

Algorithms For Distances Between Distributions

Suresh Venkatasubramanian

AT&T Labs – Research

EPFL, Jul 15, 2005

Outline

1 Introduction

2 Three Applications

- The Information Bottleneck
- Nonparametric Change Detection
- Scan Statistics

3 Algorithms

- Representation And Approximation
- Fast Distance Estimation
- Small-space Distance Estimation

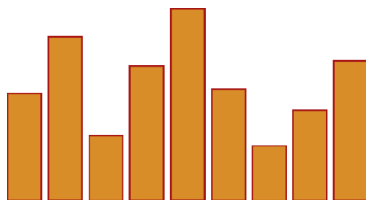
Distributions and Histograms

A family of distributions $p(\mathbf{x}; \theta)$ parametrized by θ

- The distribution could be described by a smooth density function:

$$p(\mathbf{x}; (A, \mu)) = C \exp((\mathbf{x} - \mu)^\top A (\mathbf{x} - \mu))$$

- Or it could be described by a set of buckets and counts:



$$p = p(\mathbf{x}, \{p_1, p_2, \dots, p_k\})$$

Distances between distributions

For two distributions p, q , the distance $d(p, q)$

- satisfies **reflexivity**: $d(p, p) = 0$
also will satisfy (in most cases) **isolation**: $d(p, q) = 0 \Rightarrow p = q$.
- Will occasionally be **symmetric** $d(p, q) = d(q, p)$.
- Will rarely satisfy the **triangle inequality**:

$$d(p, q) + d(q, r) \geq d(p, r)$$

but might satisfy a relaxed version:

$$d(p, q) + d(q, r) \geq c \cdot d(p, r), c < 1$$

A Rogues' Gallery

Kullback-Leibler Distance

$$\text{KL}(p, q) = \sum_i p_i \log \frac{p_i}{q_i}$$

The Jensen-Shannon Distance

$$\text{JS}_{\alpha, \beta}(p, q) = \alpha \text{KL}(p, m) + \beta \text{KL}(q, m), m = \alpha p + \beta q, \alpha + \beta = 1$$

χ^2 -Distance

$$\chi^2(p, q) = \sum_i \frac{(p_i - q_i)^2}{q_i}$$

Δ -Distance

$$\Delta(p, q) = \sum_i \frac{(p_i - q_i)^2}{p_i + q_i}$$

Hellinger-Matusita-Bhattacharya Distance

$$d_H(p, q) = \left[\sum_i (\sqrt{p_i} - \sqrt{q_i})^2 \right]^{\frac{1}{2}}$$

The Rogues' Club

f-divergence For convex $f : \mathbb{R} \rightarrow \mathbb{R}$, $f(1) = 0$,

$$D_f(p, q) = \sum_i p_i f\left(\frac{q_i}{p_i}\right)$$

Bregman divergence For convex $\phi : \mathbb{R}^d \rightarrow \mathbb{R}$

$$D_\phi(\mathbf{p}, \mathbf{q}) = \phi(\mathbf{p}) - \phi(\mathbf{q}) - \langle \nabla \phi(\mathbf{q}), \mathbf{p} - \mathbf{q} \rangle$$

Renyi Divergence For $\alpha > 0, \alpha \neq 1$,

$$D_\alpha(p, q) = \frac{1}{\alpha - 1} \log \sum_i p_i^\alpha q_i^{1-\alpha}$$

All of the above classes can be derived from axiomatic considerations.

Main Focus of this talk

Computing distances between distributions is important in many settings. A rigorous examination of issues of efficiency and approximation yield intriguing and nontrivial problems.

Outline

1 Introduction

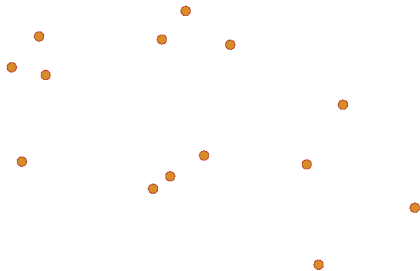
2 Three Applications

- The Information Bottleneck
- Nonparametric Change Detection
- Scan Statistics

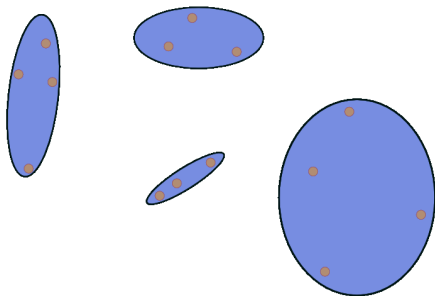
3 Algorithms

- Representation And Approximation
- Fast Distance Estimation
- Small-space Distance Estimation

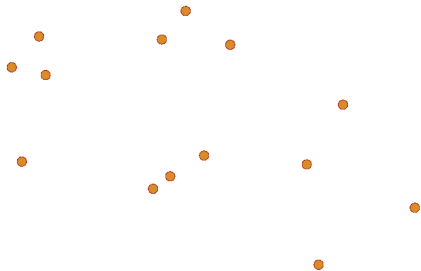
Soft Clustering



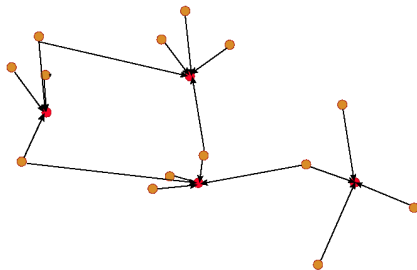
Soft Clustering



Soft Clustering



Soft Clustering



In soft clustering, a point may have multiple cluster memberships, but the “weight” of all memberships must sum to one.

The Information Bottleneck Method [TPB]

X is a collection of “points” defined as vectors over a set Y . In practice, we are given an array of the joint distribution $p(x, y)$.

X	Y
Documents	Words
Hi-D points	Dimensions
Strings	q -grams

A *cluster* is a distribution over Y , described by

Variable	Meaning	Equivalent in “hard” clustering
$p(t x)$	Cluster memberships for $x \in X$	Subsets of X
$p(t)$	Total mass of cluster	Size of cluster
$p(y t)$	Description of cluster center	Coordinates of cluster center

Defining Cluster Quality

Two goals:

- Minimize cluster “description length”
- Maximize quality (or minimize error)

Idea: Use **Mutual Information**

$$\begin{aligned} I(X; Y) &= \sum_x \sum_y p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \\ &= H(X) - H(X|Y) = H(Y) - H(Y|X) \end{aligned}$$

$I(X; T)$ small $\Rightarrow T$ **compresses** X very well.

$I(T; Y)$ large $\Rightarrow T$ is a good representation of Y .

Think of the rate distortion theorem

The IB functional

Definition (Information Bottleneck)

Given X , Y , minimize the functional

$$F = I(X; T) - \beta I(T; Y)$$

Solution:

$$p(t|x) \propto p(t)e^{-\beta \text{KL}(p(y|x), p(y|t))}$$

$$p(y|t) = \frac{1}{p(t)} \sum_x p(y|x)p(t|x)p(x)$$

$$p(t) = \sum_x p(x)p(t|x)$$

Note:

- Everything in one cluster $\Rightarrow I(X; T) = 0$
- $T = X \Rightarrow I(T; Y)$ is maximized.

Hard-Clustering Limit [DMK]

If we fix k clusters, and let $\beta \rightarrow \infty$, we get probabilistic k -means: Cluster a collection of distributions into k clusters such that

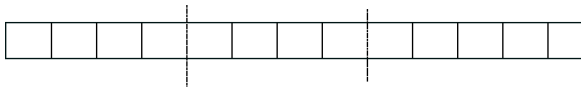
$$\sum_i p(C_i) \sum_{p \in C_i} \text{KL}(p, c_i)$$

is minimized (c_i is center of C_i).

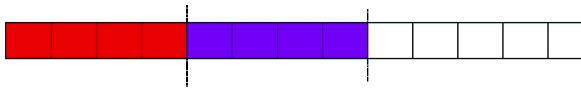
This is analogous to ℓ_2^2 -clustering

Thus, clustering of distributions is similar to ℓ_2 -based clustering, with $\text{KL}(p, q)$ playing the role of ℓ_2^2 (this is not accidental).

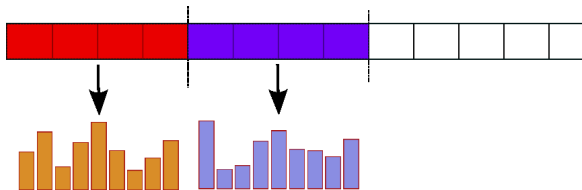
“Nonparametric” Change Detection



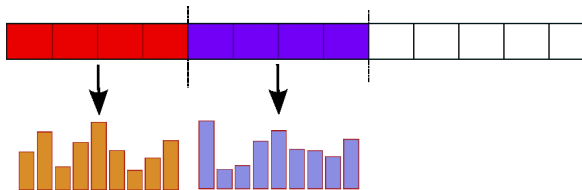
“Nonparametric” Change Detection



“Nonparametric” Change Detection



“Nonparametric” Change Detection



- Checking for signals in neuronal pathways
- Anomaly detection in network monitoring
- “Data cleaning”: Have billing records changed (unexpectedly) from yesterday to today ?

A General Approach

- Could try fitting a model
 - Determine parameters for model from both windows, and compute the difference.
 - How do we know what family to use ?
- More general: Build a distribution out of data, and compare distributions using a general distance.

But which one ?

Using the KL distance

The KL distance generalizes known statistical tests of difference

- If p, q are Gaussians with same variance, then $\text{KL}(p, q) = (\mu_p - \mu_q)^2$, which gives us the *t-test*
- The KL distance behaves (to first order) like a χ^2 -function.

The KL distance relates to optimal hypothesis testing:

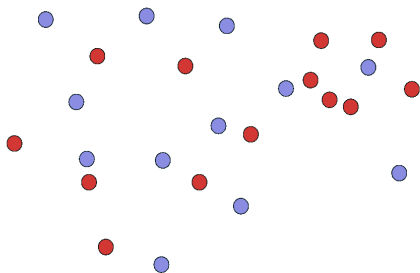
Theorem (Neyman-Pearson, Stein's Lemma)

If null hypothesis is "no change occurred", then measuring KL distance in this way is the asymptotically optimal test in terms of minimizing Type I/II error. Moreover, the KL distance is the exponent of the misclassification error.

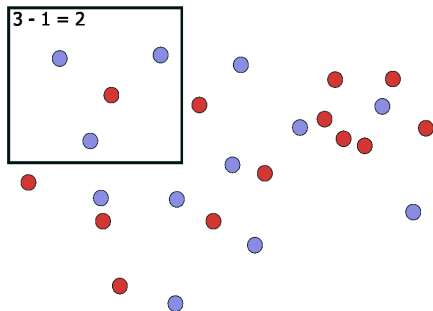
Procedure [DKVY05,KMV05]

- 1 Fix window size W
 - 2 **FOR** Two most recent windows
 - 3 Construct distribution from each
 - 4 Compute KL distance between them
 - 5 Do *bootstrapping* for statistical significance (“p-value”) testing.
 - 6 If significant, record change.
 - 7 **END FOR**
- Distributions are constructed by making a spatial partition using a quad-tree like structure.
 - Bootstrapping requires re-sampling of window data.
 - Overall running time is proportional to window size and depth of spatial partitioning structure for each update.

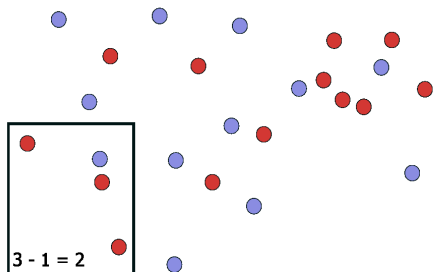
Discrepancy



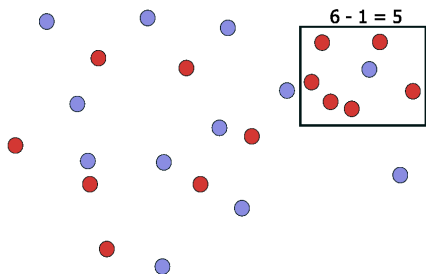
Discrepancy



Discrepancy



Discrepancy



Likelihood-based measures

Assume that data is drawn from a one-parameter exponential family.

Let μ_1 be maximum likelihood parameter value inside region; let μ_2 be maximum likelihood parameter outside.

Null Hypothesis: $\mathcal{H}_0: \mu_1 = \mu_2$

We can compute likelihood ratio $\Pr(\frac{\overline{\mathcal{H}_0}}{\mathcal{H}_0})$

This yields statistical discrepancy function of the form

$$\text{KL}(\{r, 1 - r\}, \{b, 1 - b\}) \text{ (Kulldorff scan statistic)}$$

or

$$\chi^2(\{r, 1 - r\}, \{b, 1 - b\})$$

or other forms (where r is fraction of red points in region, and b is fraction of blue points in region).

Problem (Statistical Discrepancy)

Maximize statistical discrepancy over space of shapes (rectangles, circles, cylinders, ...).

From Convex To Linear [APV05]

Likelihood-based discrepancy functions are often convex in their parameters. Thus, we need to find a shape maximizing a convex function.

To avoid uninteresting solution, one generally assumes *support condition*: each region must contain at least some fixed number of points to be considered interesting.

Any convex function can be replaced by a suitably small set of linear functions that approximate it to any desired error.

We can compute the maximum discrepancy rectangle to within error ϵ in time $O(\frac{n^2 \log^2 n}{\epsilon})$. Best prior result was $O(n^4)$ (exact/trivial).

Key idea: Showing that the eigenvalues of the Hessian of the KL distance have bounded growth rate as a function of n , the number of data points..

Outline

1 Introduction

2 Three Applications

- The Information Bottleneck
- Nonparametric Change Detection
- Scan Statistics

3 Algorithms

- Representation And Approximation
- Fast Distance Estimation
- Small-space Distance Estimation

Representation and Approximation

Problem

Can we compute distributional distances efficiently ?

This can be done easily in time $O(d)$, if the vectors are d -dimensional. But what if d is large ?

Problem

Can we reduce dimensionality while preserving distances approximately ?

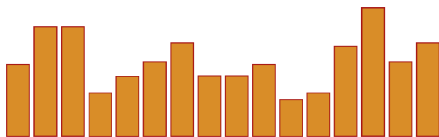
This is a big can of worms ! Lots of work on combining distributions, classification, independence, *etc.*

Histograms: A Simple Example

Extensive study of histograms in database/algorithms community. Histograms are used to maintain approximate profiles of a database so that query optimization can be performed effectively.

Problem (Histogram construction)

Given a function defined on n data points and an error function, find a piece-wise constant function with B pieces that minimizes error to the original function.

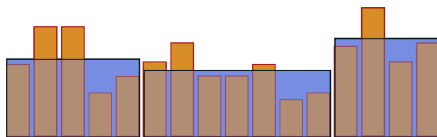


Histograms: A Simple Example

Extensive study of histograms in database/algorithms community. Histograms are used to maintain approximate profiles of a database so that query optimization can be performed effectively.

Problem (Histogram construction)

Given a function defined on n data points and an error function, find a piece-wise constant function with B pieces that minimizes error to the original function.



Maintaining small size histograms

Two step process:

- 1 Fix a partition of the interval $[1 \dots n]$ into B sub-intervals.
- 2 For a *fixed* partition, determine the best “representative” for each interval.

Dynamic programming allows us to enumerate over all partitions efficiently. But how do we compute the representative ?

Maintaining small size histograms

Two step process:

- 1 Fix a partition of the interval $[1 \dots n]$ into B sub-intervals.
- 2 For a *fixed* partition, determine the best “representative” for each interval.

Dynamic programming allows us to enumerate over all partitions efficiently. But how do we compute the representative ?

Lemma

For any distance defined by a separable Bregman function, the optimal representatives can be determined [GV05].

- Bregman distances are assymmetric in general: are we minimizing $D(p, q)$ with varying p or q ?
- For ℓ_2 , the arithmetic mean gives the best representative (this is well known).
- For the KL distance, using geometric mean of the values in an interval as the representative is optimal.
- Not the end of the story: we don't know how bad the optimal B -bucket histogram is, and more sophisticated techniques that work for ℓ_2 should be examined.

Property Testing

Problem

How many samples of p and q do I need to make to estimate the distance between them accurately?

Formally, we distinguish whether distance is at most ϵ^2/\sqrt{n} or at least ϵ .

Some examples:

- ℓ_1 : $O(n^{2/3}\epsilon^{-4})$ [TFRSW]
- ℓ_2 : $O(\epsilon^{-4})$ [GR]

Theorem (GMV05)

For the Jensen-Shannon/Hellinger/Triangle distances, $O(n^{2/3}\text{poly}(1/\epsilon))$ samples are necessary and sufficient

Main idea: If individual p_i values are large, then this provides a good estimate. If not, then we “remove” large values, and use ℓ_2 to approximate the distance on the low probability regions.

Estimation over Streams

Problem

How much space do I need to accurately measure (within a $(1 + \epsilon)$ factor) the distance between distributions p and q , when they are presented as a stream of samples from the distribution?

In a stream problem, data arrives as items x_1, x_2, \dots, x_n , and the space used must be $o(n)$. When streaming *distributions*, the distribution is defined by frequencies of data items, and the data is ordered adversarially.

Thus, the data stream will look like $(+, a_1), (-, a_2), (-, a_3), \dots$, where $(+, a)$ denotes a sample of value a from p , and $(-, b)$ denotes a sample of value b from q .

Again, some examples:

- l_1, l_2 : $O(\text{polylog}n)$ space suffices

We don't know how to obtain a stream-based approximation algorithm with the above memory bounds.... But...

Estimating Entropy

Theorem (GMV05)

In polylogarithmic space we can approximate the entropy of a data stream to within a factor of $\frac{e}{e-1} + \epsilon$.

In space $O(\frac{n^\alpha \log^2 n}{H})$, we can estimate entropy H upto a factor of $\frac{1}{\alpha}(1 + \epsilon')$.

$$JS(p, q) = H(p) + H(q) - 2H((p + q)/2)$$

Why are these results interesting?

Sampling is the most natural way of interacting with a distribution. Our algorithms reveal how sampling (a small-time operation) interacts with streaming (a small space operation) for processing distributions. Specifically,

Theorem

Any algorithm that via sampling (or generalized sampling) approximates some symmetric function of a distribution can be simulated by a 2-pass streaming algorithm.

*Moreover, if we consider a stream model where data arrives in random order rather than adversarial, then the above algorithm can be simulated in a **single pass**.*

Summary

- Computing distances between distributions is an important primitive, and many applications require the ability to compute these distances efficiently and scalably.
- We currently have a few tools for analyzing and approximating such distances; need more general techniques (possibly from primal-dual theory and differential geometry)
- Estimating distances between distributions on streams, and computing small space representations of distributions are fundamental problems to tackle.

Acknowledgements

- Deepak Agarwal
- Tamraparni Dasu
- Shankar Krishnan
- Sudipto Guha
- Sampath Kannan
- Andrew McGregor
- Jeff Phillips
- Kevin Yi

Thanks!

Suresh Venkatasubramanian (suresh@research.att.com)
<http://www.research.att.com/~suresh>