Parametrizing the easiness of machine learning problems

Sanjoy Dasgupta, UC San Diego

Outline

- Linear separators
- Mixture models
- Nonparametric clustering
- Nonparametric classification and regression
- Nearest neighbor search

Learning linear separators

Finding the best linear separator for a data set is NP-hard.
But this problem is a basic primitive that can be solved easily.

1950s: "***margin***" is a measure of the easiness of a linear separation task.

1. Separable data (1950s)
Default: linear programming.
With margin: the three-line perceptron algorithm.

2. Non-separable data (1990s)
Notion of soft margin.
Optimize the soft margin: SVM.

Generalization bounds depend on margin.

Progress in algorithms for linear classification and in statistical analysis of such procedures has been driven by the notion of margin.

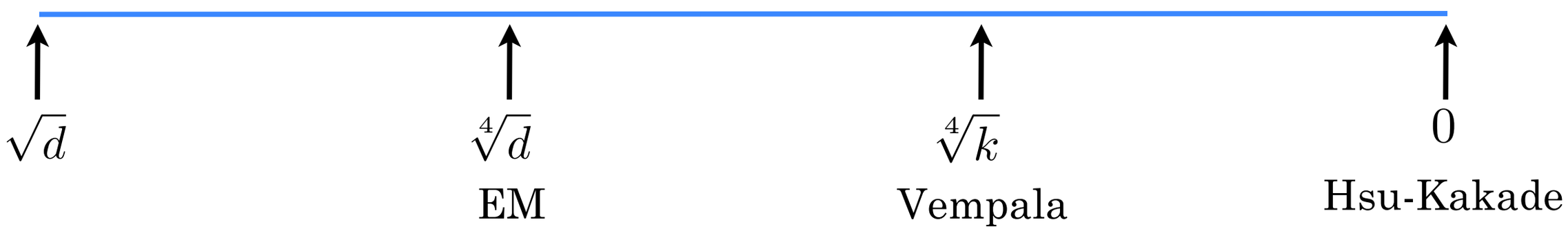
Learning mixture models

Given data from an unknown mixture of k Gaussians in R^d, determine the parameters of that mixture model.
Studied since 1894.

Statistical literature fixated upon a local search procedure: EM.

Recent progress has been driven by a measure of easiness: the ***separation***.
How many standard deviations apart are the centers of the Gaussians?

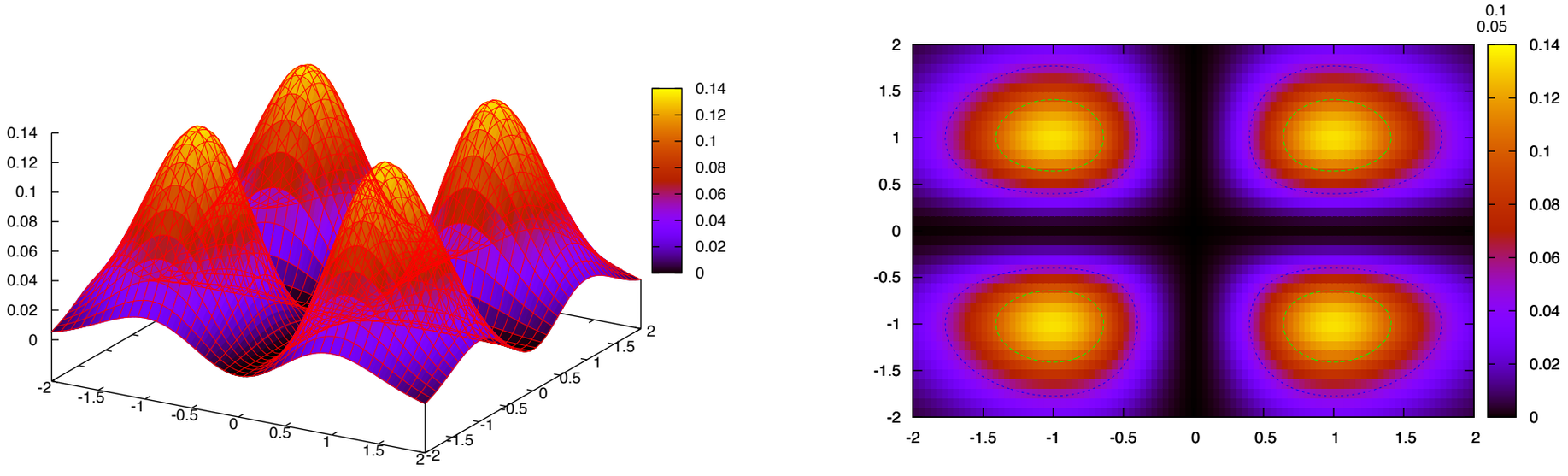
Algorithms that assume a certain amount of separation:



Nonparametric clustering

Data points $X_1, \ldots, X_n$ are independent random draws from an unknown density f on $\mathbb{R}^d$

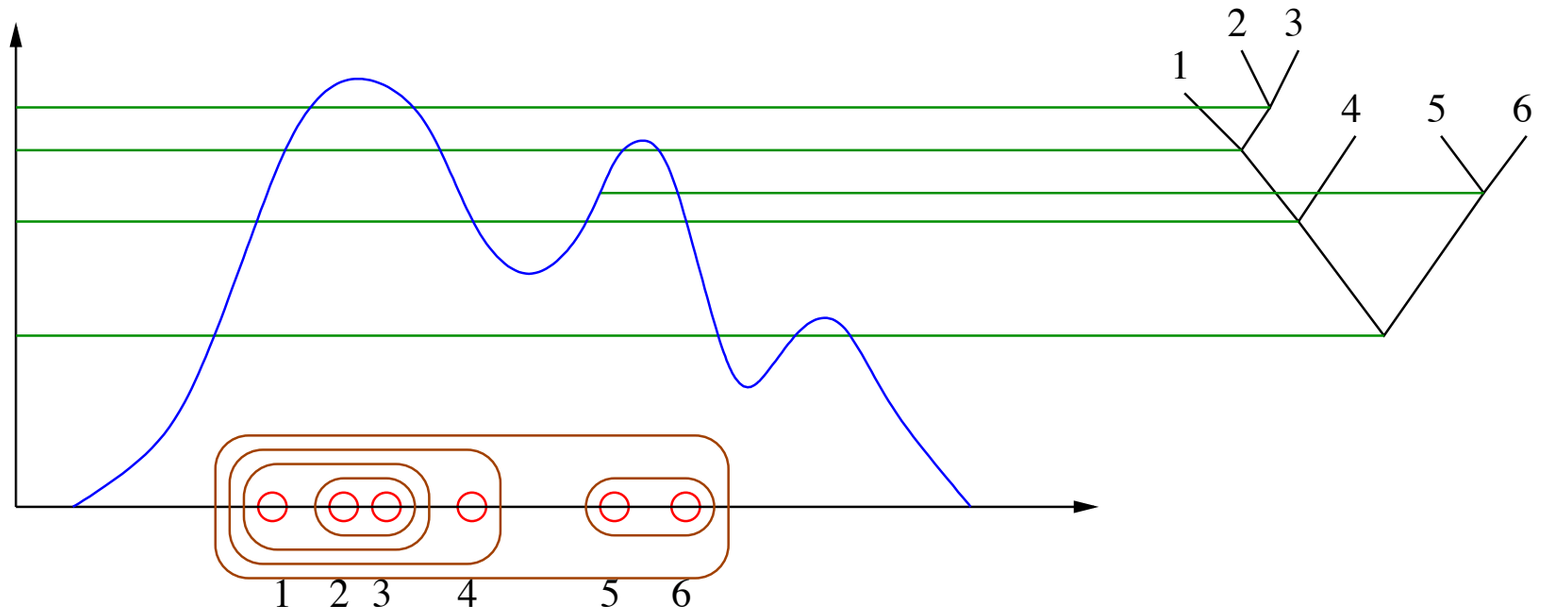
- Different random sample $\Rightarrow$ similar clustering (if n is large)
- As $n \to \infty$: approach "natural clusters" of f



cluster $\equiv$ connected component of $\{x : f(x) \geq \lambda\}$, any $\lambda > 0$

These clusters form an infinite hierarchy, the *cluster tree*.

Converging to the cluster tree



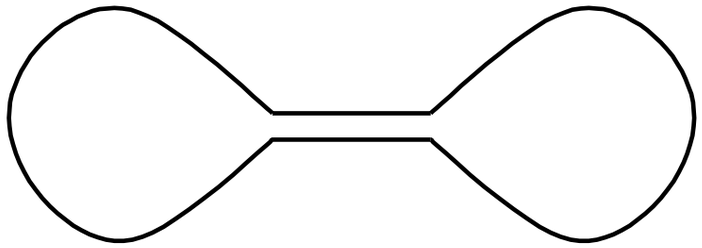
Consistency: Let A, A' be connected components of $\{f \geq \lambda\}$, for any λ. In the tree constructed from n data points X_n, let A_n be the smallest cluster containing $A \cap X_n$; likewise A'_n. Then:

$$\lim_{n \to \infty} \text{Prob}[A_n \text{ is disjoint from } A'_n] = 1$$

Hartigan (1975): Single linkage is consistent for d = 1.
Hartigan (1982): Single linkage is not consistent for d > 1.
Chaudhuri-D (2009): A slight variant of single linkage is consistent for all d.

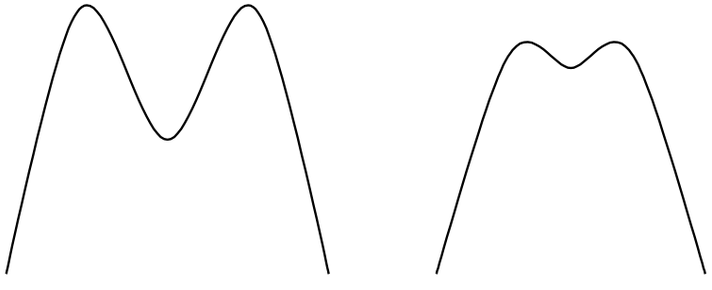
Cluster salience

Effect 1: thin bridges

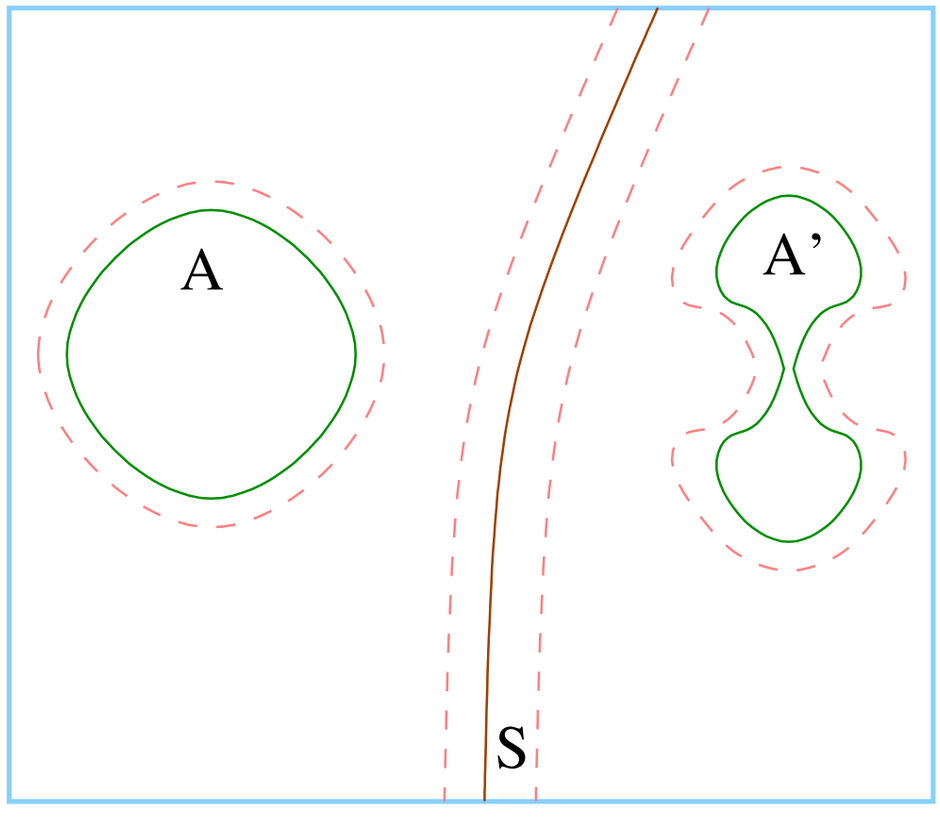


For any set Z, let Z_σ be all points within distance σ of it.

Effect 2: density dip

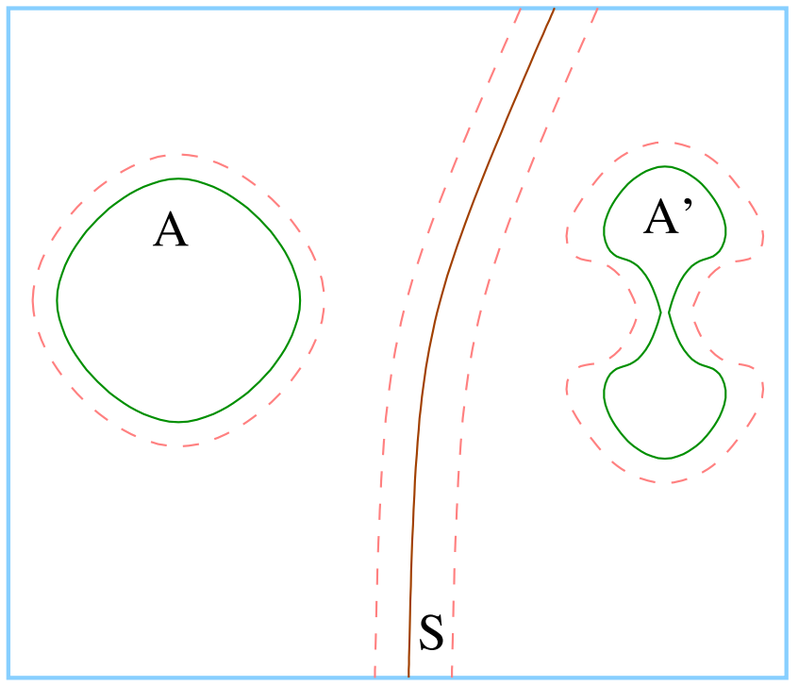


A and A' are (σ, ϵ)-separated if:
- separated by some set S
- max density in $S_\sigma \leq$
$(1-\epsilon)$(min density in A_σ, A'_σ)



Rate of convergence

A and A' are (σ, ϵ)-separated if:
- separated by some set S
- max density in $S_\sigma \leq$
$(1-\epsilon)$(min density in A_σ, A'_σ)



With high probability, for all connected sets A, A': if they have minimum density λ and are (σ, ϵ)-separated, then they will lie in separate subtrees when the number of samples satisfies

$$n \geq \frac{d}{\lambda \epsilon^2 \sigma^d}.$$

"Matching" lower bound: there is a distribution whose clusters require at least this many samples to distinguish.

Open problem:
more general notion of cluster salience, and algorithms that are adapted to it.

Nonparametric regression

The traditional bane of nonparametric statistics is the curse of dimension.
Stone (1977): for data in R^D, convergence rate $1/n^{1/D}$.

Three sources of recent rejuvenation:

1. Manifold learning
   Geometry/physics of the underlying domain constrains data to lie near a low-dimensional manifold.

2. Sparse data/models
   For instance, text data.

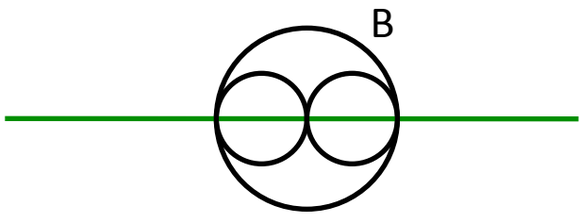
3. Bayesian nonparametrics
   Models that are exploit prior knowledge and smoothness to give meaningful predictions even with limited training data.

Is there a good unifying framework for 1,2, and other similar situations?

Doubling dimension

Set $S \subset R^D$ has doubling dimension d if for any Euclidean ball B, the subset $S \cap B$ can be covered by 2^d balls of half the radius.

S = line
Doubling dimension d = 1



S = k-flat
Doubling dimension d = O(k)

S = set of N points
Doubling dimension d = O(log N)

S = set of k-sparse points in R^D
Doubling dimension d = O(k log D)

S = d-dimensional submanifold of R^D with finite condition number
Doubling dimension d = O(k) in small enough neighborhoods

Rates of convergence for nonparametrics

A variety of methods that automatically adapt to the doubling dimension d of a data set in R^D (much of it due to Kpotufe):

Tree-based regression and classification:

$$n^{-1/D} \implies n^{-1/(d \log d)}$$

Kernel regression and classification:

$$n^{-1/D} \implies n^{-1/d}$$

K-nn regression and classification:

$$n^{-1/D} \implies n^{-1/d}$$

Open problem: more general notion of intrinsic dimension that captures "degrees of freedom".

Nearest-neighbor search

Given n data points X:
    Build a data structure on X.

Given subsequent query q:
    Use the data structure to find NN(q,X) efficiently.

Goals:
    Data structure size O(n), or maybe a bit more
    Query time o(n), hopefully much less

Three fairly common choices:
1. In metric spaces: hierarchical cover of some sort
2. In Euclidean space:
        K-d tree (or variants thereof) with defeatist search
        Locality-sensitive hashing

Often-quoted conjecture: any scheme for NN search requires either the query time or the size of the data structure to be exponential in the dimension.

Approximate NN search

For some c > 1, enough to return a point in X that is at most c times as far away as the true NN.

Locality-sensitive hashing (Indyk, Andoni) for Euclidean data:

Data structure size n^{1+1/c^2}
Query time n^{1/c^2}
Constant probability of success

Some problems:

- This is efficient only for fairly large c.
- The constant c has very different implications for different data sets.

Low intrinsic dimension

Can we do exact NN search in time exponential in the *intrinsic* dimension rather than the apparent dimension?

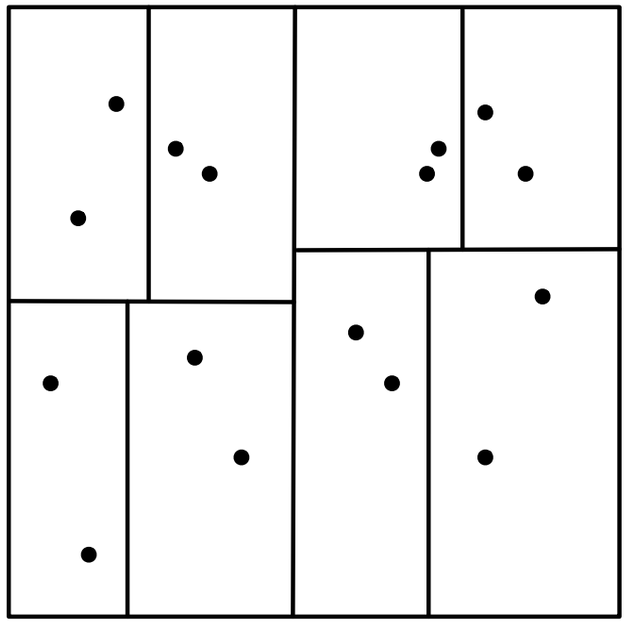
Distribution P (over a metric space) is a *doubling measure* of dimension d if for all points x and radii r,

$$P(B(x, 2r)) \leq 2^d P(B(x, r))$$

Many methods have query time $2^{O(d)}$ polylog(n) in this setting.
For instance, *cover tree* (Beygelzimer, Kakade, Langford)

Problem: this is too strong a notion of intrinsic dimension.

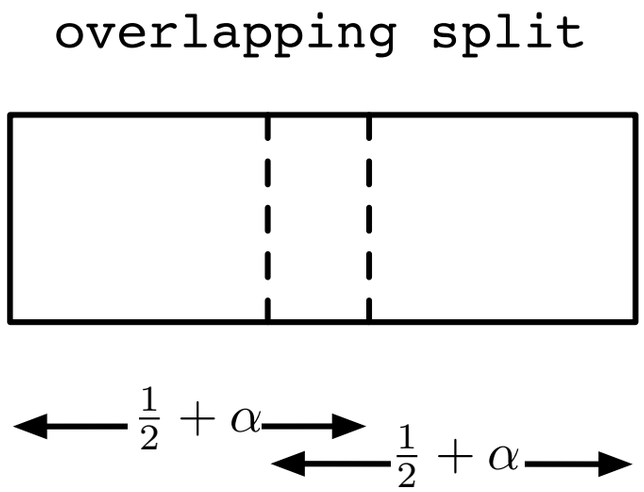
Exact NN search using spill trees



K-d with defeatist search:
answer a query q by the NN in q's leaf node

Frequently not the true NN.
Degrades with dimension.

Spill tree (Liu, Gray; Mount et al; others): same thing, but with
- overlapping cells
- random split directions



Analysis of spill tree

For any set of n points $x_1, \ldots, x_n \in R^d$ and any query $q \in R^d$, let $x_{(1)}, \ldots, x_{(n)}$ be a reordering of the points by increasing distance from q. Then the probability of failing to get the exact NN is proportional to

$$\frac{1}{n} \sum_{i=1}^{n} \frac{\|q - x_{(1)}\|}{\|q - x_{(i)}\|}.$$

Open problem: a measure of the easiness of a dataset-query configuration.