Locality Sensitive Hashing

Boxun Zhang, Spotify

me

data scientist at Analytics Research, Spotify

PhD@TU Delft, NL

Spotify

leading streaming service

30M+ tracks

2B+ tracks

59 markets

75M+ users

30M+ subscribers

Spotify’s recommender system

Google News

things in common?

nearest neighbor search

given a data point, find other similar ones in a dataset

proximity search, similarity search

naive approach

for each point, calculate distance with all other points

repeat for all data point

so far so good?

$$O(n^2)$$

pair-wise comparison, too slow!

avoid pair-wise comparison

locality sensitive hashing

approximate pair-wise distance

two *nearby* points remain *nearby*
after a *projection* operation

projection

nearby

Find similar songs

Build similarity measure

collaborative filtering

user-item matrix

matrix factorization

latent factors

Search for similar songs

annoy

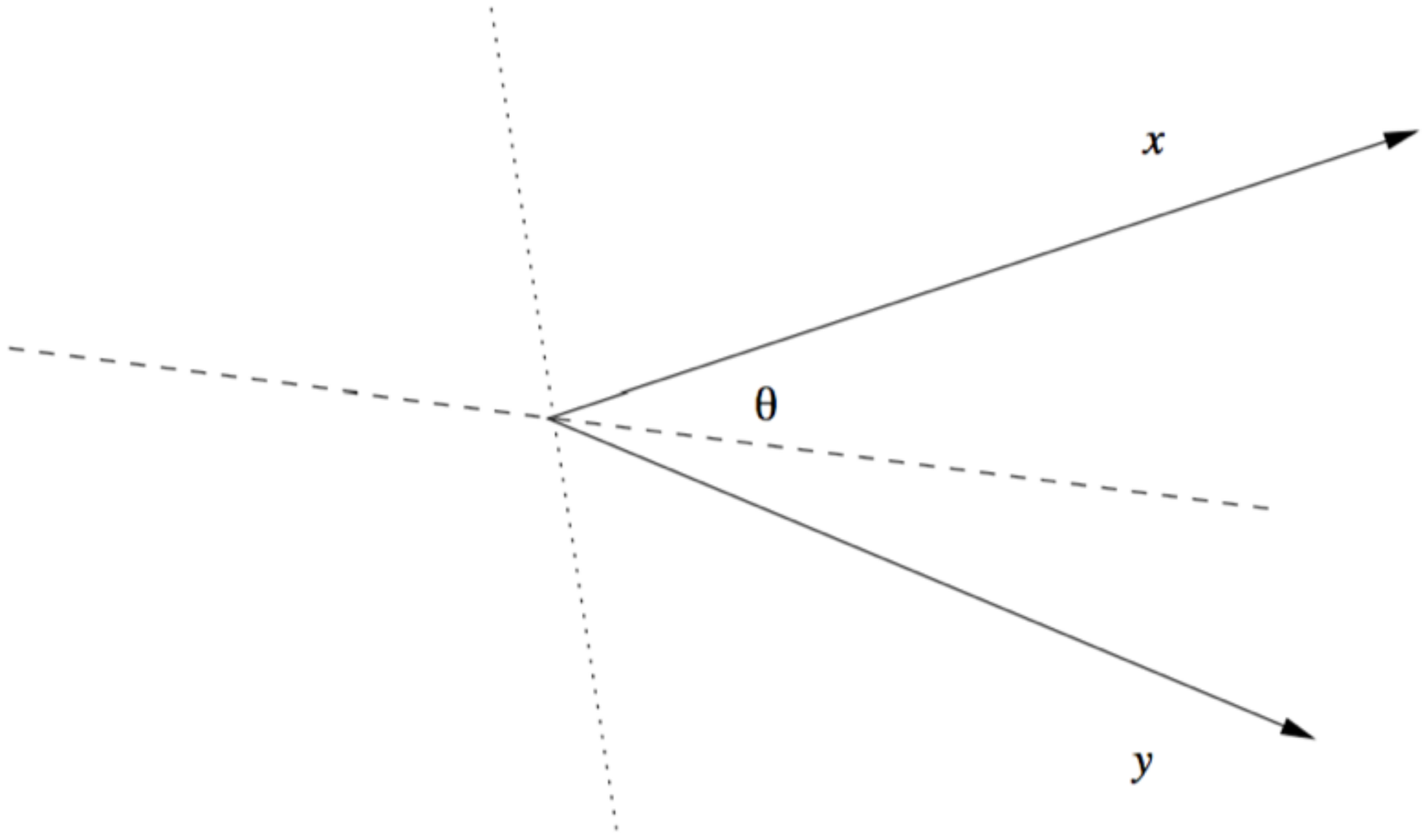
Approximate **N**earest **N**eighbors **O**h **Y**eah

How it works

build a binary tree in the data space

split space with random hyperplanes

build several of such trees — a forest


x
θ
y

Find similar songs

O(n) for each split

O(n) for build a tree

O(n) for build a forest

O(logn) for search

$$O(n) + O(n \cdot \log n)$$

github.com/spotify/annoy

LSH for euclidean distance

project vectors onto a line that is segmented into buckets of equal size

dot product

multiple projections

tuning the bucket size is non-trivial

more examples

To learn more

Mining of Massive Datasets, Ch.3

Locality-Sensitive Hashing for Finding Nearest Neighbors

Thank you, and we're hiring!