to which a decision procedure is supposed to respond ***is different from the training set of prior observations which may have been used to learn*** $d$***!***

#### 2.1.4.2 Deterministic procedures

Deterministic decision procedures make decisions which are a function of the new observation.
A deterministic decision procedure is $d : ran(D) \to A$.

#### 2.1.4.3 Randomized procedures

A randomized decision procedure takes decisions with some randomness, given new observations. So, the randomized procedure is in the convex hull of certain decision procedures.
If the decision procedure randomized, range is the set of random variables with range $A$.

**Combining decision procedures** Randomized procedures can often be written as a 2-step meta-procedure with step 1: random choice of decision procedures, and step 2: application of the decision procedure chosen.

#### 2.1.4.4 Examples

In case classification, the classification rule can be considered to be a decision procedure.
In the case of distribution estimation, given a point (the observation), the estimated pdf can be considered to be the decision procedure.
In case of hypothesis testing, a certain hypothesis can be considered to be the decision procedure.

## 2.2 Risk R of decision procedure d

### 2.2.1 Motivation and setting

We want to choose a decision procedure from $H_n$ (so named because it is often learned after looking at a sample of size $n$). We want to pick decision procedure with least expected loss; that motivates the following definition. This is essentially the problem of statistical inference.

### 2.2.2 Risk

$R : T \times \{d\} \rightarrow \Re$. $R(\theta, d) = E_D[L(\theta, d(D))]$. In case $d$ is randomized, the expectation is also over random bits used by $d$.
In case of the estimation or prediction problem, if you use squared distance as $L$, you get the bias - variance decomposition: See estimation section.

### 2.2.3 Uncertain ground truth case

#### 2.2.3.1 The need

Suppose that $\theta$ is constant every time you apply the chosen $d$. Even so: When you pick $d$, you don't know $\theta$, the ground truth.
Also, maybe $\theta$, which decides the goodness of response, changes with the actions taken by the decision procedure.
For example, in classification, $\theta$ corresponds to the label $y$ corresponding to the example $x$ seen by the classifier. Even if the classifier knew the distribution $f_{Y|X}(y|x)$ generating the examples, it cannot in general be certain about $y$ given $x$.

#### 2.2.3.2 Frequentist and epistemological approaches

Rooted in the two distinct but overlapping interpretations of probability, we observe two overlapping approaches to statistical inference. The frequentist approach tries to deal with uncertainty by relying on sampling, while only partially restricting the hypothesis space.
Bayesian/ epistemological inference approach tends to posit and quantify prior beliefs about the ideal decision, and use this together with sample evidence to reach a conclusion.

#### 2.2.3.3 Prior beliefs about ground truth

We can model the uncertainty in the ground truth using a probability distribution $\theta \sim P_T$ over $T$; so now take
$R'(d) = E_{D,P_T}[L(\theta, d(D))] = \int_T E_D[L(\theta, d(D))]P_T(\theta)d\theta$.
As probability theory is being fully used to model uncertainty, this is called Bayesian risk evaluation.

#### 2.2.3.4 Prior beliefs about best d

For every $\theta$, there is a best decision procedure $d(D)$. So, alternatively, one model uncertainty in what the best decision procedure is directly as $d \sim P'(d)$ and look at $\theta = f(d_1)$. Then, one can write the risk
$R'(d) = E_{D,P'}[L(f(d_1), d(D))] = \sum_{d_1} E_D[L(f(d_1), d(D))]P'(d_1)$.

#### 2.2.3.5 Additive form

Let $g(d; d_1) = E_D[L(\theta, d(D))] = E_D[L(f(d_1), d(D))]$.
Then, $R'(d) = E_{P'}g(d; d_1)$. If the loss $g(d; d_1)$ had a sharp drop around $d$, or if $P'(d)$ was highly concentrated around $d_1$, we could approximate this as $R'(d) = g(d; d_1)P'(d)$.
For convenience for use in optimization problems, one often takes logarithms on both sides to get: $R''(d) = \log g(d; d) + \log P'(d)$.
This roughly motivates the following form used in practice: $R''(d) = R(d) + lr(d)$, where $L$ is loss function, r() is regularizer. The regularizer, r() ensures that the prior belief about $\theta$ is taken into account while evaluating the risk of decision procedure $d$.
Example: Suppose that we are estimating $\mu$: eg: avg age. But, prior belief is that it is $\mu_0$. Thence a decision procedure: $R(\mu, \hat{\mu}(D)) = 0.2\mu_0 + 0.8n^{-1}\sum X_i$.

**Strict restrictions** Finding $\min_d R(d)+lr(d)$ is same as $\min R(d) : r(d) \leq c_l$. This is a strict restriction on the choice of $d$, unlike a more flexible prior assumption about the pdf of $d$ which is the hallmark of epistemological/ Bayesian inference; where irrespective of a low prior probability assigned to a certain $d$, the weight of evidence could lead to choosing that $d$.

### 2.2.4 Geometry of R

Visualize $R(\theta, d)$ with $\theta \in T$. Make a function space, label each dimension with some $\theta \in T$. For any fixed $d$, $R(\theta, d)$ is a vector in this space. Given set $S = \{d\}$, can get set $S'$ of all randomized decision procedures derivable from these. Risks $R(S')$ are points in the convex hull of $R(\theta, d)\forall d \in S$. In 2 d: it is a convex polygon: only straight edges.

### 2.2.5 Empirical risk

Aka empirical avg loss. $\hat{R}(d;D) = n^{-1}\sum_{x_i \in D} L(\theta, d(x_i))$. You have: $R(d) = E_D[\hat{R}(d;D)]$.
Where we have prior beliefs about the nature of the best $\theta$ or $d$, empirical risk is used with a regularization function to define an alternate risk function. See Bayesian risk evaluation for details.

### 2.2.6 Minimal risk

#### 2.2.6.1 Definition

The lowest (expected) risk any decision procedure $d$ can achieve is often called the Bayes Risk.
It is not necessarily 0. For example, suppose that we were considering classification of points drawn from the distribution $f_{Y|X}(y|x), f_X(x)$. Even if the decision procedure had complete knowledge of $f_{Y|X}(y|x), f_X(x)$, its risk in general would not be 0 as there is the possibility that $\hat{y} = argmax_y f_{Y|X}(y|x)$ is the wrong labeling for $x$.

#### 2.2.6.2 Risk Consistency of d

Let $D_i$ be data of $i$ sample points. Then does $d(D) \rightarrow^p \theta$ [In case of the parameter estimation task]? Or does $R(\theta, d) \rightarrow^p minRisk$?

#### 2.2.6.3 Risk persistence

Ground truth $\theta$. Let $\hat{t}$ be the chosen decision procedure.
Take $\hat{\theta} = \arg\min_{t \in H_n} R(t)$. How close is $R(\hat{t})$ to $R(\hat{\theta})$?
As $n \rightarrow \infty$, usually want $\hat{t} \rightarrow \theta$ or better: $R(\hat{t}) - R(\theta) \rightarrow 0$. But it may not be possible as decision procedures can't get to it, or maybe $\theta$ changes with n.
Approximation error $R(\hat{\theta}) - R(\theta)$. Then does $d$ at least minimize estimation error $R(\hat{t}) - R(\hat{\theta})$?

#### 2.2.6.4 In High dimensional setting

Let the dimensionality of data-points be $p$. Suppose $p >> n$ and we still want a decision procedure which works well. What is risk persistence here? Hence a new notion: As $n \rightarrow \infty$, we still want check if $R(\hat{t}) - R(\hat{\theta}) \rightarrow 0$; while $p$ scales with $n$.

## 2.3 As a POMDP

POMDP's are well suited to abstract active learning; but it is informative to consider it more generally.

### 2.3.1 State and observation

The ground truth, or parameter $T$ corresponds to the state of the world in case of a POMDP. The observation $D$ gives some clue about the current state.

### 2.3.2 Transitions

Unlike active learning problems, often the state transitions and observations thereof are independent of the agent's actions. So, the agent often has no control over input distribution. 'Parameters' in decision theory may confusingly include both those describing the ground truth and those describing the transition function.

### 2.3.3 Action and Loss

The loss function corresponds to the reward which depends on current state and the action taken.

### 2.3.4 Policy

The decision procedure $d$ corresponds to the POMDP's policy; except that the belief state is not explicitly considered for our purposes.

### 2.3.5 Risk vs value

In the case of MDP's, it is common to assess policies using the total expected reward over a possibly infinite horizon, so a discounting factor $g \in [0, 1]$ is needed. [Expected] risk of a decision procedure focuses on the expected reward.

# 3 Procedure for choosing decision procedure d

## 3.1 Overview, motivation

Some discipline is necessary in order to avoid common errors which consume effort and resources. The below describes steps common to many problems including density estimation and classification.
Formulate the problem in decision theoretic terms properly - pick an appropriate 'risk/ loss function' (possibly incorporating prior belief), a candidate feature set. Then fix a model family (possibly after a literature survey) and develop a model selection procedure. Then use empirical risk estimation using cross validation in order to avoid overfitting to training data. Repeat.

## 3.2 Picking the right hypothesis space H

### 3.2.1 Hypothesis space H

Often we constrain ourselves to picking a decision procedure from a certain limited set, the hypothesis space $H$.
$H$ is usually specified by imposing a structure which must be satisfied by all decision procedures in it; so it may also be called a model family.
$H$ can be further restricted by specifying the loss function appropriately: this is described elsewhere.

### 3.2.2 Motivation, Factors to consider

Picking a limited hypothesis space $H$ is often necessary because considering all possible hypotheses may be computationally infeasible. Also, we may want $H$ to be convex, so that we can then efficiently minimize empirical risk over it.
Prior beliefs/ requirements about the ideal decision procedure also influence the choice of $H$.
Constraining $H$ just for the sake of avoiding overfitting is considered elsewhere.

#### 3.2.2.1 Approximation vs estimation error tradeoff

$\hat{\theta} = \arg\min_{t \in H_n} R(t)$. If $H_n$ large, we can reduce approximation error $|R(\theta) - R(\hat{\theta})|$, but the decision procedure learning process will find it tougher to get to $\hat{\theta}$ in terms of time, memory or number of samples required. In other words, estimation error $|R(\hat{\theta}) - R(d)|$ high. ***Larger hypothesis classes generally outperform smaller ones - if given enough data.***
If $H_n$ too small, approximation error is high, estimation error low. ***When training data is limited, smaller hypothesis class is better. But, Misleading if assumption about the best decision procedure is false.***

### 3.2.3 Parameters

A decision procedure may be concisely specified using a small number of parameters $\theta$ - perhaps as a function over sufficient statistics extracted from observations ($S = s$). Other decision procedures require $O(|s|)$ of parameters to be specified.
Hypothesis classes are called non-parametric or parametric based on the decision procedures which constitute them. Decision procedure learning methods may similarly be parametric or non-parametric, based on the hypothesis class used.
Non-parametric hypothesis classes are usually bigger and more flexible.
Many of the advantages and disadvantages associated with using parametric and non-parametric classes therefore follow from comparison of the use of large and small $H$ described earlier.

#### 3.2.3.1 Common structural assumptions

Sparsity. Low matrix rank. Block structure: Eg: zeros appear in blocks/ entire rows/ columns; row sparsity.

## 3.3 Loss function choice

Choosing a risk or loss function enables us to compare decision procedures.
Constraining $L$ in order to avoid overfitting during empirical risk minimization is considered elsewhere.

### 3.3.1 Evaluation

When ultimately evaluating a decision procedure, one should use the loss function without prior belief biases (unlike the one possibly used in empirical risk minimization). This actual risk of the decision procedure chosen is called the generalization error.
In research reporting: Final evaluation/ test data must never be used in fitting (hyper)parameters for this results in overfitting to the test/training set combination.

## 3.4 Theoretically find the best d

### 3.4.1 Best d for fixed $\theta$

Use geometry of $R$ described earlier. If $\theta$ were fixed, then the best $d$ would correspond to the point of the convex hull $R(S')$ with least $R(\theta, d)$: This will always be a corner.

### 3.4.2 Minimum expected (Bayesian) risk

Pick $d$ with min expected risk.

#### 3.4.2.1 Geometry

Let dim(T) be finite.
If $\pi(\theta)$ is the prior, then risk is $\sum_i \pi(\theta_i)R(\theta_i, d) = c$ where $d$ is the best response for a certain $\theta$. So, this is a certain point in the 'base' of the convex hull $R(S')$. So set of risks of such decision procedures are represented in the 'base' faces of the convex polyhedron.
Visualize this in 2D.
Comparing geometries: Every bayesian risk minimizing decision procedure with full support is also admissible.

### 3.4.3 Choose an admissible procedure

d is admissible if $\nexists d' : [\forall \theta R(\theta, d') \leq R(\theta, d)]$, $\exists \theta : [\forall d' : R(\theta, d') < R(\theta, d)]$.

#### 3.4.3.1 Geometry

Take convex hull $R(S')$. Given $d$, to find a d' which 'dominates' $d$ for some fixed $\theta_i$, while being as good as $d$ for all other $\theta$: drop a line through $d$ which is $\perp$ axis corresponding to $\theta_i$. The lowest point in the hull along this line dominates every other point in the hull.
So, possible admissible procedures correspond to base of the convex hull, except faces which are perpendicular.
So, every admissible procedure is a min Bayes risk procedure for some prior.

### 3.4.4 The adversarial setting: minimax procedure

See game theory. Ye assume that the adversary knows what you pick, assume that adversary is rational, and choose the $\hat{d} = \inf_d \sup_\theta R(\theta, d)$.

#### 3.4.4.1 Geometry

Take convex hull $R(S')$. Find the 'lowest' point in the hull which meets the ball $\|\theta\|_\infty = c$ (max norm: looks like a square).

### 3.4.5 Compare risk profiles of d over range of $\theta$

$R(\theta, d)$ can change for different values of $\theta$. So, can plot $R(\theta, d)$ vs $\theta$ curves for various $d$ and compare.

## 3.5 Empirical risk minimization

Aka model selection, training the model, learning the parameters (if the hypothesis class is parametric).
Solve: $\hat{t} = argmin_t \hat{R}(t, D)$. So, you try to get best fit to training data from $H$.
If both $H$ and $\hat{R}(t, D)$ are convex, you have a convex optimization problem.

### 3.5.1 Fit to D vs generalization ability

A problem with empirical risk minimization is the limited amount of data available. This means that minimizing $\hat{R}(t, D)$ may be quite different from minimizing $R$.

#### 3.5.1.1 Training error

$\hat{R}(t, D)$, derived only from a loss function, not incorporating any bias about the decision procedure, is aka the training error. It is useful in diagnozing underfitting and overfitting.

#### 3.5.1.2 Overfitting

Sometimes, an algorithm may pick a decision procedure which minimizes empirical risk on the training data/ training error (ie maximizes fit to $D$), but has high risk $R$ (low generalization ability).
As in this case, the decision procedure varies highly with the training data $D$, it is called the 'high variance' case.

#### 3.5.1.3 Underfitting

The decision procedure chosen by the algorithm may have a high generalization error and high training error. This could be due to the non-convexity of the empirical risk function.
This case is aka the 'high bias case'.

### 3.5.2 Overfitting and model complexity

#### 3.5.2.1 Model complexity

Complex model classifies training data excellently but performs poorly on test data. So, the complex model is unstable.
So, look for simple hypothesis, in Occam's razor style; minimize description length. Keep low model complexity/ level of specialization while still fitting the training set well.

#### 3.5.2.2 Limiting number of parameters

As number of data points $N$ increases, this problem becomes less. Rough heuristic: $N \geq 5$ or 10 times the number of adaptive parameters in model.

#### 3.5.2.3 Polynomial regression example

Eg: Fitting a polynomial with parameters $w$ to a sine curve: what should the degree $d$ be?
For a given $N$, generalization ability increases (error on test set decreases) with $d$, then levels off until $d$ = N-1 when it sharply drops (error increases): overfitting: simple interpolation occurs there, and magnitude of coefficients ($w_i$) sharply increases there. So use regularization, with $w_0$ omitted in regularization term.

### 3.5.3 Avoiding overfitting

#### 3.5.3.1 Altered risk function

When we have prior belief about the best decision procedure, it can be incorporated into the expression for the risk function to be used for empirical risk minimization.

Take $t_n = argmin_t \hat{R}(t,(X_i)) + \lambda r(t)$, where $\hat{R}$ is the original risk function, $r()$ is regularizer and $\lambda$ is the regularization parameter.

#### 3.5.3.2 Altered loss function

The altered risk function can be interpreted as arising from an altered loss function constructed especially for empirical risk minimization - one incorporating the prior belief.

#### 3.5.3.3 Other derivations

This alternate objective function may also be derived by other means: for example using the Bayesian/ Schwartz or Akaiki Information Criteria (BIC or AIC) in case of parametric density estimation.

#### 3.5.3.4 Hyper-parameters

Hyper-parameters such as the regularization parameter represent the prior belief about the decision procedure. They could include choice between model families.

### 3.5.4 Statistical efficiency analysis

If $L$ is convex, differentiable, strongly convex around t' wrt certain $C$ (see vector spaces ref). Then take $A = H_n$ as the hypothesis space and B a certain space outside it, so that decomposability holds wrt r(): $\forall a \in A, b \in B : r(a+b) = r(a) + r(b)$. Then, pradeep etal bound distance $d(t', t_n)$.

#### 3.5.4.1 Accuracy of empirical risk estimate

**Core Idea**: I ‖f the $H$ is small and if $\hat{t}$ is learned using many samples, the empirical risk is close to actual risk.
If $G$ finite, can bound $Pr(\sum_i (L(t,x_i) - E[L(t,x)]) \geq \epsilon)$ using martingale concentration bound (Hoeffding). Can then apply union bound to get bound on $Pr(\sup_t \sum_i (L(t,x_i) - E[L(t,x)]) \geq \epsilon)$. [The union bound is very intuitive when probability is viewed as a measure.]
If $G$ infinite, take $\epsilon$ covering wrt $\|.\|_\infty$ to get $G_\epsilon$, take union bound over that. Thence bound: $N(\epsilon, G, \|.\|_\infty) exp(\frac{-2n\epsilon^2}{9B^2})$.
These same principles are used in 'Occam razor' and the 'VCD Occam razor' in computational learning theory.

#### 3.5.4.2 Bound deviation from optimum h: bound empirical risk error

**Core Idea**: I ‖f the $H$ is small and if $\hat{t}$ is learned using many samples, the risk due to $\hat{t}$ is close to that of $\theta$.

Thence, $R(\hat{t}) - R(\theta) = R(\hat{t}) - \hat{R}(\hat{t}) + \hat{R}(\hat{t}) - \hat{R}(\theta) + \hat{R}(\theta) - R(\theta) \leq 2\sup_t(R(t) - \hat{R}(t))$: as $\hat{R}(\hat{t}) - \hat{R}(\theta) \leq 0$ from empirical loss minimization.
Thence,

$$R(\hat{t}) - R(\theta) \leq 2n^{-1} \sup_t \sum_i (L(t, x_i) - E[L(t, x)]).$$

So, want to bound $Pr(\sup_t \sum_i (L(t, x_i) - E[L(t, x)]) \geq \epsilon)$. Take function space $G = \{L(t, .) \forall t\}$.

### 3.5.5 Check generalization ability

Aka validation.

#### 3.5.5.1 Motivation

Given limited data and having picked a decision procedure $\hat{t}$, one is often motivated to approximate the generalization ability. Then, one can judge whether over-fitting or under-fitting has happened.

#### 3.5.5.2 General procedure

One partitions the limited data in some way into training data and validation data, which is aka 'hold out set', a stand-in for the data the decision procedure will ultimately be evaluated on.
Then, learn parameters of different models on first part of training data and test for generalization ability by measuring the empirical loss (without any regularization for prior biases about the decision procedure) on the hold-out set.

#### 3.5.5.3 Multiple rounds for robustness

For the sake of robustness, this can be repeated many times, and the empirical risk associated with a given model is taken to be the average of empirical risks measured during these runs. The model with the least average empirical risk is considered the best.

#### 3.5.5.4 Cross validation

A particular variant is k-fold cross validation. Here, the data is divided into $k$ equal sets. There are $k$ different validation rounds. In each round, one of these $k$ sets is designated the 'hold-out' set, while the union of the remaining sets is designated the training set.
Cross-Validation is better than simple validation because the latter could involve picking parameters overfitted to a particular training/ test set combination.

### 3.5.6 Tuning risk minimization

#### 3.5.6.1 Diagnosis

High generalization error is often due to overfitting or underfitting. It is easy to distinguish the two by comparing the training error (t) with the cross validation error (v) (both of which should not include the prior belief/ regularization component). In the former (high bias case), $t \approx v$ and both are high. In the latter (high variance) case, $v >> t$.
According to the diagnosis, one can determine whether more suitable hypothesis spaces and feature sets should be tried, or whether one can achieve better results using just the currently available data.

#### 3.5.6.2 Picking hyperparameters

Theoretically, these would come from the risk function where some weight is given to the prior belief.
In practice, we often do not know what weight should be assigned to penalty for violation of prior belief. So, $\lambda$ is often learned (ie: the strength of the belief is calculated) using validation.

## 3.6 Combining Decision procedures

Rather than picking $d \in H$ with the minimum risk, it is sometimes possible to combine multiple decision procedures in $H$ to produce a randomized decision procedure with lower risk. This combination may be based on a 'bestness' probability distribution over $H$. Eg: combination of several classifiers or rankers to produce superior classifiers/ rankers.

## 3.7 Offline vs online learning

If new information is provided one at a time, the agent may want to keep improving its decision rule after each new observation. This is online learning.

## 3.8 Interpreting the decision procedure selected

This is important if the decision procedure is being learnt in order to figure out the laws of nature: what genes are triggered in a certain situation? Models learnt from training data is often very hard to interpret/ match with intuition. Sparse solutions are often easier/ more rational to interpret.

# 4 Sample

A statistical population exists. Thence, a sample of $N$ points is drawn, from which we attempt to infer properties of the population.

## 4.1 Properties

Sample statistics are considered elsewhere.

### 4.1.1 Sample bias

Pick 2 people: he is either Indian or Chinese; but there exist over 180 other countries.

### 4.1.2 Completeness, accuracy of examples

It is possible that, some points are not completely specified. For a certain point $X$, the component $X_j$ may not be specified. This ambiguity allows optional auxiliary data to be considered.

### 4.1.3 Independence of data points

In some cases, $\{X_i\}$ are usually iid $\sim D'$. Eg: Predicting whether a visitor to a store is likely to be a customer or a criminal.
In other cases, the input points $\{X_i\}$ are not identically distributed.

#### 4.1.3.1 Sequential

Firstly, sequential data points can be ordered: $(X_i)$. In addition, $X_i \not\vdash X_{i-1}$. So, data points need not be considered sequential merely because of the presence of a Time/ Position feature. Eg: Identifying words with spelling mistakes in a sentence.

#### 4.1.3.2 Adversarial

Or they may be chosen adversarially : See game theory ref.

#### 4.1.3.3 Active choice

Or, as in the case of active learning problems, the learner can take actions to change input distribution. Reinforcement learning is considered in the AI survey.

### 4.1.4 Labeling of the data.

Some features, aka the label, may be a (unknown) function of the others. Thence, deducing dependence of label on other features is the prediction problem.

#### 4.1.5 In case of small sample

Usually insufficient data to guess shape of distribution; can only see large effects: so need large sample to see small effects; only extreme outliers stand out remarkably.

##### 4.1.5.1 Few high dimensional data-points

$D >> N$: so the parameter space to explore is huge: maybe 1 parameter for each feature. Yet want to be able to estimate the parameters generating the observations. This is only possible as there is low dimensional intrinsic structure in the data; can do well using a small hypothesis space.

##### 4.1.5.2 Examples

Take the brain activity vs neuronal activity matrix. Activity-levels of millions of neurons is feature for each data-point. Brain activity is highly defined by a very small number of spiking neurons.
Gene expression vs exterior condition matrix.
Social networks: Individual activity vs group action matrix.

# Part II

# Simplification, exploratory analysis

Whatever the statistical problem is, exploratory analysis is often the first step in solving the problem.

## 5 Data exploration

One often studies the empirical distribution of the data and estimates the central tendency, range, median, mode, characteristics of outliers, number of missing values. It is further described in the distribution structure learning part.

One may also cluster the data.

# 6 Data preparation

## 6.1 Motivation

Data preparation often involves massaging attribute values to fit the requirements of operators/ models. For example, some operators cannot handle continuous values, others cannot handle polynomial values, some have problem with missing data.

## 6.2 Changing the range

Ploynominal features can be converted to binomial features using a binary representation.
Binominal features can be treated as numeric inputs.
Continuous valued features can be converted to discrete valued features by binning them. The bins may be defined by a regular grid on the range, or irregularly to ensure roughly equal cardinality.

## 6.3 Dealing with missing values

Continuous missing values can be replaced with average values or with 0's as appropriate under the circumstances. Nominal missing values can be replaced with the mode. Or missing value may be imputed using various predictive models.

## 6.4 Saling, centering, allowing bias

### 6.4.1 Motivation

To interpret the model parameters learned using the training data (eg: to see which input component has more predictive value, and to deduce its bias), it is often beneficial to apply a transformation to the input vector.
In the discussion below, we assume that the input has undergone this transformation.

### 6.4.2 Centering to 0

Let $\mu$ be the mean of the given training points $\left\{\hat{X^{(i)}}\right\}$. Thence derive the transformation $X = \hat{X} - \mu$.

#### 6.4.3 Scaling

Let $S$ be a diagonal matrix with diagonal elements $1/\sigma_i$, where $\sigma_i$ is the standard deviation of the input component $i$. Thence, get the transformation $SX$.

#### 6.4.4 Constant variable

We assume that the input RV $X$ includes the constant 'variable' $X_0 = 1$.

## 7 Finding a simpler, more useful representation of the data

### 7.1 Feature extraction

Use some $\phi$: d-dim X formed by input vars $\rightarrow$ m-dim Feature space; $\phi$ is a vector function. Basis of feature space are the 'basis functions' ($\phi_i$). [**Incomplete**]

#### 7.1.1 Dimensionality reduction

Remove irrelevant/ less relevant features, merge duplicate features. Eg: synonymous words in documents.

##### 7.1.1.1 Importance

Treating duplicate features as if they were different harms ability to classify or cluster data points. Irrelevent features also harm predictive ability, by acting as noise.

**Data visualization** Project high dimensional data to 2D or 3D space. Also see graph drawing/ embedding.

##### 7.1.1.2 Extant of dimensionality reduction

The best dimension should match the inherent dimensionality of the data.
If there are not-very-useful features, there will be a steep drop in the performance gain by including those features. Often this is not the case, and the choice is rather arbitrary.

### 7.2 Using Kernel function to implicitly map data to a feature space

See vector spaces ref for info about kernels.

### 7.2.1 The Kernel trick

Aka Kernel substitution.
Can reformulate hypothesis model, perhaps a predictor, and any associated optimization objective such that input vector $x$ enters only in terms of inner products. Then can substitute this product with kernel function $k(x, x')$; implicitly using some $\phi(x)$; can try out various kernels to achieve good performance. Can also use +ve semi-definite d*d kernel matrix K to define inner product in feature space.

### 7.2.2 Use

Learning problem; classification or clustering; may be easier to solve in some kernel. Eg: Can't find linear separator for 2 concentric rings of points in original space, but can do so in a high dimensional space.
Save space: If features are high dimensional, may not want to explicitly form feature vectors. Just use the kernel function.

## 7.3 Casting data into a graph

Take data $\{X_i\}$; use similarity measure $S(X_i, X_j)$; Cast into weighted similarity graph: Set $w_{u,v} = f(S(u, v))$. Casting data into a graph simplifies the clustering task: there you only care about the distance.

### 7.3.1 $\epsilon$ neighborhood graph

$(u, v) \in E \equiv d(u, v) < \epsilon$,
where $d(u, v) = g(S(u, v))$. Usually unweighted as $\epsilon$ distance does not matter.

### 7.3.2 k nearest neighbor directed graph

$(u, v) \in E \equiv v \in KNN(u)$.

### 7.3.3 Fully connected graph

Useful if S(u, v) like Gaussian kernel: $e^{-\|u-v\|^2/(2\sigma^2)}$.

## 7.4 Find similar features

### 7.4.1 With Clustering

Coclustering does this.

### 7.4.2 Find covariance of various features

#### 7.4.2.1 Sample points and their mean

Each sample pt $X^{(i)}$ is a $d$ dim vector. The mean, $E[X] = \mu$; sample mean is $\bar{X} = n^{-1} \sum_i X^{(i)}$.

#### 7.4.2.2 Sample Covariance matrix C

Estimates $\Sigma$. So, $C_{i,j} = (n-1)^{-1} \sum_k (X_i^{(k)} - \bar{X}_i)(X_j^{(k)} - \bar{X}_j)$ estimates $cov(X_i, X_j)$. So, $C = (n-1)^{-1} \sum_i (X^{(i)} - \bar{X})^T (X^{(i)} - \bar{X})$.
So, $C$ is symmetric +ve semi-definite; and if $n < d$, $C$ is singular.

## 7.5 Identify a good metric

### 7.5.1 Generalized interpoint distance

Aka Mahalonobis distance. Take covariance matrix C, $E[X] = E[Y] = \mu$; X, Y sample points. $d(X,Y) = ((X-Y)^T C^{-1} (X-Y))^{1/2}$. Negates bias due to having features which mostly say the same thing while finding distance.

# 8 Dimensionality reduction

## 8.1 General motivations

Perhaps one wants to find closest vectors to a given vector - perhaps for the purpose of executing the nearest neighbor algorithm.
Computational efficiency - as in the case of Most variable subspace identification (PCA).
Noise reduction - it could be that many of the features in a vector are not very informative.

## 8.2 Latent factor modeling

### 8.2.1 Problem

Here, one derives generative models to describe affinity of one discrete variable (say $U$) with another (say $V$): eg: features and objects, documents and words. In the process, one derives low dimensional representation of both these entities.

#### 8.2.1.1 Matrix view

Suppose that you are given a matrix $A$ whose entires represent affinities between two types of entities. One may need to find a model the strength of this association which is robust to noise in the observations/ which is succinct.

#### 8.2.1.2 Linear model

A linear model would be: $A_{ij} = \langle u_i, v_j \rangle$, where $u_i$ and $v_j \in R^k$ are low dimensional representations of entities $i$ and $j$. Precisely, given that those entities are represented by $A_{i,:}, A_{:,j}$, we want to find $U^k, V^k$ such that $A \approx U_k^T V_k$.

#### 8.2.1.3 Motivation

So the motivations described for dimensionality reduction in general apply. In addition, this can be used to find unobserved affinities between entities.

### 8.2.2 Matrix factorization by SVD

Taking the top singular vectors, we know that: $\left\| A - U_k \Sigma_k V_k^T \right\|_2^2$ is minimzed. This is a very common form of 'Latent semantic analysis'.

### 8.2.3 Non-negative matrix factorization

There could be other constraints such as requiring that the lower dimensional representations be non negative. Non negative matrix factorization is considered elsewhere.

### 8.2.4 Probabilistic modeling

Probabilistic models for affinities between the two entity types are considered in the probabilistic models survey.

## 8.3 Linear dimensionality reduction

A linear map (Eg: a projector) is used on the data to reduce dimensions. So, ratio of distances amongst points are preserved.

### 8.3.1 Most variable subspace identification

Aka Principal component analysis (PCA).

#### 8.3.1.1 Problem

Suppose that you have a $m \times n$ data matrix $A$ of $n$ $m-$dimensional data points. Suppose that we want a very good low dimensional representation of these data points. We have a bunch of points, and we want to pick $k < m$ orthogonal axes along which the data has the greatest variability.
One can visualize most of the data points as being contained in a m-dimensional hyperellipse, whose top $k$ axes we want to use to represent the data points in a low dimensional space.

#### 8.3.1.2 Motivation

Suppose one needs to compare a $v \in R^m$ with all column vectors in $A$ - as in the case of object matching. Computing $n$ inner product is an $O(m^2 n)$ operation. By dimensionality reduction, we want to turn this into an $O(k^2 n)$ operation. Besides, the latter may be a way to overcome noise - ie ignore unimportant features.

#### 8.3.1.3 Preprocessing, problem statement

One can always get a matrix $B$ with rows centered around the mean. This will enable us to write the covariance matrix as $C = n^{-1}BB^T$. We want to find a linear transformation $L$ such that $\left\|LL^T - BB^T\right\|_F^2$ is minimized.

#### 8.3.1.4 Solution

Using the properties of the SVD. $B = U\Sigma V^*$, $BB^T = U\Sigma^2 U^*$. This covariance matrix can be approximated by $nC_k = U_k \Sigma_k^2 U_k^*$. So, from $\Lambda_k = \Sigma_k^2 = U_k^* BB^T U_k$, we see that the low dimensional transformation of $B$, which ensures that most of the variability in the data is preserved, is given by the orthogonal map $U_k^* B$.

#### 8.3.1.5 Best target dimension k

Sometimes, can pick the top $k$ ev, so that there is a steep gap betweek $\lambda_q$ and $\lambda_{q+1}$.

#### 8.3.1.6 Comments

***So, the top ew/ sw of the covariance matrix define the subspace of highest variability.***

### 8.3.2 Factor analysis

Model thus: $x - \mu = Lf + \epsilon$. $x \in R^d, L \in R^{d*k}$. Reducing $x$ to $k$ dimensional vertex. Ideally, cov(f) = I, $f \perp \epsilon, E[f] = 0$. $L$ is loading matrix, f are factors. Arrange centered data points as columns of X-M. Then, trying to factor this into LF.

## 8.4 Supervised linear dimensionality reduction

Labelled data. Done for classification etc..

### 8.4.1 Linear discriminant analysis

#### 8.4.1.1 The problem

(Fisher) Want to do dimensionality reduction for the purpose of classification. If $x \in R^d$, want to project data points to some k-1 dimensional hyperplane; what is the best hyperplane to do this?
You want to maximize after-projection inter-class scatter: separate means widely, but minimize after-projection intra-class scatter.
Usually k-1 dim hyperplane desired: then you can find a hyperplane between every pair of classes. So you project with $y = W^T x$ for orthogonal $d * (k-1)$ dim W.

#### 8.4.1.2 The solution

Before projection: Take $S_T = \sum_x (x-m)(x-m)^T$: total scatter; $S_W = \sum_{i=1}^k \sum_{x \in C_i} (x-m_i)(x-m_i)^T$: within class scatter matrix; $S_B = \sum_{i=1}^k n_i (m_i - m)(m_i - m)^T$: between class scatter matrix. So, $S_T = S_W + S_B$. Note: $S_B$, the sum of $k$ rank 1 matrices, has rank k-1; so only k-1 ew's are non 0. If $k = 2$, can define differently: $S_B = (m_1 - m_2)(m_1 - m_2)^T$.
After projection scatters will be: $S'_W = W^T S_W W, S'_B = W^T S_B W$. Find $\max_W \frac{|W^T S_B W|}{|W^T S_W W|}$ or maybe $\max_W tr((W^T S_W W)^{-1}(W^T S_B W))$, with W having (k-1) independent columns. Maximized by top k-1 generalized ev: see linear algebra ref.
If $S_W$ is invertible, same as ev problem $S_W^{-1} S_B x = \lambda x$: solution is the top k-1 ev; but matrix is assymetric, so finding ev harder. If just projecting to a line, can also get problem $S_W^{-1/2} S_B S_W^{-1/2} x = \lambda x$. See linear algebra ref.

## 8.5 Non-linear dimensionality reduction

### 8.5.1 Kernel PCA

Represent data in feature space corresponding to some kernel, use PCA.

### 8.5.2 Manifold learning

Consider the swiss roll/ tissue paper roll dataset. This is a 2 dimensional manifold in a 3-dim space.
[**Incomplete**]

### 8.5.3 Measuring goodness

How well are k-NN properties preserved? Neighborhood rank preserved even if magnitude is not.

# 9 Cluster data points

## 9.1 The clustering problem

Given $N$ points, want $k$ clusters. Often, $k$ is not known.

### 9.1.1 Use

Summarizing data is important for generalization, understanding future data points.
Supplying labels to unlabeled data; thence classifying new data-points. Eg: Face recognition using Eigenfaces. Use to estimate distribution support and thence in novelty detection.

### 9.1.2 Criteria: Continuity vs compactness

Consider a starfish: The continuity critierion will identify the 5 arms as 5 clusters, but the comactness criterion will fail. Consider points produced from two gaussians with distinct centers: the compactness criterion appears better.

### 9.1.3 Extensions

Coclustering data-points along with features.

#### 9.1.3.1 Find Non-redundant clusterings

Aka Disparate clusters. Want clusterings based on unrelated criteria. Eg: can classify people based on sex, race etc.. $k$ Disparate clusters can be thought of as lying in $k$ orthogonal subspaces.

#### 9.1.3.2 With background clutter

Consider clustering stars in the night sky to find galaxies. Important for clustering algorithm to ignore clutter.

### 9.1.4 Evaluation of clustering

In case the $k$ true labels are known: Just use ways of evaluating classification. Can always do this by generating artificial data.

### 9.1.5 Challenges

#### 9.1.5.1 Curse of dimensionality

As dimensions/ features grow, more difficult to group similar things together; there could be irrelevant or noisy features.
Use dimentionality reduction techniques.

#### 9.1.5.2 Number of clusters k

The actual number depends on the data. Choice of $k$ is mostly based on heuristics.
Some methods take $k$ as input, others discover $k$ themselves.

#### 9.1.5.3 Identify important clusters

Clusters which matter remain distinct at different levels of coarsity.

**Cluster visualization** Look at clusters in 2D or 3D at varying coarsity levels to identify important clusters.

### 9.1.6 Approaches

#### 9.1.6.1 Views of the data

Visualize data points as points in feature space.
Or as 'data points (X) vs features (Y)' matrix A. In case features are binary (eg: word present or absent), get contingency table P(X, Y), an estimate of the joint probability matrix.

#### 9.1.6.2 Density estimation

**Parametric** Try to find distribution of data in the input space. Usually fit some multimodal distribution: the different modes define different cluster representatives.
Eg: Fit a mixture of $k$ Gaussians centered at various spots in the input space.
Advantage of having a parametric model: can extrapolate what the data will look like with more sampling.

**Non-Parametric** Can do non-parametric density inference, then do density shaving: ignore points with low density, identify the modes.

**Relative performance** Non parametric methods usually perform better, given enough data-points.

#### 9.1.6.3 Centroid based vs agglomerative clustering

Described fully in another section.
Centroid based clustering methods are usually fast: the costliest step is often assignment of points to clusters: O(kn). Agglomerative methods usually involve comparison between all cluster-pairs for the purpose of agglomeration; so are slower: $O(n^2)$.
Centroid based methods require $k$ to be known before hand, they need initial points. Agglomerative methods find varying number of clusters: from $N$ to 1; it is up to the user to know where to stop - this can be difficult.

## 9.2 Agglomerative clustering

Bottom up approach. Start off with $N$ clusters: 1 for each point; pick nearest pair of clusters; merge them; repeat till you have $k$ clusters.

### 9.2.1 Intercluster metrics

Distance between means. Or between closest pair of points: tends to produce elongated clusters: clustering by continuity criterion. Or between farthest pair of points: clustering by compactness criterion. These correspond to the 2 clustering criteria.

## 9.3 Centroid based clustering

Use centroids or cluster representatives to cluster.

### 9.3.1 Mean: Best cluster representative wrt Bregman div

Given Bregman div $d_f$ based on convex function f. Show by easy algebra that $n^{-1}\sum_i d_\phi(X_i, z) - n^{-1}\sum_i d_\phi(X_i, \mu) = d_f(z, \mu) \geq 0$. So, mean is best cluster representative wrt 2 norm: 2-norm is also a bregman divergence.

### 9.3.2 k means clustering

#### 9.3.2.1 Objective

**As minimizing within cluster scatter** Find $k$ centroids, make $k$ partitions (Vorinoi tesselations) in the input space:
$S' = (S'_i) = argmin_S \sum_{i=1}^{k} \sum_{x_j \in S_i} d(x_j, \mu_i)$.

**As maximizing inter-cluster scatter** Use scatter matrices/ scalars $S_B, S_W, S_T$ as in LDA. For any bregman divergence: $S_T = S_B + S_W$. Implicitly tries to maximimze $S_B$ : Between cluster scatter.

#### 9.3.2.2 Algorithm

Start of with $k$ vectors $(m_i^0)$ as means of $(S_i)$; At time t, you have: $(m_i^t)$. Reassign all points to the $S_i^t$ corresponding to the closest $m_i^t$; calculate new means $(m_i^{t+1})$ as the centers of these $(S_i^t)$; repeat.
If $d$ is any Bregman div, $k$ means minimizes this at each iteration: Alg finds better clustering, Mean is best cluster representative.
Time: O(kndt): very fast.

#### 9.3.2.3 As low rank factorization with alternating minimization

Let X be the $d \times n$ data matrix. Doing $X \approx MW$, where M is the $d \times k$ means matrix, and $W \in \{0,1\}^{k \times n}$ denotes membership. For strict partitioning, there is the constraint $w_i \in I_k$.
So, k-means is equivalent to solving $\min_{M,W} \|X - MW\|_F$ by alternatively minimizing M with W fixed, and W with M fixed subject to constraints on W.

#### 9.3.2.4 Drawbacks and extensions

This is a greedy algorithm, does local minimization of the objective function. Highly sensitive to initial conditions. Can end up with empty clusters, with bad initialization. So, have varied initialization strategies.
Fails to cluster data points arranged in concentric rings. So use the kernel trick here: get kernel $k$ means.

### 9.3.3 With GMM

You model the observed data with $k$ normal distributions $(D_i)$ specified by means $(\mu_i)$, covariances $(\Sigma_i)$, prior probabilities of a point being generated by $(D_i)$, aka mixture weights, $(p_i)$. If you find the best parameters for this model, you can assign points to the cluster associated with the $D_i$ most likely to have generated it.
Start with some guessed parameters. Repeat the following steps iteratively: a] Assign each point to the $D_i$ most likely to have generated it: do $\min_i(\log p_i)(x - m_i)^T\Sigma_i^{-1}(x - m_i)$: same as minimizing a weighted 'Mahalonobis distance'; $(\log p_i)$ can be seen as shrinking $D_i$ appropriately by acting on $\Sigma_i^{-1}$. Let $n_i$ count the points assigned to cluster i. Update $p_i = n_i/n$. b] Update parameters: $\mu_i = n_i^{-1}\sum_{x_j \in i} x_j; \Sigma_{i,(j,k)} = \frac{1}{n_i - 1}(x_i, j - \mu_j)(x_i, k - \mu_j)$: note unbiased estimater used in estimating covariances.

#### 9.3.3.1 Generalizing k-means

k-means corresponds to GMM clustering with each Normal Distribution in the model constrained to being spherical: at each step you assign the point to the cluster with $\mu = \arg\min_{m_i}(x - m_i)^T(x - m_i)$.

### 9.3.4 With non-negative matrix factorization

Let X be the $d \times n$ data matrix. Doing $X \approx MW$, where $M \in R_+^{d \times k}$ means matrix, and $W \in R_+^{k \times n}$ denotes membership strength.

### 9.3.5 Finding the initialization points

Bad initialization points can lead to bad clusters, good ones lead to good clusters. Density estimation useful here. [**Incomplete**]

## 9.4 Co-clustering

Cluster both the rows and columns of P simultaneously. Thus dealing with duplicate/ synonymous/ irrelevant features simultaneously while clustering.

### 9.4.1 Objective: Information loss minimizing

Find maps $C_X : \{x_1, ..x_m\} \rightarrow \{\hat{x_1}, .., \hat{x_k}\}$;
$C_Y : \{y_1, ..y_m\} \rightarrow \{\hat{y_1}, .., \hat{y_l}\}$. $(C_X, C_Y)$ is a coclustering; yields corresponding joint distribution matrix $p(\hat{X}, \hat{Y})$. Best co clustering has minimum mutual information loss: $\min I(X;Y) - I(\hat{X}, \hat{Y}) = K(p(X,Y)||q(X,Y))$ where $q(X,Y) = p(\hat{X}, \hat{Y})p(X|\hat{X})p(Y|\hat{Y})$.

#### 9.4.1.1 The monotonic optimizer

$K(p(X,Y)||q(X,Y)) =$
$K(p(X,Y,\hat{X},\hat{Y})||q(X,Y,\hat{X},\hat{Y}))$.
For any clustering, q preserves marginals and conditionals:
$q(\hat{x},\hat{y}) = p(\hat{x},\hat{y}), q(x,\hat{x}) = p(x,\hat{x}), q(y,\hat{y}) = p(y,\hat{y}), p(x) = q(x), p(x|\hat{x}) = q(x|\hat{x})$ etc..
So, $K(p(X,Y,\hat{X},\hat{Y})||q(X,Y,\hat{X},\hat{Y})) =$
$\sum_{\hat{x}} \sum_{x:C_X(x)=\hat{x}} K(p(Y|x)||q(Y|\hat{x}))$;
similar form in terms of $K(p(X|y)||q(X|\hat{y}))$.
Thence information theoretic coclustering alg: Start with $C_X^{(0)}, C_Y^0$; at step t, for each row $x$, set $C_X^{(t+1)}(x) = argmin_{\hat{x}} K(p(Y|x)||q^{(t)}(Y|\hat{x}))$; recompute distributions $q^{(t+1)}$; at step t+2 similarly recluster columns finding local minima; repeat. This minimizes objective function monotonically. Experiments on document clustering tasks show better clustering than 1D clustering.

## 9.5 Using Graph clustering

For graph clustering methods, see graph theory ref.

# Part III

# Distribution structure learning

## 10 Problems

### 10.1 Conditional distributions and notation

Got observations of events: RV $X$ took values $\{x_i\}$, deduce/ model the process causing those events. In general, we want to model the conditional distributions $f_{X_r|X_{\neg r}}$. Often, we use the alternate notation $Y = X_r$ and $X = X_{\neg r}$.

### 10.2 Connection to modeling marginal density

Note that $X_{\neg r}$ may be empty, so that marginal/ unconditional distribution modeling - which is estimating $f_X(X)$ - is a special case of conditional distribution modeling.
Techniques which are suitable for modeling conditional distributions can be directly applied to such special cases. Techniques specialized for modeling unconditional distributions can be applied to modeling one conditional distribution $f_{X_r|X_{\neg r}=x_{\neg r})}$ at a time.

#### 10.2.1 Problem structure

For discrete probability distribution $p$, valid values of $p$ form the probability simplex. Also, expectation is linear in $p$, variance is concave in $p$. So, can specify many convex optimization problems using these constraints.

## 11 Estimating parameters

### 11.1 Estimate parameters using statistics

The distinction between choosing parametric and non-parametric approaches are considered in the decision theory section.

#### 11.1.1 Statistic, estimator

A statistic $\hat{t} = \hat{g}(X)$ is a function of the sample X; an observable random variable. When it is used to estimate some parameter, it is called an estimator. t can be estimated by estimating $\theta$.

##### 11.1.1.1 Point estimation of the parameter

If $\hat{t}$ tries to approximate $t$, it is an estimator.

### 11.1.2 Distribution of a statistic

Aka Sampling distribution. Standard deviation of sampling distribution called standard error.
Find by manual calculation of probabilities of values of $\{X_i\}$; or by simulation or assume $\{X_i \sim N(\mu, \sigma^2)\}$, using mgf $n^{-1} \sum Y_i \sim N(\mu, \sigma^2/n)$.

### 11.1.3 Summarize Central tendency

Sample and population expectation
$(\bar{X}, \mu = E[X])$, median (m with $F(m) = 1/2$), mode.

#### 11.1.3.1 AM, GM, HM

Suppose we have $n$ numbers $(a_i)$. Arithmetic mean is $n^{-1} \sum_i a_i$: the name reminisces the arithmetic series. Geometric mean is $\prod_i a_i^(n^{-1})$, and harmonic mean is $n^{-1}(\sum_i a_i^{-1})^{-1}$.
Using weights $p_i \in [0, 1]$ such that $\sum_i p_i = 1$, these quantities can be generalized to define weighted arithmetic, geometric and harmonic means.
$\mu \geq GM \geq HM$. This and other inequalities are considered in the complex analysis survey.

#### 11.1.3.2 Modeling accuracy

If one is trying to use the sample mean to quantify the 'average' phenomenon, it is important to pick the right random variable.
From Tao: 'For instance, consider the question of what the population density of the United States is. If one does a simple average, dividing the population of the US by the area of the US, one gets a density of about 300 people per square mile, which if the population was spread uniformly, would suggest that each person is about 100 yards from the nearest neighbour. Of course, this does not conform to actual experience. It is true that if one selects a random square mile patch of land from the US at random, it will contain about 300 people in it on the average. However, not all such patches are equally inhabited by humans. If one wants to know what what density the average human in the US sees, rather than the average square mile patch of land, one has to weight each square mile by its population before taking an average. If one does so, the human-weighted population density now increases to about 1400 people per square mile - a significantly different statistic.'

#### 11.1.3.3 Combining arithmetic means of subpopulations

(From Tao's blog.) When combining averages of small sub-populations together to form an average of the combined population, one needs to weight each sub-average by the sub-population size in order to not distort the final average. If the sub-populations being averaged over vary, this can then lead to Simpson's paradox.

Eg: it turns out that in most departments, women had a slightly higher success rate in their applications than men, but in the university as a whole, women had a lower success rate. The ultimate reason for this was that women tended to apply to more competitive departments, which lowered their overall average success rate.

### 11.1.4 Other statistics and parameters

#### 11.1.4.1 Summarize variability or dispersion

Sample and population Variance
$(S^2, \sigma^2)$, standard deviation $(S, \sigma)$. Also, range: max - min.

#### 11.1.4.2 Order statistics

max or nth order statistic $X_{(n)}$, min or first order statistic $X_{(1)}$, kth smallest sample point $X_{(k)}$.
$f_{X_{(k)}}(x) = \frac{n!}{(k-1)!(n-k)!} F_X(x)^{k-1}(1 - F_X(x))^{n-k} f(x)$: consider ways of selecting the kth smallest sample point while ignoring ways of ordering the rest, probability of k-1 of them being smaller and n-k being larger. Treat like any pdf; can find corresponding cdf by integration.
Also, $f_{X_{(j)},X_{(k)}}(x, y) =$
$\frac{n!}{(j-1)!(k-j-1)!(n-k)!} F_X(x)^{k-1}(F(y) - F(x))^{k-j-1}(1 - F_X(x))^{n-k} f(x)f(y)$.

#### 11.1.4.3 Other statistics

Proportion $\bar{p}$, p: $n\bar{p} \sim bin(n, p) \to N(np, np(1-p))$; $\bar{X_1} - \bar{X_2}, \mu_1 - \mu_2$; $\bar{p_1} - \bar{p_2}, p_1 - p_2$. Good pivotal quantity for these: $\frac{\hat{t}-t}{\sigma_{\hat{t}}} \sim N(0, 1)$.

## 11.2 Estimator properties

Properties of good estimators: low bias, low variance, completeness, consistency, sufficiency.

### 11.2.1 Bias

$B(\hat{t}) = E[\hat{t}] - t$. Easy to find unbiased estimator given biased estimator.

### 11.2.2 Mean square error: Bias variance decomposition

$mse(\hat{t}) = E[(t - \hat{t})^2] = var[\hat{t}] - B(\hat{t})^2$ using bias definition.

### 11.2.3 Relative efficiency of unbiased estimators

$eff(\hat{t}_1, \hat{t}_2) = var[\hat{t}_2]/var[\hat{t}_1]$. To compare variances of estimators.

### 11.2.4 Consistency of unbiased estimators

$\hat{t}_n$: derived from sample of size n. Consistent if $\forall \epsilon : lt_{n\to\infty} Pr(|\hat{t}_n - t| \leq \epsilon) = 1$. Using Chebyshev's thm, consistency if $lt_{n\to\infty} var[\hat{t}_n] = 0$. ie: $\hat{t_n} \to_p t$: $(\hat{t_n})$. Let $\hat{t_n^{'}} \to_p t'$. $\hat{t} + \hat{t'} \to_p t + t$; $\hat{t}\hat{t'} \to_p tt'$; $\hat{t}/\hat{t'} \to_p t/t'$. Also, if $g() \to R$ continuous, $g(\hat{t_n}) \to g(t)$.

### 11.2.5 Sufficiency of unbiased estimator

#### 11.2.5.1 Motivation from MLE

Given sample vector $X = x$, suppose that we want to estimate $T = t$ using estimator $\hat{T} = \hat{t}$. So, we want to find $t$ maximizing $f_{T|X}(t|x) = \frac{f_{X|T}(x|t)f_T(t)}{f_X(x)}$.
We may use MLE (assuming all $t$ equally likely), find t maximizing $L(t|x) = f_{X|T}(x|t) = f_{X|\hat{T},T}(x|\hat{t},t)f_{\hat{T}|T}(\hat{t}|t)$.
So, if $f_{X|\hat{T},T}(x|\hat{t},t)$ is independent of $T$: $f_{X|\hat{T},T}(x|\hat{t},t) = f_{X|\hat{T}}(x|\hat{t})$, same as maximizing $f_{\hat{T}|T}(\hat{t}|t)$; can discard $X = x$ after getting $T = t$. So, this sufficient statistic summarizes all info in a sample about a parameter.
Not necessarily unbiased. All good estimators, which are unbiased, are functions of sufficient statistic.

#### 11.2.5.2 To show sufficiency if distribution family known

Show $f_{X|\hat{T},T}(x|\hat{t},t) = f_{X|\hat{T}}(x|\hat{t})$; its form is not a function of t. So, show factorization $f_{X|T}(x|t) = f_{X|\hat{T},T}(x|\hat{t},t)f_{\hat{T}|T}(\hat{t}|t) = g(\hat{t})f(\hat{t},t)$; Or show $\frac{f_{X|T}(x|t)}{f_{\hat{T}|T}(\hat{t}|t)}$ not a function of t.

#### 11.2.5.3 To find sufficient statistic

Start with $f_{X|T}(x|t)$, factorize it into
$g(...)f(\hat{t},t)$; in the part which is a $f(..,t)$, find sufficient statistic $\hat{t} = h(X)$. Eg: $f(..,t) = e^{-\sum X_i}$, $\sum X_i$ be the minimal sufficient statistic; if $f(..,t) = e^{-\sum X_i - \sum X_i^2}$ $\sum X_i$ and $\sum X_i^2$ are joint sufficient statistics.

### 11.2.6 Statistical efficiency

How many observations do you need to get error $d(\hat{t},t) < \epsilon$?

## 11.3 Find estimator for some parameter

Also see model selection techniques: where you estimate all parameters which define the model from the data.

### 11.3.1 From sufficient statistic

Find sufficient statistic, then construct unbiased estimator as a function of it.

### 11.3.2 Minimum variance unbiased estimator (MVUE)

(Rao-Blackwell)
For parameter t, take any estimator $\hat{t}$, sufficient statistic U; $\hat{t'} = E[\hat{t}|U]$ (Rao-Blackwellization); then $E[\hat{t'}] = t, var[\hat{t'}] \leq var[\hat{t}]$: $var[\hat{t'}] = E[E[\hat{t}|U]^2] - t^2 \leq E[E[\hat{t}^2|U]] - t^2 = var[\hat{t}]$. Rao Blackwellization can't improve variance further. So, an unbiased estimator which is a function of the sufficient statistic yields MVUE. To find MVUE do this.

### 11.3.3 Method of moments

Assume sample moments are good estimators of population moments. Set $E[Y|\theta] = \frac{\sum Y_i}{n}$; thence get $f(\theta) = \frac{\sum Y_i}{n}$; then solve for $\theta$. If number of parameters in $\theta$ is high, find other moments: $E[Y^k|\theta] = \frac{\sum Y_i^k}{n}$. Result is usually consistent, but not always sufficient.

## 11.4 Confidence Interval

### 11.4.1 Definition

Find intervals of probable values: confidence intervals $(\hat{t_1}, \hat{t_2}) : Pr(\hat{t_1} \leq t \leq \hat{t_2}) \leq 1-a$. Can also use one sided intervals using only upper or lower confidence limits: $(-\infty, \hat{t_2}), (\hat{t_1}, \infty)$.
Contrast with point estimation.

### 11.4.2 General procedure

#### 11.4.2.1 Pivotal quantity for estimate

Suppose you have a point estimator $\hat{t}$, which can be expressed as a function of some pivotal quantity $q$. This quantity has the following property: $q = g(t, ..)$, which is a function of $t$ whose distribution function does not depend on $t$, but may depend on other known/ guessable parameters - like sample size.
Sometimes, the pivotal quantity is the estimator itself.

#### 11.4.2.2 Procedure

For a given sample, one can find $q$ ***- the pivotal quantity***, find or bound its distribution function $p$ and finally find suitable confidence interval $q \in (a, b)$ which translates to the confidence interval estimate: $t \in (\hat{t_1}, \hat{t_2})$.

### 11.4.3 Pivotal quantity deviation bounds

Theoretical calculations can provide deviation bounds for the pivotal quantity distribution - Eg: the use of Chernoff deviation bounds in case of binomial random variables.

#### 11.4.3.1 By repeated sampling

If one can sample repeatedly from the actual distribution, one can estimate the confidence interval for a given estimator.

#### 11.4.3.2 By Bootstrap sampling

**Process** If one is not able to take repeated samples from the actual distribution, one can repeatedly sample (with replacement) from a uniform distributions over the available sample set.
The justifcation is that this distribution is close to the original distribution, so conclusions drawn from it are not too erroneous.

**Properties** [**Incomplete**]

### 11.4.4 Pivotal quantity for ratio of variances

With F sampling distribution: $\frac{S_1^2\sigma_2^2}{S_2^2\sigma_1^2} \sim F_{n_1-1,n_2-1}$.

# 12 Mean, variance of real valued RV

## 12.1 Mean: estimation

### 12.1.1 Consistency

Aka Law of large numbers
Let $\{X_i\}$ iid. $\hat{X}_n = n^{-1}\sum^n X_i$. As $var[\hat{X_n}] = \sigma^2/n \to 0$ as $n \to 0$, Weak law: $\hat{X}_n$ is a consistent estimator of $\mu$.

### 12.1.2 Normalness of estimator distribution

Aka Central limit theorem (CLT)
Take estimator $U_n = \frac{\bar{X}-\mu}{\frac{\sigma}{\sqrt{n}}}$. $lt_{n\to\infty} Pr(U_n \le u) = \int_{-\infty}^{u} \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt$: so approaches CDF of N(0,1): See convergence of moment generating function below. So, as n increases, $var[\bar{X}]$ becomes smaller: visualize pdfs of $X, \bar{X}_{30}, \bar{X}_{50}$; see how curve becomes more normal and gets thinner and taller. Generally, can use CLT when $n > 30$.

#### 12.1.2.1 Proof showing MGF $M_{U_n}(t) \to$ MGF of N(0, 1)

iid $\{X_i\}$. $m_{U_n}(t) = E[e^{\frac{t(\sum X_i - n\mu)}{\sqrt{n}\sigma}}] = \prod E[e^{\frac{t}{\sqrt{n}}(\frac{X_i-\mu)}{\sigma}}] = m_Z(t/\sqrt{n})^n$: implicitly defining Z with $E[Z] = 0, var[Z] = E[Z^2] = 1$.
But, by Taylor, $m_Z(t/\sqrt{n}) =$
$m_Z(0) + m'_Z(0)(t/\sqrt{n}) + m''_Z(h)(t/\sqrt{n})^2(1/2!)$
$= 1 + E[Z]t + m''(h)(\frac{t^2}{2n})$ for some $h \in (0, t/\sqrt{n})$; so $m_Z(t/\sqrt{n}) = 1 +$

$m''(h)(\frac{t^2}{2n}) \to 1 + \frac{t^2}{2n}$ as $n \to \infty$. So, $m_{U_n}(t) \to (1 + \frac{t^2}{2n})^n \to e^{t^2/2}$, MGF of N(0, 1).

### 12.1.3 Normal distr: Pivotal quantity to estimate mean

Student's t distribution used to estimate $\mu$ when distribution is assumed to be Normal, n is small and $\sigma$ is unknown. Tables only go up to n = 30 or 40. If $\sigma$ were known, would use normal distribution, or if $n > 30$ would estimate $\sigma$ and use normal distribution tables.
As $(n-1)\frac{S^2}{\sigma^2} \sim \chi^2_{n-1}$, $\sqrt{n}\frac{\bar{X}-\mu}{S} \sim t_{n-1}$. [**Incomplete**]

### 12.1.4 Goodness of empirical estimate

Can apply Chernoff bounds and Azuma Hoeffding inequality etc.. to judge goodness of empirical estimate.

## 12.2 Variance estimation

### 12.2.1 The biased and unbiased estimators

$S^2 = n^{-1}\sum(X_i - \bar{X})^2$ biased: $B[S^2] = n^{-1}E(\sum X_i^2 - 2\bar{X}\sum X_i + n\bar{X}^2) - \sigma^2 = n^{-1}(nE[X^2] - 2E[n\bar{X}^2] + nE[\bar{X}^2]) - \sigma^2 = n^{-1}(n\sigma^2 + n\mu^2 - nvar[\bar{X}] + n\mu^2) - \sigma^2 \to n^{-1}(n-1)\sigma^2 - \sigma^2$ from central limit thm. So, defined as $S^2 = (n-1)^{-1}\sum(X_i - \bar{X})^2$ to get unbiased estimator. Difference small as $n \to \infty$.

### 12.2.2 Normal distr: Pivotal quantity to estimate variance

$N(\mu, \sigma^2)$ assumed. If $S^2 = \frac{\sum(X_i-\bar{X})^2}{n-1}$, then $(n-1)\frac{S^2}{\sigma^2} \sim \chi^2_{n-1}$ [**Find proof**]. So, can use this as pivotal quantity.

## 12.3 Sequential data Sample statistics

### 12.3.1 k-step Moving averages

Suppose that $ran(X_i) \in R$, and that the sample size is $n$.

#### 12.3.1.1 Simple moving average

This is simply the mean of the last $k$ $X_i$.

#### 12.3.1.2 Exponential Weighed

Here, one uses an exponentially decreasing weight (with decreasing $i$) while taking a weighted average of $k$ $X_i$.

#### 12.3.1.3 Applications

This is useful while predicting stock prices, for example.

# 13 Density estimation

## 13.1 Importance

Fitting a model to observations, ie picking a probability distribution from a family of distributions, is an important component of many statistics tasks where one reasons about uncertainty by explicitly using probability theory. Such tasks are labeled 'Bayesian inference'.

## 13.2 Choosing the distribution family

### 13.2.1 Observe empirical distribution

Draw a bar graph, see what the curve looks like.

### 13.2.2 Given expected values of fns $\{E[\phi_i(X)] = \mu_i\}$ and a base measure h

Suppose we want to modify h as little as possible, under KL divergence, so that it has $E[\phi(X)] = \mu$. If h is U, then this is same as finding a maximum entropy distribution with $E[\phi(X)] = \mu$. Then, the solution belongs to the exponential family generated by h and $\phi()$: see probabilistic models ref.

### 13.2.3 Given dependence among features

Use graphical models - see probability ref.

## 13.3 Parametric density estimation

Described in a separate chapter.

## 13.4 Non parametric Probability Density estimation

Estimate distribution on input space using $N$ samples $(x_i)$, without limiting attention to a certain set of distributions.

### 13.4.1 Histogram and the Kernel histogram

A distribution from bar-graph of frequency vs input interval. Can simply use a histogram.

### 13.4.2 Kernel density estimation

(Parzen). $p(x) = \frac{1}{Nh}\sum_{i=1}^{N} K(\frac{x-x_i}{h})$, for kernel K. A smoothening of the histogram using kernels.

#### 13.4.2.1 Kernel function for density estimation

Non negative real valued integrable K(x) satisfies: $\int_{-\infty}^{+\infty} K(u)du = 1$ (ensures PDF qualities during density estimation); $K(-u) = u$ (ensures mean of the PDF is the data point during density estimation). So, K(x) akin to kernel of an integral operator: see functional analysis ref.

#### 13.4.2.2 Using Gaussian radial basis functions

Aka Gaussian kernel.
$K(u) = \frac{1}{2\pi}e^{-\frac{u^2}{2}}$: Gaussian function with mean 0, variance 1.
Taking 1 Gaussian distribution/ adding 1 bump for each data point. h, controlling the variance of the bump, called the smoothing parameter/ bandwidth.
***Can approximate any distribution by mixture of Gaussians!***

## 13.5 Estimate probability measures

### 13.5.1 Use empirical measures

#### 13.5.1.1 Empirical measure

The estimated measure using $n$ samples for a given function is $v_n(f|D) = n^{-1}\sum I_f(X_i)$, where $D = \{X_i\}$ is a set of samples drawn iid from $v$.

#### 13.5.1.2 Goodness of estimate: single event

By law of large numbers, as $\lim_{n\to\infty} E[\frac{\sum f(X_i)}{n}] = E_X[f(X)] = v(f)$. Also, we can use the Hoeffding inequality (see probabilistic models survey) to bound $Pr_D(|v(f) - v_n(f|D)| \geq \epsilon)$ and see that it decreases exponentially with increasing $n$ and $\epsilon$.

**Bound variability in estimate** From central limit theorem, we know that, as $n \to \infty$, the sample distribution approaches $N(v(f), \frac{\sigma}{\sqrt{n}})$.
Also, we can use: $Pr_{D,D'}(|v_n(f|D) - v_n(f|D')| \geq \epsilon) \leq Pr_{D,D'}(|v_n(f|D) - v(f)| \geq \epsilon/2 \vee |v_n(f|D') - v(f)| \geq \epsilon/2) = 2Pr(|v_n(f|D) - v(f)| \leq \epsilon/2)$ and use the Hoeffding inequality again.

#### 13.5.1.3 Goodness of estimate for a class of events

Let $F = \{f\}$ be a class of events (or binary functions) defined on the input space $X$, on which $v$ is a measure. Let $E_F(m)$ be max dichotomy count: see boolean functions survey.

(Vapnik, Chervonenkis).
Then $Pr(\sup_{f\in F}|v_n(f|D)-v(f)|>\epsilon)\leq 8E_F(n)exp(\frac{-n\epsilon^2}{32})$.
[**Proof**]: If we were to use the union bound and the Hoeffding inequality naively, we would have a factor of $2|F| >> 8E_F(n)$ on the RHS. So, we want to get to a point where we need only take the union bound over a small subset of $F$.
So, first we show that
$Pr_D(\sup_{f\in F}|v_n(f|D)-v(f)|>\epsilon)\leq 2Pr_{D,D'}(\sup_{f\in F}|v_n(f|D)-v_n(f|D')|>\epsilon/2)$, for sample set $D'$ acquired in the same way as $D$. Lemma [**Proof**]: $Pr_{D,D'}(\sup_{f\in F}|v_n(f|D)-v_n(f|D')|>\epsilon/2)\geq Pr_D(\sup_{f\in F}|v_n(f|D)-v(f)|>\epsilon)Pr_{D'}(|v_n(f|D')-v_n(f)|<\epsilon/2||v_n(f'|D)-v(f')|>\epsilon)$; and the latter factor can be seen to be small: $<1/2$ using Chebyshev inequality. This is called the 'ghost sample technique'. □
So, now we need only bound $Pr_{D,D'}(\sup_{f\in F}|v_n(f|D)-v_n(f|D')|>\epsilon/2)$. One way to deal with this is to take the union bound now over the set of all dichotomies induced over $D\cup D'$ to get the bound: $E_F(2n)Pr(D,D')(|v_n(f|D)-v_n(f|D')|>\epsilon/2)$, which can then be bounded as described elsewhere. □

**Sharper bound** The sharper bound we require can be obtained using a different analysis, involving $E_F(n)$ instead. The process of picking $D, D'$ and calculating $n^{-1}|\sum_i f(X_i)-\sum_i f(X_i')|$ is equivalent to the process of picking $D, D'$, and then picking $n$ bits $s$, and then finding $n^{-1}|s_i(\sum_i f(X_i)-\sum_i f(X_i'))|$ : in other words, we do this $n$ times: pick a pair of points and assign it to $D$ and $D'$ at random. So, we bound $Pr_{D,D',s}(\sup_{f\in F}n^{-1}|s_i(\sum_i f(X_i)-\sum_i f(X_i'))|>\epsilon/2)\leq Pr_{D,D',s}(\sup_{f\in F}[|n^{-1}\sum_i s_i f(X_i)|\geq\epsilon/4\vee|n^{-1}\sum_i -s_i f(X_i')|\geq\epsilon/4])=2Pr_{D,s}(\sup_{f\in F}|n^{-1}\sum_i s_i f(X_i)|\geq\epsilon/4)$. A union bound over $E_F(n)$ members of $F$ and a Hoeffding inequality can now be applied to bound this.

### 13.5.2 Estimate CDF using empirical CDF

(Glivenko Cantelli). Pick $\{Z_i\}\sim F$, the CDF.
Then $Pr(\sup_z|F_n(z)-F(z)|>\epsilon)\leq 8(n+1)exp(\frac{-n\epsilon^2}{32})$. Pf: Apply VCD theorem with open intervals as classifiers.
Answer to: What $n$ do you need to achieve low error? Thence Borel - Cantelli: $\lim_{n\to\infty}\sup|F_n(z)-F(z)|=0$ with probability 1.

## 14 Parametric density estimation

## 14.1 Problem and solution ideals

### 14.1.1 Density estimation using a distribution class

Suppose that parameter $t$ specifies the distribution $f_t(x_r|x_{\neg r})$, where $x_{\neg r}$ can be empty! Let $T$ be parameter space spanned by such $t$; it represents the class of distributions which can be specified in this form.

Given finite data set $\left\{x^{(i)}\right\}$, we want to approximate an unknown target distribution $D(x_r|x_{\neg r})$ which may not belong to this distribution family using $T$. The approximation can be a weighted combination of combination of distributions in $T$.

#### 14.1.1.1 Related problems

Note that estimating $t$ which is good at predicting the value of $x_r$ given $x_{\neg r}$ by doing $h(x_{\neg r}) = \max_{x_r} f_t(x_r|x_{\neg r})$ is a separate problem, where a different estimation procedure which minimizes the classification error $Pr(h(x_{\neg r}) \neq x_r)$ (corresponding to the 0/1 classification loss) may be used.

### 14.1.2 Solution ideas

Empirical risk minimization, for various forumulations of risk functions which in someway also incorporate prior belief about the best $t$.

## 14.2 Approximation with Normal distribution

Aka Laplace approximation. Suppose that we have the probability distribution $p(x) = Z^{-1}f(x)$.

### 14.2.1 Algorithm

Here, one finds a mode / strict local maximum $x'$ of the distribution $p(x)$, which corresponds to a mode of $f(x)$, using numerical techniques. Then, one creates a normal distribution $N(\mu = x', \sigma)$ around this point.

#### 14.2.1.1 2nd order approximation of log f

$\log N(x', \Sigma)$ is a quadratic function. So, we try to find $\Sigma$ such that $\log N$ approximates the 2nd order approximation of $\log f$.
As $\nabla f(x') = 0$, taking the quadratic approximation
$\log f(x) \approx \log f(x') + (f(x'))^{-2}(x - x')^T \nabla^2 f(x')(x - x')$. So,
$f(x) \approx f(x')exp((f(x'))^{-2}(x - x')^T \nabla^2 f(x')(x - x'))$
The RHS can now be used to construct $\Sigma$. As $x'$ is a mode,
$-2(f(x'))^{-2}\nabla^2 f(x') \succ 0$ as expected.

### 14.2.2 Properties

#### 14.2.2.1 Estimating Z

Z can be approximated to equal the normailizer of N, which is easily calculated. Note that knowledge of Z is not required to get the approximation $N$.

#### 14.2.2.2 Non-uniqueness

Different modes yield different approximate distributions.

## 14.3 Log loss minimization

Aka Maximum likelihood estimation (MLE).

### 14.3.1 Optimization problem, estimate

Likelihood function: $L : T \to [0, 1]$.
$L(t|\left\{x^{(i)}\right\}) = f_t(\left\{x_r^{(i)}\right\} | \left\{x_{\neg r}^{(i)}\right\}) = \prod f_t(x^{(i)}|x_{\neg r}^{(i)})$.
$\hat{t} = argmax_t L(t|x)$. This may be a biased estimator; but is always a sufficient statistic, as it is defined on $L(t|x)$. Often, an equivalent optimization problem: minimizing log-likelihood is used.

#### 14.3.1.1 Functional Invariance property

If you want to estimate $g(t)$, $g(\hat{t})$ is the MLE of g(t). From definition.

#### 14.3.1.2 Avg Log likelihood function

Take $l(t|\left\{x^{(i)}\right\}) = \ln(L(t|\left\{x^{(i)}\right\}))$, and do $\min_t -n^{-1} l(t|X)$. Useful as often $L$ and distribution of X are from exponential family.
Example: In case of $N(\mu, \sigma^2)$, $\ln f(\left\{x^{(i)}\right\}|\mu, \sigma^2) = -\frac{1}{2\sigma^2}\sum(x^{(i)}-\mu)^2 - \frac{N}{2}\ln(\frac{\sigma^2}{2\pi})$; by maximization, MLE is $\mu = \bar{x}, \sigma^2 = N^{-1}(x^{(i)}-\mu)^2$: because biased estimator is used, $N^{-1}(x^{(i)} - \mu)^2$ often underestimates.

### 14.3.2 Other perspectives

#### 14.3.2.1 As log loss risk minimization

The negative log-likelihood of a single sample-point, $-\log L(t|x)$, is also called the 'log loss' in the general decision theoretic framework. So, by doing maximum likelihood estimation, we are actually minimizing empirical log-loss.

#### 14.3.2.2 Priors as regularizers

If you add regularizer $r(t)$, you are imposing a prior distribution on $t$; so you are doing bayesian inference. The optimization problem becomes: $\min l(t|X)+r(t)$.

#### 14.3.2.3 As empirical code-length divergence minimization

Let $\mathcal{F}$be a class of distributions,
let $D$ be the actual distribution of $X$. In the limit where $n \to \infty$, maximum likelihood estimation tries to find $\arg\min_{F\in\mathcal{F}} E_D[-\log F(X)]$. This is the same problem as finding $\arg\min_{F\in\mathcal{F}} E_D[-\log F(X)] - E_D[-\log D(x)] = \arg\min_{F\in\mathcal{F}} KL(F||D)$. So we are finding a member of $\mathcal{F}$, with minimum code-length divergence to $D$.

### 14.3.3 Derivatives of log likelihood

#### 14.3.3.1 Score function : Sensitivity of log Likelihood

$V(t,X) = \nabla_t \log L(t|X) = L(t|X)^{-1}\nabla_t L(t|X)$: variability of $L(t|X)$ normalized by $L(t|X)$: like conditioning in numerical analysis.

**Mean wrt X** Under some regularity conditions,
$E_X[V(t,X)] = \int_x (f_{X|T}(x|t)^{-1}\nabla_t f_{X|T}(x|t)) f_{X|T}(x|t))dx =$
$\nabla_t \int_x f_{X|T}(x|t))dx = \nabla_t 1 = 0.$

#### 14.3.3.2 Variance wrt X of sensitivity score of likelihood

Aka Fisher Information matrix. As $E_X[V(t,X)] = 0$, $I(t) = E_X[V(t,X)^2] = E_X[(\nabla_t \log L(t|X))^2|t]$. Measures information about t in the observable RV X. If I(t) is high for RV X, then the absolute value of the sensitivity score is high in expectation.
If conditions like those in $E_X[V(t,X)] = 0$ hold,
$E_X[(f_{X|T}(x|t))^{-1}\nabla_t^2 f_{X|T}(x|t)] = 0$ will hold, and so
$I(t) = -E_X[(\nabla_t^2 \log L(t|X))^2|t]$.

### 14.3.4 Computational cost

If this optimization problem can be accurately and efficiently: great!

#### 14.3.4.1 Computing partition function

Suppose only $f(x,t)$ proportional to th epdf $f_t(x_r|x_{\neg r})$ is specified. Doing $\max_r \frac{f(x,t)}{\sum_y f(y,t)}$ is not the same as doing $\max_r f(x,t)$, as the normalizer (aka partition function) $Z(t) = \sum_y f(y,t)$, even though independent of $x$, varies with $t$.
If range of $x$ is huge, and Z(t) does not have a closed-form solution, computing $Z(t) = \sum_y f(y,t)$ can become costly and MLE is impractical.

**Ease in case of conditional probabilities** Suppose that $range(X_t)$ is actually small. Here, rather than having to sum over the entire range of $X$, which may be $range(x_r)^{|V|}$ in size, we just sum over $range(x_r)$ values to get $Z(t) = \sum_y f(y,t)$.

#### 14.3.4.2 Pseudolikelihood maximization

In case finding/ maximizing $f_t(x_r|x_{\neg r})$ is hard due to the need to compute $Z(t)$, but finding $f_{X_{r_j}|X_{\neg r_j}}$ is easy. So we can consider maximizing the pseudo-likelihood function $\prod_{r_j} f_{X_{r_j}|X_{\neg r_j}}$ instead.

## 14.4 Non uniform model for P(t)

Aka Maximum a posteriori (MAP).

### 14.4.1 Objective, estimate

Posterior probability of model considering observations $\propto$ likelihood of observations given the model $\times$ prior probability of model.
So, solve : $argmax_t f_{X|T}(x|t)f_T(t)$, or $argmax_t f_X(x)$ + regularizer.

### 14.4.2 Relation to MLE

MLE is a special case: it ignores prior distribution/ assumes it are uniform. So, often superior MLE. Eg: In MLE, upon seeing 4 H in coin tosses, you would conclude that the coin will always come up H.
The MLE optimization objective with a regularizer is a case of MAP, where the regularizer implitly defines a prior.

### 14.4.3 Defining prior distributions

Often, the prior distribution on $T$ may be specified, and the regularizer associated with the risk function may be derived thence.

#### 14.4.3.1 Hyperparameters for prior distribution of parameters

$f_{T|C}$ may have hyperparameters C, which may inturn come from distribution $f_{C|D}$ with hyperparameters $D$. Eventually must fix (hyper)parameter, perhaps with n-fold cross validation.
Eg: Can have hyper-parameters: $\sigma$ for label noise , $\alpha$ for prior over $\theta$.
These are akin to parameters, but of a special kind.

#### 14.4.3.2 Conjugate prior for a likelihood

You got distribution family $f_{X|T=t}$. Conjugacy: If $f_{X|T=t}$ and $f_T(t)$
have the same form, finding $Pr(t|X) = f_{(X_i)|t}((x_i))f_T(t) = \prod_i f_{X|T=t}(x_i)f_T(t)$ simpler. Then can update these probabilities with each incoming observation $X_i$ easily.
Eg: For k-categorical distribution: $Pr(X_j = x|p) = \prod_{i=1}^{k} p_i^{x_i}$, dirichlet distribution is the conjugate: $f_{P|A=a}(p) = \prod p_i^{a_i - 1}$ for hyperparameters a.

## 14.5 Model combination

Aka Fully Bayesian approach. We may first determine posterior distribution over parameters $f_{T|S}$ in the training stage, where $S$ is the training set. We may then arrive at the model $f_{L|X,S} = \int_{t \in T} f_{L|X,T=t} f_{T|S}(t)$.
So you use an ensemble of hypothesis distribution models.

## 14.6 Information criteria

Bayesian/ Schwartz information criterion (BIC): $\min 2^{-1}L(D,t)+p\log n$, where $p$ is the number of parameters needed to specify $t$, and $n$ is the number of samples.

### 14.6.1 Use

These are good if certain 'Gaussianness' assumptions hold, otherwise, the result of using them can be misleading. They allow us to pick $t$ without having to, for example, do cross-validation, which would have been necessary if $\min L(D,t)+lr(t)$ were used instead.

# 15 Support estimation

## 15.1 Estimate support of a distribution D

Find set $S'$ such that $Pr(x \notin S') < p \in (0,1]$, given sample $S$. Can be solved by probability density estimation techniques, but actually simpler.
Visualization: take the input space; draw solid ovals around sampled points; the algorithm will draw a dotted oval around these, which will represent the support of the distribution.

### 15.1.1 With soft margin kernel hyperplane

Aka One Class SVM or OSVM.
Given $N$ examples $\{x_i\}$; project to some feature space associated with kernel $k(x,y)=\phi(x)^T\phi(y)$; want to find hyperplane $w^T\phi(x)-\rho$ such that all points in the support fall on one side of the hyperplane, outliers fall on the other side: support identifier $f=sgn(w^Tx-\rho)$; so, allowing a soft margin, want to solve $\max_{\rho,w}\frac{\rho}{\|w\|}+C\sum\xi_i$ such that $w^T\phi(x_i)+\xi_i\geq\rho,\xi_i\geq 0$; $\equiv$ obj function: $\min_{w,\xi,\rho}\|w\|^2/2+\frac{1}{\nu N}\sum\xi_i-\rho$, for some coefficient $0\leq\nu\leq 1$.
Thence get Lagrangian:
$L(w,\xi,\rho,\alpha,\beta)=\|w\|^2/2+\frac{1}{\nu N}\sum\xi_i-\rho-\sum\alpha_i(w^T\phi(x_i)+\xi_i-\rho)-\sum\beta_i\xi_i$ with $\alpha,\beta\geq 0$.
Set derivatives wrt primal vars $w,\xi,\rho$ to 0 to get: $w=\sum_i\alpha_i\phi(x_i);\alpha_i=\frac{1}{\nu N}-\beta_i\leq\frac{1}{\nu N},\sum_i\alpha_i=1$. Thence, the support identifier becomes
$f=sgn(\sum_i\alpha_ik(x_i,x)-\rho)$; dual optimization problem becomes
$max_\alpha -2^{-1}\sum_{i,j}\alpha_i\alpha_jk(x_i,x_j)$ subject to $0\leq\alpha_i\leq(\nu N)^{-1},\sum_i\alpha_i=1$. Solving this, discover w; then recover $\rho$ using $\rho=w^T\phi(x_i)$ for $x_i$ with $\alpha_i\neq 0;\beta_i\neq 0$ (support vector with $\beta_i>0$): $\exists x_i$ as $\sum\alpha_i=1;\alpha_i\geq 0$.

#### 15.1.1.1 Choosing kernel, tuning parameters

$\nu\propto$ softness of the margin, number of support vectors, thence the runtime, sensitivity to appearence of novelty.

With Gaussian kernel, any data set is seperable as everything is mapped to same quadrant in feature space.
[**OP**]: How to decide width of Gaussian kernel to use? Can you use information about the abnormal class in choosing the kernel?

#### 15.1.1.2 Comparison with thresholded Kernel Density estimator

If $\nu = 1, \alpha_i = 1/N$, support identifier $f = sgn(\sum_i \alpha_i k(x_i, x) - \rho)$ same as one using a Kernel (Parzen) Density estimator. What happens when $\nu < 1$?

#### 15.1.1.3 Comparison with using soft margin hyperspheres

For homogenous kernels, $k(x, x)$ is a constant and the dual minimization problem
$\min_\alpha \sum_{i,j} \alpha_i \alpha_j k(x_i, x_j) - \sum_i \alpha_i k(x_i, x_i)$ and the support identifier
$f = sgn(R^2 - \sum_{i,j} \alpha_i \alpha_j k(x_i, x_j) + 2\sum_i k(x_i, x) - k(x, x))$ is equivalent to the minimization problem derived from the hyperplane formulation. So, all mapped patterns lie in a sphere in feature space; finding the smallest sphere containing them is equivalent to finding the segment of the sphere containing the data points, which reduces to finding the separating hyperplane.

#### 15.1.1.4 Connection to binary classification

Hyperplane $(w, \rho = 0)$
separates $\{(x_i, 1)\}$ from $(-x_i, -1)$ with margin $\rho / \|w\|$ and vice-versa.

### 15.1.2 Using soft margin hyperspheres

Aka Support vector data description. Here one solves: $\min_{R,\xi,c} R^2 + \frac{1}{\nu N} \sum_i \xi_i$ subject to $\|\phi(x_i - c)\|^2 - \xi_i \leq R^2, \xi_i \geq 0$.
After using the Lagrangian, finding the critical points and substituting, this leads to the dual $\min_\alpha \sum_{i,j} \alpha_i \alpha_j k(x_i, x_j) - \sum_i \alpha_i k(x_i, x_i)$ subject to $0 \leq \alpha_i \leq \frac{1}{\nu N}, \sum \alpha_i = 1$, and the solution $c = \sum \alpha_i \phi(x_i)$ corresponding to support identifier $f = sgn(R^2 - \sum_{i,j} \alpha_i \alpha_j k(x_i, x_j) + 2\sum_i k(x_i, x) - k(x, x))$ [**Check**].

### 15.1.3 Using Clustering

Cluster the sample, draw boundaries around the clusters. Eg: Use $k$ means clustering.

## 15.2 Novelty detection

### 15.2.1 Problem

Aka Outlier detection.
In general, we want to find outliers - unlikely data-points according to the conditional distributions $f_{X_r|X_{\neg r}}$.

#### 15.2.1.1 As One class classification

view as a problem where there are multiple classes, but all training examples are from one class only.

#### 15.2.1.2 Motivation

Outliers are detected either to focus attention on them or to remove them from consideration.

### 15.2.2 Using density estimation

Do density estimation; call apparently improbable data points novel.

### 15.2.3 Using support of the distribution

Find distrubution support, call anything outside the support an outlier.

#### 15.2.3.1 Ransack

One learns a model $M$ (either $f_{X_r|X_{\neg r}}$ or $E[f_{X_r|X_{\neg r}}]$) using data set $S$.
Then, one finds $S' \subset S$ for which $err(M; x) \geq t \forall x \in S'$.
$S'$ is then added to the set of outliers.
Finally, one repeats the entire procedure till the set of outliers is stable.

### 15.2.4 Boundary methods

#### 15.2.4.1 K nearest neighbors

Estimate local density of $x$ by taking avg distance to $k$ nearest neighbors; similarly estimate local density of each neighbor; call $x$ novel if its local density is much smaller than that of the neighbors.

#### 15.2.4.2 Support vector data description

[**Incomplete**]

#### 15.2.4.3 PCA

Simplify the data using PCA. [**Incomplete**]

## 16 Conditional independence structure: discrete case

## 16.1 Problems

In all of the following, we suppose that $f_X(x)$ is given by a graphical model.

### 16.1.1 Model Estimation

The most ambitious goal is to estimate the parameters associated with the distribution. This can often be accomplished by minimizing the log loss associated with the conditional distribution $f_{X_i|X_{\Gamma(i)}}$, once the structure of the underlying graphical model has been estimated (ie feature selection is done).

### 16.1.2 Edge recovery

Deduce the graph encoding the conditional independence relationships among features.

#### 16.1.2.1 Ising models: Signed edge recovery

Take the ising model. Deduce not just the edge, but also the sign of the correlation/factor for interaciton between the nodes. Eg: Maybe want to deduce relationship voting patterns of legislators.

### 16.1.3 High dimensional case

Often have few (f(d)) samples from high-dimensional (d-dim) data, so $f(d) << d$.

#### 16.1.3.1 Measuring performance

Redefine statistical consistency: let both $d$ and f(d) tend to $\infty$.

## 16.2 Approaches

### 16.2.1 Learn closest tree

(Chow, Liu).

#### 16.2.1.1 Aim

Here, the aim is to learn a tree structured graphical model which is closest to the underlying distribution in terms of KL divergence.

#### 16.2.1.2 Algorithm

Construct a complete graph, where each edge is labeled with the mutual information estimated from the given sample. Then, run the minimum spanning tree algorithm over this graph.

## 16.3 Learn neighborhoods

### 16.3.1 For ising models

Consider for each node the conditional probability distributions $f_{X_i|X_{\Gamma(i)}}$, suppose that in the underlying graphical model, each edge is associate with a parameter $\theta_{ij}$. Suppose that the log loss is $l(X,\theta) = \hat{E}_x[-log(f_{\theta_i}(x_i|x_{\Gamma(i)}))]$. We can use log-loss minimization with l1 regularization : $\min_{\theta_i} l(X,\theta_i) + \lambda \|\theta_i\|_1$ to learn the (sparse) neighborhood of each node (ravikumar et al). This $l_1$ regularized logistic regression problem, in the case of exponential family distributions (including binary distributions) is a convex problem and can be efficiently solved.

#### 16.3.1.1 Results

For signed edge recovery, in case the distribution satisifies certain special conditions, signed edge recovery and good parameter estimation are guaranteed. The l1 regularization ensures that $n = O(d \log p)$ samples suffice to get a good model-estimate, where $p$ is the number of nodes and $d$ is the max-degree of the underlying graphical model.
Otherwise, $O(p)$ samples would be required, which is not feasible for high-dimensional model-selection.

#### 16.3.1.2 Caveats

However, experiments suggest that picking the right choices of $\lambda$, and thresholding after optimization can be very tricky. The $\lambda$ suggested may be applicable only for models with large $p$.

#### 16.3.1.3 Analysis technique

To prove the guarantees, one considers the optimality criteria of the convex optimization problem mentioned earlier, which says $\nabla l(X,\theta_i) + w = 0$ for some subgradient vector $w$. Then, we can begin to construct a solution $\hat{\theta}_i, w$ satisfying this condition, based on knowledge of actual parameters $\theta_i^*$. One then shows that the remaining optimality conditions and the claimed properties are satisfied.

### 16.3.2 For discrete graphical models

In this case, each edge is associated with $(m-1)^2$ parameters $\theta_{ij}$, and each node is associated with $m-1$ parameters, where $m$ is the size of the state-space. The algorithm for the Ising model case can be extended to the case of discrete graphical models, except that we will need to use $l1/l2$ regularization: $\|\theta_{ij}\|_2$.

This is multi-class logistic regression with l1/l2 regularization, which is a convex program. The same caveats and advantages as earlier apply.

# 17 Hypothesis testing

## 17.1 Model selection given 2 models

Aka Confirmatory data analysis: Test hypotheses, as against Exploratory data analysis: Find hypotheses worth testing.
Which process is more likely to have generated the data? Which model is better at explaining the observations? Model selection, with only 2 models.

## 17.2 Hypotheses

### 17.2.1 Null hypothesis

$H_0 : t = t_0$ or $t \leq t_0$

### 17.2.2 Alternate hypothesis

$H_a$; can be 1 sided like $t > t_a$ or 2 sided: $t \neq t_0$ or $|t - t_0| \geq k$.

### 17.2.3 The decision

So, you decide if parameter $t \in T_1$ or if $t \in T_2$.

## 17.3 Experiment/ Test

Pick sample; find value of estimate test statistic $\hat{t}$; accept $H_a$/ reject $H_0$ if $|\hat{t} - t_0| > |t' - t_0|$; fail to reject $H_0$ otherwise. Critical value t' defines $H_0$ rejection region. Visualize as shaded area under $\hat{t}$ pdf curve. So, you always do hypothesis testing assuming $H_0$ is true.

### 17.3.1 Errors

#### 17.3.1.1 Type 1

Erroneously accept $H_a$: $\alpha = Pr(\hat{t} > t'|t = t_0)$. Say $\alpha(= 0.05?)$ level of significance.

#### 17.3.1.2 Type 2

Erroneously fail to reject $H_0$: $\beta = Pr(\hat{t} \leq t'|t = t_a)$.

### 17.3.2 Tradeoff

Trying to decrease type 1 error involves increasing t'; But that increases type 2 error rate. Visualize error zones with regions in 2 bell curves with means slightly apart.
To simultaneously decrease both, must increase sample size.
In case of $X \sim N(\mu, \sigma^2), t = \mu$, can write $\hat{t} > t'$ as $z = \frac{\hat{t}-\mu}{\sigma/\sqrt{n}} > \frac{t'-\mu}{\sigma/\sqrt{n}}; z \sim N(0,1)$. Given $\mu, \sigma, \alpha, \beta$, can solve for t', n using $N(0,1)$ table. For small sample size, can use t distribution.

### 17.3.3 p-value of the statistic

Given a sample, got $\hat{t}$, for what $\min t'$ we will reject $H_0$ based on it? The corresponding $\alpha$ is p-value.

### 17.3.4 Power of a test

Take $H_0 : t = t_0; H_a : t = t_a$, fix $t'$. $power(t) = 1 - \beta(t)$: ability to detect if $H_a$ is true. $power(t_0) = \alpha$ [**Check**]. So, power curve has a minimum at $t_0$ [**Check**].

## 17.4 Test design

Consider goodness of test with $\alpha, \beta, power(t)$.

### 17.4.1 Best test for given $\alpha$

(Neyman-Pearson): Testing $H_0 : t = t_0, H_a : t = t_a$. Likelihood ratio test: $L = \frac{L(t_0|\hat{t})}{L(t_a|\hat{t})} \leq ?h$, $Pr(L \geq h) = \alpha$. This is the most powerful test.

### 17.4.2 Difference in differences

Suppose that using experiment A, where one compares hypotheses H1 and N (for null hyp), it is determined that N cannot be dismissed. In experiment B, one compares H2 and N and observe that N can be dismissed. From this, one cannot conclude that, while comparing H2 and H1, H1 can be dismissed: it is possible that the difference in evidence supporting H1 and H2 is small.
One should instead conduct an experiment comparing H1 and H2 directly. This has been a common mistake in medical research as of 2011!

# Part IV

# Label prediction/ identification

## 18 Problems

Aka supervised learning. There are a variety of prediction problems depending on the combination of problem components described below.

## 18.1 Core problem

### 18.1.1 Input and response variables

A label/ target/ response/ dependent $L$ depends on some predictor/ input/ independent variable $X$ (a set of features/ covariates).

### 18.1.2 Range of X and L

Input space is $D$ dimensional.
$range(L)$ may be a subset of a vector space. It may be continuous or discrete.

### 18.1.3 Labeling rule sought

The agent/ decision rule produced by the learning algorithm must label $L$ some unlabeled test point(s) $X$, after some observations/ examples/ training points $S$. As for decision theory in general, such a labeling rule may be randomized or deterministic.

## 18.2 Action space

$range(L)$ constitutes the action space in a decision theoretic view of the problem. It is possible for the action space to be expanded to include stating indecision about a label.

## 18.3 Actual phenomenon

### 18.3.1 Randomized function

In general, the labeling process can be seen a randomized function $c : ran(X) \rightarrow$ the set of RV's over range(L).

### 18.3.2 Volatility in form

IN some problems, the randomized labeling process $c$ changes with previous labelings of the labeling process ( and therefore with observations $S$). Eg: Predicting the position of a plane 5 seconds in the future, given its positions in the past few seconds.

In many other problems, the labeling process $c$ remains independent of observations.

### 18.3.3 Deterministic Labeling function

For simple phenomena, a deterministic function $c$ suffices to relate $X$ and $L$, so that $L = c(X)$.
If $ran(L)$ is discrete, $c$ is a discrete function, aka discriminant function: $c : dom(X) \to \{C_k\}$.

#### 18.3.3.1 Features

The labeling function can often be expressed using a feature mapping function $\phi(X)$. The various dimensions of $ran(\phi)$ aka features of the input.

### 18.3.4 General noise model

#### 18.3.4.1 Using a randomized noise function

Usually, the following model can be used to described the phenomenon: $Y = f(X), L = g(Y)$, where $f$ is a deterministic labeling function, $g$ is a randomized function, called the noise function, which maps $ran(f)$ to a random variable $L$.
$g$ is usually considered to be symmetric around the expectation.

#### 18.3.4.2 Using a Noise variable

Dependence of $L$ on $X$ can be written in terms of a deterministic noise application function $h$ and a random noise variable $N$, $L = h(f(X), N)$, where $f$ is a deterministic labeling function.

### 18.3.5 Noise in case of vector labels

Let $L = h(f(X), N)$ describe the dependence of $L$ on $X$, as described above. If $ran(L)$ is part of a vector space, $h()$ can often be described arithmetically.

#### 18.3.5.1 Additive noise

An additive noise application model is common: $L = f(X) + N$, and $N$ is usually has a symmetric (usually normal) distribution centered around 0.

#### 18.3.5.2 Multiplicative noise

Multiplicative noise models of the form $h(Y) = NY$ are also interesting. In this case, $N$ is centered at 1.

## 18.4 Example/ training points

The general properties/ peculiarities of a sample (eg: correlatedness, completeness, online vs offline learning) in general is considered elsewhere; here we are concerned with peculiarities associated with samples in case of the prediction problem.

### 18.4.1 Labeled

Given $N$ example points $S = \{(X_i, L_i)\}$. When such labeled examples are provided as part of the problem, the problem is called supervised learning.

### 18.4.2 Unlabeled

It is possible that we are additionally given a set $U$ of unlabeled points. In such a case, the problem is called 'semi-supervised learning'. The reason maybe that sometimes, easy to get data points, but expensive to label them; or maybe labels are noisy.

### 18.4.3 Alternative labels

For some data points belonging to the input space $ran(W)$, labels $K_i$ may be provided. So, examples are pairs $(W_i, K_i)$. Let the underlying labeling functions be $f_K : ran(W) \to$ RV's with range $ran(K)$ and $f_L : ran(X) \to$ RV's with range $ran(L)$.
When the two input spaces and labeling functions are related, this additional data helps in predicting $L$. Aka transfer learning problem. Eg: Such examples may help deduce relevant features.
In case of cross domain learning, $ran(K) = ran(L)$, but possibly $ran(W) \neq ran(X)$. Eg: search query result relevance identification.
In case of cross category learning, $ran(K) \neq ran(L)$.
In some applications both the label range and the input space may be the same, but the classificaiton functions may still be different. Eg: Netflix movie ratings by various people.
Eg application: Robot learns to stand using new legs faster using lessons learned when learning to stand using old legs.

## 18.5 Distribution on test points

This is an essential factor in calculating the risk of labeling rules.
This 'test distribution' usually usually is close to the training data distribution.

### 18.5.1 Transduction vs induction

If the test points are known during training: transduction; eg: semi-supervised learning: the labeling problem is called transduction.

Otherwise, the labeling problem is called one of induction. This is a harder problem.

# 19 Risk and evaluation

It may be essential to model $Pr(L|X, S)$; this is called **Inference** in the context of probabilistic graphical models.

## 19.1 Loss functions: labeling single data points

Different measures of goodness/ error functions are appropriate for different scenarios.

### 19.1.1 Loss functions: vector labels

Loss functions in this case are often defined to penalize deviation from the actual label symmetrically.
See loss functions in regression section.

### 19.1.2 Loss functions for classification

Below we mainly consider the loss functions $l(\hat{y}, y)$ which are without regularization to account for prior belief. Loss functions used in regression can directly be applied to this case: Eg: like squared difference.
Good loss functions are realistic, maybe convex and smooth too - so that the corresponding empirical risk minimization problem (for picking classifiers) they lead to is tractable.

### 19.1.3 0/1 loss

$l() = I[\hat{y} \neq y]$. The corresponding risk will be $Pr(\hat{y} \neq y) = E[I[\hat{y} \neq y]]$, the misclassification rate.
The minimal risk classifier has the form: $h(X) = \arg\max_L Pr(L|X, S)$. The maximization procedure is called **Decoding** in the context of probabilistic graphical models.
However, the corresponding empirical risk minimization is non-convex, is NP hard and cannot is not approximable to a constant fraction of goodness. So, auxiliary loss functions are used.

#### 19.1.3.1 Minimal risk: Binary classification

Aka Bayes risk. Suppose that data is generated by the model specified by specifying the pdfs $f_{X|Y=1}$, $f_{X|Y=0}$, $Pr(y)$. Then, the best possible classifier is one which has accurate knowledge of the generative model, and even this classifier, in general, has a non zero risk. Its risk is given by $\sum_y Pr_{x:Pr(y|x)\geq 1/2}(\neg y)$, or by $E_x[\min_y Pr(y|x)]$.

#### 19.1.3.2 Connection to log loss risk: binary classification

Whenever $I[\hat{y} \neq y]$, we know that $Pr_t(y|x) \geq 1/2$. So, $E_{x,y}[-\log Pr_t(y|x)] \geq (-\log Pr_t(y|x))Pr(\hat{y} \neq y) \geq \log 2 Pr(\hat{y} \neq y)$. Hence, upper-bound on log loss risk is also an upper bound for $Pr(\hat{y} \neq y)$.

### 19.1.4 Log loss

Suppose that our predictor is based on the model $f_{Y|X,T=t}$. Then log loss is $-log f_{Y|X,T=t}(y|x)$. This punishes the model for assigning low probabilities to an observation. Minimizing log loss corresponds to maximum likelihood estimation. The corresponding risk is $E_{x,y}[-log f_{Y|X,T=t}(y|x)]$.

## 19.2 Loss functions: labeling multiple data points

### 19.2.1 Confusion matrix

For qualitative evaluation, make k*k confusion matrix $C$ with $C_{i,j}$ as number of points belonging to class i predicted as belonging to class j.

### 19.2.2 True and false positives

$U :=$ set of all points. $y(c) :=$ points belonging to class c. $\hat{y}(c) :=$ set of points predicted to belong to class c.
True positives: $tp(c, \hat{y}) := |y(c) \cap \hat{y}(c)|$. False positives: $fp(c, \hat{y}) := |\hat{y}(c) - y(c)|$. False negatives: $function(c, \hat{y}) = |y(c) \cap (U - \hat{y}(c))|$. True negatives: $tn(c, \hat{y}) = |(U - y(c)) \cap (U - \hat{y}(c))|$.

### 19.2.3 Precision, recall, specificity

Micro averaged precision : $P(\hat{y}) = \frac{\sum_c tp(c,\hat{y})}{\sum_c (fp(c,\hat{y}) + tp(c,\hat{y}))}$.
Recall, or completeness or sensitivity or true positive rate:
$R(\hat{y}) = \frac{\sum_c tp(c,\hat{y})}{\sum_c (function(c,\hat{y}) + tp(c,\hat{y}))}$. Measures ability to identify items belonging to class c.
Specificity: $S(c, \hat{y}) = \frac{tn(c,\hat{y})}{tn(c,\hat{y}) + fp(c,\hat{y})}$. Measures ability to discard items not belonging to class c. $1 - S(c, \hat{y})$ is false positive rate.
F-measure, the harmonic mean of precision and recall, is also used to evaluate success.

#### 19.2.3.1 Emphasis on one '+ve' class

If you only care about performance from the perspective of one class, as in the case of link prediction, let c range over only that class in the summations above.

#### 19.2.3.2 Sensitivity - specificity tradeoff

Ideally, want to increase both sensitivity and specificity. But to increase sensitivity, the classifier often needs to take more risks in classifying an entity as 'positive'. There will be many cases where -ve entities are declared +ve: there is decrease in specificity.
Visualize two normal curves over 1-D feature: one for the -ve case and one for the +ve case. On observing a feature, a classifier uses a cutoff to identify +ve cases. Compare with tradeoff between type-1 and type-2 errors in hypothesis testing.

#### 19.2.3.3 Sensitivity vs 1-specificity curve

Aka Receiver operating characteristic (ROC) curve. Take a parametrized family of predictors/ tests to identify +ve cases; the predictors are distinguished by the cutoff they choose in making classifications using the same scores for items. ROC considers the sensitivity vs specificy tradeoffs of various tests belonging to this family.
So, for all tests, you plot sensitivy and 1- specificity on a graph, and join these points by a straight line : a piecewise linear function starting at 0, and ending at 1; not a step function.

**Area under curve (AUC)** Ranges of sensitivity and 1-specificity are [0, 1]. All curves startat 0, where every item is classified -ve, and end at 1, where every item is classified +ve.
A good test (family) is as close as possible to the left axis: parameters can be tuned to increase sensitivity without sacrificing specificity too much. An ideal test has area under the curve (AUC) 1. The higher the AUC, the better the test family. A random predictor has AUC 0.5.
So, AUC measures the test's discrimination, or the ability to separate +ve cases from the -ve cases.

#### 19.2.3.4 Precision/ recall tradeoff

Recall monotonically increases with number of points classified as +ve. Precision monotonically decreases with the number of false positives. Often precision and recall have inverse relationship: if you classify all points as +ve, you have very high completeness but bad recall.
Often want to see how these change with classification parameters: so draw

plots.

# 20 General Solution properties

## 20.1 Empirical risk minimization vs expert systems

One way to predict $L$ given $X$ is to learn the process by which $L$ is generated: one can manually create a labeling algorithm by listing out rules which influence labeling: for example by studying expert human animals.
Often, you cannot know enough about how stuff works in order to have explicit rules. Eg: Expert system for handwriting detection can be tougher to make than a training a statistical model. And, repeatedly, in various fields, statistics based automatic learning of rules has outperformed manually developed expert systems. Eg: Natural language processing.
Here, we mainly consider statistical learning.

## 20.2 Hypothesis classes

General discussion about hypothesis classes described in the Decision Theory chapters applies.
Suppose that the labeling rule, on observing $X$, produces the labeling rule $Y$.

### 20.2.1 Probabilistic models

As the hypothesis class, one may choose a family of probabilistic models for $f_{Y|X}$ specified either directly as in the case of discriminative probabilistic modeling, or indirectly by specifying a generative model: that is, modeling $f_{X,Y}$ (which may involve modeling $f_{X|Y}, f_Y$).
Model/ parameter selection for such hypothesis classes is described in the distribution structure learning part.

### 20.2.2 Mean or Mode models

Alternately, rather than modeling the pdf $f_{Y|X}$, one may choose a hypothesis class composed of labeling functions of the form $f(X; W) \to ran(Y)$ where $W$ is the parameter variable.
This may actually correspond to proposing deterministic labeling functions which return either label mode or the mean (in case of vector labels). In the latter case, we may also be interested in specifying or modeling $var_Y[Y|X = x]$ to indicate our confidence in the estimate.

#### 20.2.2.1 Comparison to probabilistic models

Depending on the loss functions, the risk of different randomized labeling rules with the same expectation may be different. So, when it is reliably possible

to do so, modeling the pdf $f_{Y|X}$ is probably a better as one can then evaluate confidence, risk etc.. involved in making a classification decision.
Yet, task of modeling the expectation or mode of $Y$, being relatively simpler, lends itself to better modeling and learning.

### 20.2.3 Probabilistic models: comparison

Often, modeling $f_{Y|X}$, requiring fewer modeling assumptions, may be more easily and accurately done.
But, modeling $f_{X,Y} = f_{X|Y}f_Y$ gives us more expressiveness. For example, in case of unbalanced data it may be important to explicitly model $f_Y$; and it can be estimated easily using the empirical distribution based on $\{(X_i, L_i)\}$.

#### 20.2.3.1 Discriminative model corresponding to generative model

Consider the form of the discriminative model $f_{Y|X,W}$ yielded by the generative model $f_{X|Y}f_Y$, which is parametrized by $W$. When derived using the generative model, the range of $W$ (and therefore $f_{Y|X,W}$) is restricted by the form of $f_{X|Y}f_Y$. So, the hypothesis class $D$, where $W$ has no restrictions, is atleast as large as the hypothesis class constituted by labeling rules corresponding to the generative mode. So, the pros and cons of hypothesis classes of different sizes, as discussed in the decision theory chapter, apply.

#### 20.2.3.2 Ease in using unlabeled points

Suppose that the distribution family $Pr(y, x|w)$ is parameterized by $w$. Then, in picking the best $w$, one often maximizes the likelihood of observing data-points $\left\{(x^{(i)}, y^{(i)})\right\}$.
If we have unlabeled observations $\left\{x^{(j)}\right\}$, as in semi-supervised learning to make predictions: you can then select $w$ which account for these unlabeled points along with labeled points!

## 20.3 Discrete deterministic labeling rules

### 20.3.1 Decision surfaces

One can view discrete labeling rules $h$ as dividing $ran(X)$ into decision regions separated by decision boundaries/ decision surfaces, which correspond to all $x$ such that $Pr(Y = y_1|X = x, S) = Pr(Y = y_2|X = x, S)$ where $S$ is the training data.

### 20.3.2 k-ary classifier from binary classifier

k-ary classification reducible to binary classification in many ways.
Can learn many 'one against rest' classifiers; and then assign the class corresponding to the deepest distance the point achieves from the decision hyperplane.

Can learn many one against one classifiers; and then use majority vote for prediction. This approach often results in ambiguous regions.

### 20.3.3 Curse of dimensionality

[**Incomplete**]Exponential increase in volume associated with adding extra dimensions: can't calculate and record $f_{X|C_k}(x)$ or $Pr(C_k|x)$ for exponential number of cases as you don't have so many examples. More difficult to answer 'How are data points belonging to same class similar?'
See also curse of dimensionality subsection in the clustering problem.

## 21 With additional unlabeled data-points

Assume points belonging to the same class cluster together, using unlabeled data with labeled data, you can draw better decision boundaries around clusters belonging to various class: probability density is important in classification.

## 21.1 Generative approaches

[**Incomplete**]

## 21.2 Label propagation on graphs

Make the similarity graph G=(V, E) of the n points, $\{X_i\}$. Some nodes $X_T$ have labels, others don't have labels; let $k$ be the indicator vector with $k_i = [i \in T]$; take K = diag(k).

### 21.2.1 Quadratic criterion

#### 21.2.1.1 Binary labels

Denote proposed labels of the n points as bits in a binary vector f. Let y be the label vector, with $y_T = c(X_T)$ and $y_{V-T}$ having arbitrary values.
$\min_{f \in \{0,1\}^n} \sum_{(i,j) \in E} W_{i,j}(f_i - f_j)^2 + l(f-y)^T K(f-y)$. This finds an f which is smooth in that $f_i$ and $f_j$ are similar if they are connected by an edge in G, but also which agrees with $y_T$ as much as possible.

#### 21.2.1.2 Discrete labels

Denote the $L$ possible labellings thus: $\{e_i\}$: as L-dim indicator vectors. Denote labels of the n points as columns in the binary matrix F, where $\forall i : F_{:,i} \in \{e_j\}$. Let Y be the label matrix of similar dimensions, with columns $Y_{:,T} = c(X_T)$.
$\min_{F_{:,i} \in \{0,1\}^n} \sum_{(i,j) \in E} W_{i,j}(F_{:,i} - F_{:,j})^2 + l \sum_i (F_{:,i} - Y_{:,i})^T K(F_{:,i} - Y_{:,i})$. This is akin to solving $L$ binary label propogation problems simultaneously: propogating 1 label bit at a time. ***Note that the binary case is a special case of this.***

***Combinatorial hardness in both cases!***

### 21.2.2 Rewriting using Graph Laplacians

The objectives are equivalent to $\min f^T L f + l(f-y)^T K(f-y)$ and $\min tr(F^T LF) + ltr((F-Y)^T K(F-Y))$. $\sum_{(i,j)\in E} W_{i,j}(f_i - f_j)^2 = f^T Lf$ is shown in graph theory ref; and the latter equivalence is easily obtained by using this as a component.
Using the normalized graph laplacian $N$, get the objective $\min tr(F^T NF) + ltr((F-Y)^T K(F-Y))$. In graph theory ref, see that $f^T Nf$ is a normalized smoothness measure.

### 21.2.3 Real relaxation: solve linear system of eqns

In both the binary and the general discrete case, drop the constraints $f \in \{0,1\}^n, F_{:,i} \in \{0,1\}^n$. Allow any real value; solve this relaxed problem to get f or F, and get to the closest binary f' and F'.
From optimality conditions, the solution is $(L+K)F = KY$.

### 21.2.4 Low rank approximation for fast solution

(L+K) is $n \times n$, and solving this system of equations is $O(n^3)$ naively. Instead, can use low rank factorization $TT^*$ where $T \in R^{n\times k}$ to solve this in $O(nk^2)$. But, finding low rank approximation requires finding sv (same as finding ev for symmetric matrices), which is $O(kn^2)$. Can surpass this using eigenfunctions. [**Find proof**]

## 22 Using labels from other viewpoints

## 22.1 Data point Neighborhood approach

Make data pts vs data pts matrix which measures similarities; filled using labels from various, usually similar, viewpoints. Thence predict label from target viewpoint. Eg: Amazon: 'others who bought this also bought that'.

## 22.2 Collaborative filtering

### 22.2.1 Latent factor approach

Assume small number of latent random variables, which combine together to form various view points and data points, cause the labels.

#### 22.2.1.1 Low rank factorization

Let values of latent RV behind data pt j be $v_j$, behind viewpoint i be $u_i$; then get $Y_{i,j} = u_i^T v_j$. Want to have few random variables, so want $Y = U^T V; U \in R^{q \times N}; V \in R^{q \times D}$, a low rank factorization of Y. Thence fill missing values.
Can use SVD: finds best rank $k$ solution to $\|Y - (U\Sigma)V^*\|_F$. instead want to find $\left\|W \otimes (Y - U^T V)\right\|$, where W is mask matrix to indicate known values in Y.

### 22.2.2 Association rule mining

Look at co-occurances of various items. Eg: Walmart purchases.
Very slow.

## 23 Vector labels prediction: Regression

## 23.1 General problem

Many (x, y) pairs (observations), h(x, w) form (eg: degree of polynomial) known, parameters $w$ in h(x, w) is unknown. Want to find $w$. range(h) is not finite, it is usually continuous.
y is the response variable, $w$ is called the regression vector. The matrix formed by $x$ is often called the design matrix.
Many such continuous valued models are described in probabilistic models reference.

## 23.2 Linear regression

### 23.2.1 The problem

h(x, w) is linear in $w$ (Eg: $w_1 x^3 + w_2$x+$w_3 = 0$: $\phi_i(x) = x^i$).
Make matrix A with each row as feature vector $\phi(x)$ at data point $x$; take b at diff points; coefficients as variable vector $w$.

### 23.2.2 The solution

Want to tune $w$ so that it yields least deviation from b. You measure this using various loss functions.

#### 23.2.2.1 Quadratic loss function

Get least squares problem $\min e(w) = \min \|Aw - b\|_2$. This is symmetric, but is sensitive to outliers.

### 23.2.3 Maximum likelihood estimate with Gaussian noise

If you view y as h(x, w) + gaussian noise n, least squares solution is also the maximum likelihood solution.
Noise distribution symmetric about the mean is not sufficient to lead to least squares solution: $\min e(w) = \min \|Aw - b\|_4$ symmetrically penalizes deviation from mean just as well.

### 23.2.4 Imposing prior distributions on w

Solutions below assume quadratic loss function to measure deviation from b. Priors implied by regularizers in $\min e(w) = \min \|Aw - b\|_2 + p(w)$ where p is some penalty function. Usually $p(w) = \|w\|_k$.

#### 23.2.4.1 Quadratic regularizer

Assuming gaussian noise, the maximum a-posteriori solution yields the ridge regression problem.

#### 23.2.4.2 Priors which prefer sparse w

Can use lasso, or compressed sensing. See optimization ref.

**Statistical efficiency** N samples, $d$ dimensions. $E[\left\|\hat{\theta} - \theta^*\right\|_2] \leq \sqrt{\frac{s \log d}{N}}$.

### 23.2.5 Solution

See optimization ref.

## 24 Prediction with fully labeled data

Goals and formulations are presented elsewhere.

## 24.1 Binary classification

See the many hypothesis classes/ parametrized model families in the colt ref, boolean functions ref.

## 24.2 Non parametric methods

### 24.2.1 k nearest neighbors

A discriminative, non parametric approach. Number of examples: N. Samples: $S = (x_1, ..x_N)$. There are $k$ classes. To classify $x$, find $k$ nearest neighbors in S; take their majority vote.
So, can't ever throw away data points.

## 24.3 Linear models for discrete classification

Linear separability of data sets or feature space: Decision surfaces are (p-1) dim hyperplanes in feature space.
$h(x, w) = f(w^T x + w_0) = f(v^T x')$ with $x' = (x, 1)$: $f$ is activation function or link function; $w$ is the weight vector; $w_0$ is the bias. So without loss of generality, we can restrict ourselves to considering only hyperplanes passing through the origin.
Decision surfaces are h(x, v) = constant or $v^T x'$ = constant, so linear in $x$; Not linear in terms of $w$ due to possible non linearity of $f$: so called generalized linear model.
For binary classification, this becomes a halfspace: see boolean function ref and colt ref.
For geometric properties of separating hyperplane, see boolean function ref.

### 24.3.1 Arbitrary separator from fully separable training set

When the training set is fully separable, one can pick one of the many separating hyperplane easily, for example, using linear programming. For other such algorithms, see computational learning theory ref.
To select the best among the candidate hyperplanes, one can use some sort of regularization, as in maximum margin classifiers.

### 24.3.2 Winnow: multiplicative update

Let weight of $c$ be $W$. Set weights $a_i = 1$. Multiplicative update rule: if $x_i$ agrees with c(x), set $a_i = a_i(1 + \delta)$; set $a_i = a_i/(1 + \delta)$ for others. $mb = O(W^2 \log n)$. [**Find proof**]Sometimes better than additive update rule used in the perceptron algorithm.
This is very similar to the 'panel of experts' algorithm. Also see note in the section on perceptron algorithm comparing this with the 'panel of experts' algorithm.

### 24.3.3 Perceptron learning alg for halfspaces

#### 24.3.3.1 The problem

The classifier: A hyperplane c through origin perfectly classifies labeled data; unit vector $u \perp c$ defines c; $c(x_i) = sgn(\langle u, x_i \rangle)$.
The data: $x \in R^n$; $\|x_i\| = 1$; $S = \{x_i\}$. Geometric margin of X wrt u: $g = \min_{x \in S} |\langle u, x \rangle|$; or $sgn(\langle u, x_i \rangle)\langle u, x_i \rangle \geq g$. Note: g = function(S).
Want to find $c$.

#### 24.3.3.2 The algorithm

$u_0 = 0$. Additive update rule: If mistake on $x_i$: $u_{i+1} = u_i + sgn(\langle u, x_i \rangle)x_i$: hyperplane orientation tilted towards correcting the mistake.

#### 24.3.3.3 Convergence to u

$mb = O(g^{-2})$. In general, if $\|x_i\| \leq R$; $mb = O((\frac{R}{g})^2)$.
By induction, using update rule expression for $u_t$: $\|u_t\| = \|u_t\| \|u\| \geq \langle u, u_t \rangle \geq tg$. So, if the length of $u_t$ is not increasing too much, may be perceptron is getting closer to u as more mistakes made.
Also, by induction, the update rule and the fact that $u_{t-1}$ misclassified $x_{t-1}$, causing the update: $\|u_t\|^2 \leq t$.
So, $(tg)^2 \leq t$; and $t \leq g^{-2}$.

#### 24.3.3.4 Comparison

Perceptron Algorithm is usually faster than than LP. Is exponential when $g \leq 2^{-c}$: this is rare.
For a given $g$, we can find good enough halfspace with mb $O((\frac{R+D}{g})^2)$. [**Check**]Perhaps the winnow algorithm, which uses multiplicative update is more efficient.
Has connection to halfspace learning with noise. [**Find proof**]
Assumes that the data is perfectly separable. So, often less preferable than soft margin SVM's. But, a soft margin variant of perceptron algorithm is known.

**With panel of experts algorithm** Compare the perceptron algorithm $B$ with the algorithm $A$ described in the learning theory survey, which for any given sequence of inputs, using a panel of experts achieves a mistake bound comparable with the mistake bound achieved by the best expert. In the case of halfspaces, every input bit $x_i$ can be viewed as an 'expert'.
Upon making a mistake, $A$ updates the weight of only the experts which made a mistake, whereas $B$ updates weight assigned to every expert.
The weights used in $A$ were all positive, whereas weights used in $B$ can be negative: but this distinction is minor, as it can perhaps be accounted for in the 'experts algorithm' by introducing experts corresponding to $-x_i$.

## 24.4 Maximum margin classifier

### 24.4.1 The problem

A discriminative, parametric approach. Number of examples: N. Samples: $(x_1, ..x_N)$, label function $c : X \to \{\pm 1\}$.
Suppose $y(x) = w^T \phi(x) + w_0$ with $y(x)c(x) > 0 \forall x$, for some $w, w_0, \phi$. So finding a separating hyperplane (see halfspaces in boolean function ref) in some feature space.

### 24.4.2 Hard margin

#### 24.4.2.1 Primal

To maximize margin, solve: $\max_{w,w_0}[\frac{\min_n[y(x_n)c(x_n)]}{\|w\|}]$. Scale w, $w_0$ so that $\min_n[y(x_n)c(x_n)] = 1$; thence get $\equiv$ problem $\min_{w,w_0} \frac{\|w\|^2}{2}$: $y(x_n)c(x_n) \geq 1$. Prediction: sgn(y(x)).
Can solve using Quadratic programming (QP).

#### 24.4.2.2 Dual

Get Lagrangian $L(w, w_0, a) = \frac{\|w\|^2}{2} + \sum a_n[1 - (w^T\phi(x_n) + w_0)c(x_n)]$; $a_n \geq 0$. Get dual: $g(a) = \inf_{w,w_0} L(w, w_0, a)$; Set $\nabla_{w,w_0} L(w, w_0, a) = 0$: $w = \sum_n a_n c(x_n)\phi(x_n); 0 = \sum a_n c(x_n)$. So, dual problem: $\max_a g(a) = \max \sum a_n - 2^{-1}\sum_n \sum_m a_n a_m c(x_n)c(x_m)k(x_n, x_m)$: $a_n \geq 0; \sum a_n c(x_n) = 0$.
Can solve using QP too. This form is useful where $dim(\phi(x)) >> N$.

**KKT conditions** Primal feasible: $y(x_n)c(x_n) \geq 1$. Dual feasible: $a_n \geq 0; \sum a_n c(x_n) = 0$. Complementary slackness: $a_n[1 - c(x_n)y(x_n)] = 0$.
So, $\forall n : a_n = 0 \vee c(x_n)y(x_n) = 1$: in latter case, you have support vectors. So, aka Support Vector Machine (SVM). Take S: number of support vectors.

**Predictor** Substituting for w, get: $y(x) = \sum_n a_n c(x_n)k(x_n, x) + w_0$: only S terms actually appear with $a_n \neq 0$: so SVM is fast to evaluate. So, $w_0 = \frac{\sum_m [c(x_m)y(x_m) - \sum_n a_n k(x_n, x_m)]}{N}$.

### 24.4.3 Soft margins

Allow some points to be misclassified or to be below the margin, but linearly penalize such outliers.

#### 24.4.3.1 Primal

So, use slack variables: $\xi_n \geq 0$; Instead of $y(x_n)c(x_n) \geq 1$, use constraint $y(x_n)c(x_n) + \xi_n \geq 1$.
$\min C\sum_{n=1}^N \xi_n + \frac{\|w\|^2}{2}$: $C$ is tradeoff between penalty on $\xi$ and margin; saying $\sum \xi_i \leq G$; so controls model complexity. As $C \to \infty$, get hard margin SVM.

#### 24.4.3.2 Dual

Lagrangian: $L(w, w_0, \xi, a, m) = \frac{\|w\|^2}{2} + C\sum_{n=1}^N \xi_n + \sum a_n(1 - c(x_n)y(x_n) - \xi_n) + \sum \mu_n \xi_n$: $a \geq 0, \mu \geq 0$. Set $\nabla_{w,w_0,\xi}$L= 0: $w = \sum a_n c(x_n)\phi(x_n), \sum a_n c(x_n) = 0, a_n$ =C$-\mu_n$. Thence get dual g(a), with objective function same as hard-margin case with constraints: $0 \leq a_n \leq C$: as $\mu_n \geq 0; \sum a_n c(x_n) = 0$.

**KKT conditions** Primal feasible: $1 - c(x_n)y(x_n) - \xi_n \leq 0$. Dual feasible: $a_n \geq 0, \mu_n \geq 0$. Complimentary slackness: $a_n(1-c(x_n)y(x_n)-\xi_n) = 0, \mu_n\xi_n = 0$.
So, support vectors now are points on or within certain margin from hyperplane. Predictor same as hard margin case.

# 25 Sparse signal detection

## 25.1 Problem

### 25.1.1 Generating process

Suppose that the $p$ dimensional 'observation vector' $Y$ is generated from $Y_i \sim N(\theta_i, \sigma^2) = \theta_i + N(0, \sigma^2)$. $\sigma$ is known.
In addition suppose that $\theta$ is sparse. The set $K = \{i : \theta_i \neq 0\}$ is called the signal set.
$n$ observations $\{Y_i = y_i\}$ are made. Usually $n << p$.

### 25.1.2 Decision rule sought

The problem for the decision rule, given $Y_i$ (the observation), is to estimate $\theta_i$. More simply, one might seek decision rules to estimate the indicator variable $I[\theta_i \neq 0]$ given $\theta_i$.

### 25.1.3 As a classification problem

This is essentially a classification problem with some peculiarities. It is an abduction problem (the test points are known beforehand), and no labeled training set is provided.
Framing it as a classification problem is a good way to state the final goal, but one can not apply solution ideas typical of classifiers naturally. So, this view is not very informative.

### 25.1.4 Peculiarities

If the number of signals, $|K|$ were known beforehand, the problem would be trivial: one would just select the top $|K|$ elements of $\hat{E}[Y]$.

## 25.2 Risk

Identifying non-signals as signals often carries an especially high penalty. Eg: In case of gene-expression data, in response to certain conditions, the expression (ie, signalness) of each gene identified as being a signal is verified using laborious wet-lab experiments.
So, it is often hard to express a formula for evaluating the actual risk of a decision procedure, yet one can make qualitative statements about it. Yet, one

can define a simpler risk function and show that a decision procedure chosen using a certain process will be low risk. [**Incomplete**]

## 25.3 Hypothesis classes

### 25.3.1 Desired qualities

#### 25.3.1.1 Sparsity

The main point in modeling $\theta_i$ is to ensure that the model results in sparse $\theta$: that is $\theta_i$ should often be close to 0.

#### 25.3.1.2 Adaptability to different sparsity levels

The hypothesis class should include $\theta$ of different sparsity levels.

#### 25.3.1.3 Robustness to large signals

The hypothesis class should include $\theta$ with arbitrarily large components.

### 25.3.2 Probabilistic models

[**Incomplete**]

#### 25.3.2.1 Scale mixture models

Scale mixture models for $\theta_i$ say: $\theta_i|\lambda_i = N(0, \lambda_i^2)$.

# 26 Sequential data points

## 26.1 Trajectory prediction

### 26.1.1 Problem

Consider sequential data: observations $S$ consisting of labels $L_1..L_n$ observed at different positions $X_1..X_n$ (perhaps times). We want to predict $L_{n+1}$ corresponding to $X_{n+1}$.
The underlying process is such that the distribution of $L_{n+1}$ depends on both $S$ and $X_{n+1}$.

### 26.1.2 Simplifications

Even predicting a lower bound or upper bound (whp) of $L_{n+1}$ may be useful.

### 26.1.3 Applications

Predicting trajectory of a missile, market price of a security tomorrow.

# Part V

# Applications

Also see AI survey.

## 27 Search results

## 27.1 Ranking search query results

Queries $q \in Q$: bag of words, Set of documents D: a seq of words. For query $q_i$, retrieved relevant documents $D_i \subseteq D$; Got full or partial orderings $L_i$. Let $Q = \{q_i\}$. Maybe want to complete Q vs D relevence level matrix. Or, given docs D' relevant for unseen query q, want to rank D'.

### 27.1.1 Feature extraction

$\phi : Q \times D \to X \in R^D$; D is usually $\approx 10$. $\phi_i(q, d)$ could be TF/IDF, common word count etc.. Let $f : X \to R$ be scoring function; it induces a relevance order $\{z_i\}$ on D for every query $q_i$; let $F = \{f\}$.

### 27.1.2 Objectives to optimize

Let $y_i$ be ground truth relevance-ordering of D corresponding to query $q_i$; let y be induced by $g : X \to R$.
Loss function: $L(f, q_i)$ measures deviation of $z_i$ from $y_i$.
Risk: $R = \int_Q L(f, q) f_Q(q) dq$. Want to find f which minimizes L. But Pr(q) usually unknown; so minimize empirical risk: $\hat{R} = n^{-1} \sum_i L(f, q_i)$. Assume that $\left| R - \hat{R} \right| \to 0$.

### 27.1.3 Various Loss functions L

#### 27.1.3.1 Pointwise

[**Incomplete**]

#### 27.1.3.2 Pairwise

Checks if this is violated: $y_i(d) > y_i(d') \implies f(d) > f(d')$; so $(y_i(d) - y_i(d'))(f(d) - f(d')) \geq 0$. Widely used. Eg: SVM ordinal regression; ranknet (best pairwise function as of 2009); lambda-rank: $f(d) = w^T d$.

#### 27.1.3.3 Listwise

Measure badness of a bigger part of the ranking, not just correctness of ranking of pairs of documents. $q_i$ uniquely identified by $(y_i, D)$; so can consider $L(f, y_i, D)$ or $L(z_i, y_i)$.

**Using a probability distribution on permutations** Define prob distribution over permutations $Pr_f(\pi)$ with good properties: favors permutations rated highly according to f to permutations rated lower, best permutation according to f has max probability. So find f whose $Pr_f$ is closest to $Pr_g$. Can minimize cross entropy: $L(f, g) = -\sum_\pi Pr_g(\pi) \log Pr_f(\pi)$ (List net); or $L(f, g) = -\sum \log Pr(y_i|D, f)$: thereby maximizing likelihood (List-MLE). These Listwise loss calculation is hard: may need to consider n! permutations: so instead consider only ranking of top $k$ docs induced by f. List-MLE known to be best Listwise loss function based ranking method (2009), also better than all pairsise loss function based methods.

# 28 Document clustering

## 28.1 Document classification and clustering

Aka text mining. Corpus of documents. Usually position of document in corpus ignored in modeling, just like bag of words assumption.

### 28.1.1 Feature extraction

Document is a string of words; but want a simpler representation.
Maybe just the set of words which appear is most important: use word-counts.
Maybe number of occurances is also important.

**Burstiness** First time a word appears, it is very informative; but its second occurance is much less surprising/ informative. Rarer the word, the more this is true.

#### 28.1.1.1 Bag of words assumption

Aka word exchangability. Each document modeled as just a bunch of words. Ignore stopwords, words with low I(word; document).

### 28.1.2 Dimensionality reduction

Results in noise reduction: good way of tackling synonyms. Considers the entire corpus.

#### 28.1.2.1 Approaches

Can look at the feature vs item matrix, and try to find a low rank approximation for it. This yields the latent factors behind features and items.
Or, can describe a parametrized process which creates a vector in the feature space (an item), and find the parameters which generated each item. This yields the latent factors behind items.

### 28.1.3 Model class distribution using word counts

Can take word counts, then use multinomial distribution to model class distr. But this ignores burstiness; so it is appropriate only for common words.

**DCM distr** Better model to utilize avg rare, very rare words. Draw document specific multinomial distribution.

### 28.1.4 Classification

Documents naturally belong to many classes.

### 28.1.5 Clustering

Coclustering useful. Clustering words also useful in query enhancement.
Coclustering : Can view as a bipartite graph clustering problem (Dhillon).

## 29 Web portal related

## 29.1 Pick content

Match content with user intent.
Long term objective: Maximize user experience. Short term, meterable, objective: Maximize click-through rates (CTR) subject to constraints (Eg: don't show porn).

## 29.2 Pick content-layout

## 29.3 Maximize ad revenue

### 29.3.1 Match advertisements with content

[**Incomplete**]

## 29.4 Data

Inventory often expires: Eg: news articles; cause: user fatigue.
Data is huge.

### 29.4.1 User responses

Responses are multivariate: click rates, ratings etc.. Negative responses very noisy.

## 29.5 Online experimentation

Comparing models tough without experiments.
Usually allowed on a fraction of visits. A generalization of multi-armed bandits: See Game theory ref.

# 30 Spoken dialog system (domain-specific)

## 30.1 Problem

The problem is for the user to communicate a task to a machine through a verbal dialog. A dialog consists of several 'turns', with the user and the machine speaking alternately within each turn.
Since the machine's 'belief' (aka dialog state) about the user's task changes as the dialog proceeds, this problem is also called belief-tracking.

### 30.1.1 Examples

Voice mail system. Bus route information system. Route system.

### 30.1.2 Domain ontology

The set of concepts the interaction involves is often limited. For example, in the voice-mail system, the concepts may be limited to Name, Message, Action (having values 'record message', 'save message', 'delete message'). This naturally limits the language internally used by the machine, though to interact with the user a richer language may be used.

## 30.2 Interaction graph

The chain of events involved in dialog-processing is as follows. First the user has a goal $S_u$ he wants to communicate. Also, the dialog itself has a current state $S_d$, based on what has already been uttered. $S_u, S_d$ cannot be observed by the machine.
The user then speaks something - ie acts on an intention $A_u$. This is transformed into speech/ percieved action $Y_u$.
This is processed by the Automatic Speech Recognition (ASR) module, which provides a scored list of possible phrases the user has uttered.
The output of ASR is passed to a (usually domain specific) parser or natural language understanding (NLU) module. This outputs a scored list of utter-

ences: $\{(A'_u, c)\}$ in a certain well-defined language. $A'_u$ is usually in a language specific to the domain, eg: (Bus = 40, start = StopA, end = StopB).
The output of the NLU is then passed to the belief tracking system, which then produces a scored list of beliefs, or a belief state $S_m$ which is meant to approximate $(S_u, S_d, A_u)$.
A decision-making module (aka decision manager) considers $S_m$ and chooses what action $A_m$ to perform (including what to say to the user). $A_m$ in-turn affects the dialog model/ belief state.
$A_m$ is then used by the language generation modules to communicate with the user. Thence, speech/ action $Y_m$ is produced, which is then decoded as representing $A'_m$ by the user.
The above can be visualized as a directed graph with various components being represented by nodes and messages labelling arrows.
The ASR and NLU modules use Finite State Transducers for their functioning.

# 31 Others

## 31.1 Predicting movie ratings

Netflix problem: rating(movie, user, time) for some triples available; make prediction for unknown triples.