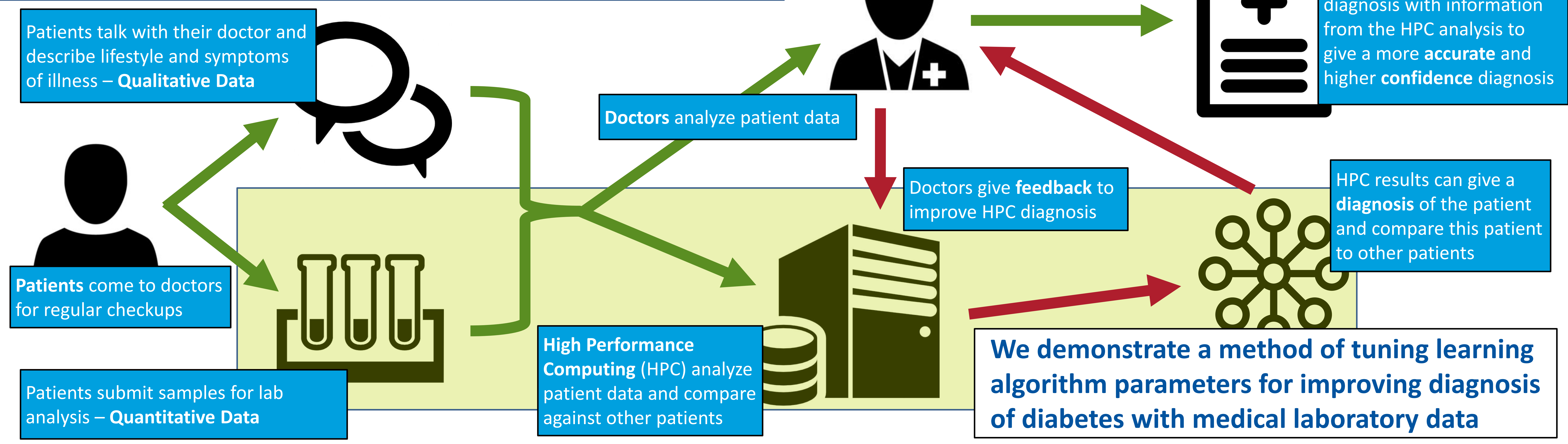
Parameter Tuning of DBSCAN for Medical Data and Diabetes Diagnosis

Michael Wyatt, Michela Taufer
University of Delaware: Global Computing Lab

Computation Assisted Diagnosis



Patients talk with their doctor and describe lifestyle and symptoms of illness – **Qualitative Data**

Patients come to doctors for regular checkups

Patients submit samples for lab analysis – **Quantitative Data**

Doctors analyze patient data

High Performance Computing (HPC) analyze patient data and compare against other patients

Doctors review HPC results

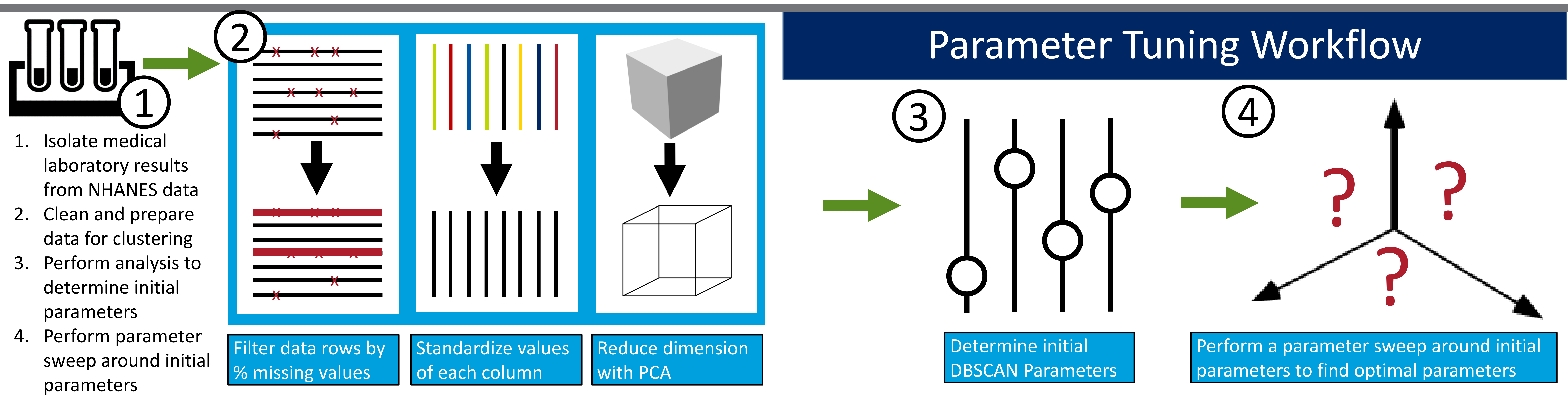
Doctors give **feedback** to improve HPC diagnosis

Doctors combine their diagnosis with information from the HPC analysis to give a more **accurate** and higher **confidence** diagnosis

HPC results can give a **diagnosis** of the patient and compare this patient to other patients

We demonstrate a method of tuning learning algorithm parameters for improving diagnosis of diabetes with medical laboratory data

Parameter Tuning Workflow



1. Isolate medical laboratory results from NHANES data
2. Clean and prepare data for clustering
3. Perform analysis to determine initial parameters
4. Perform parameter sweep around initial parameters

Filter data rows by % missing values

Standardize values of each column

Reduce dimension with PCA

Determine initial DBSCAN Parameters

Perform a parameter sweep around initial parameters to find optimal parameters

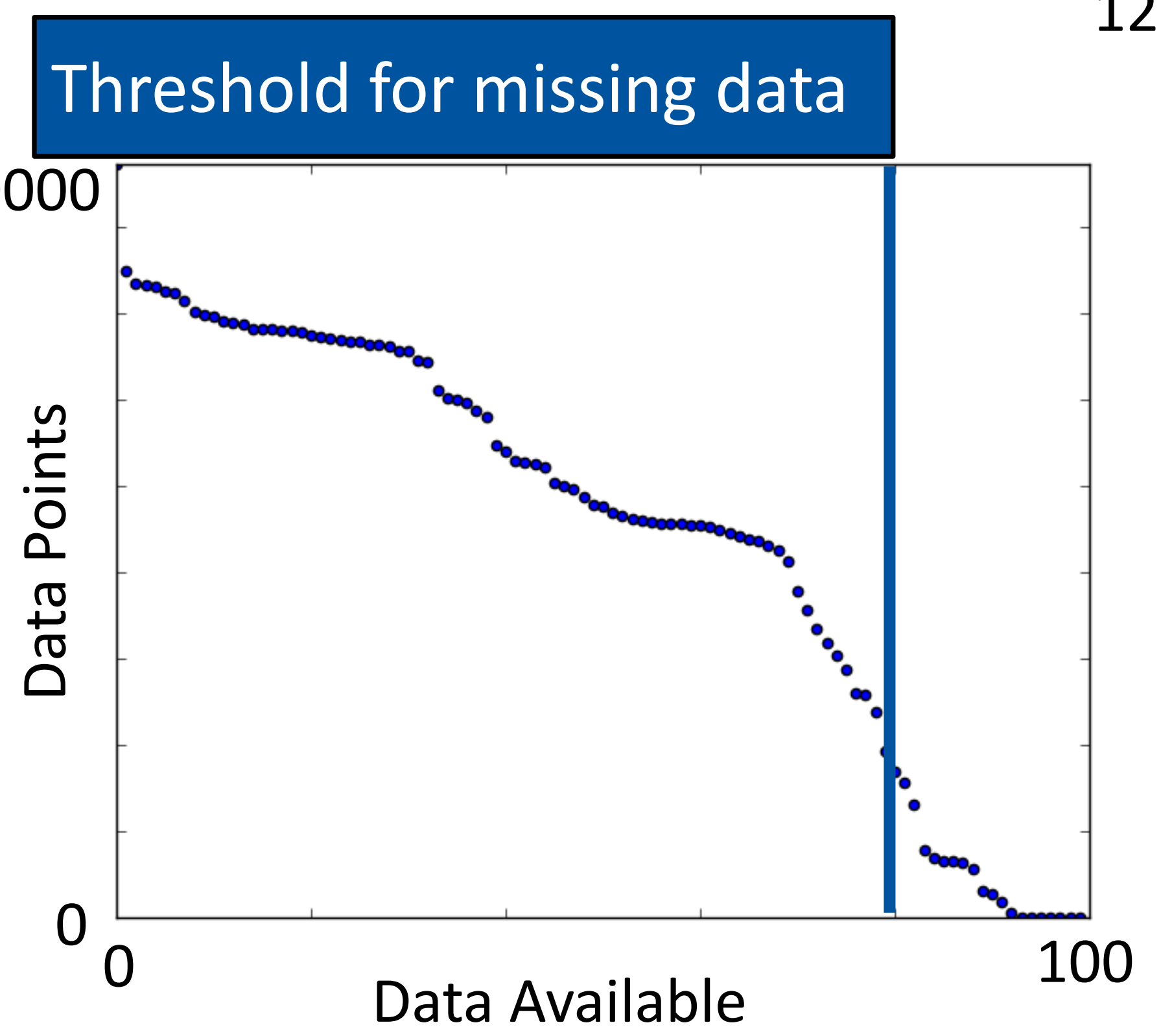
Parameter Tuning Results

- Missing data **trade-off**
  - No Missing Values → **Small Dataset**
  - Many Missing Values → **Bad Clusters**

Must find a balance between missing values and dataset size

- Patients with > 80% lab data available
  - 16,627 patients (over 1/5 original data)
  - Produces higher quality clusters

Threshold for missing data



- DBSCAN parameters affect cluster quality
  - **Epsilon**: Neighborhood to search for neighbors
  - **Min_pts**: minimum neighbors to be in a cluster

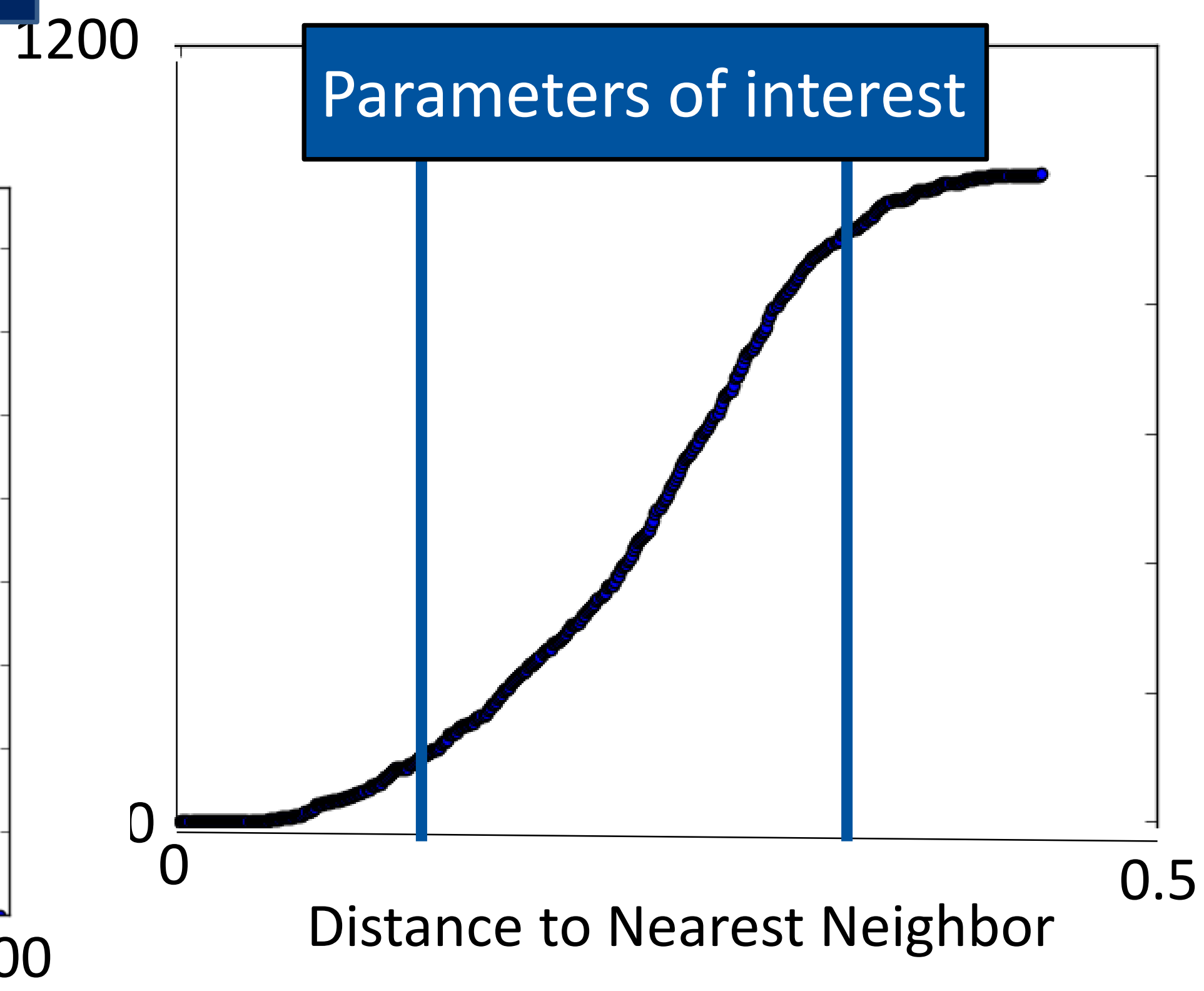
  } Parameters define cluster **density**
- Distance metrics also affect cluster quality: **Euclidean** vs. **Cosine**

- Utilizing nearest neighbor analysis, we can determine the range of epsilon values which should be tested
- We cluster data with several **epsilon** and **min_pts** values around the identified optimal values
- We measure the quality of each clustering by **percentage of points clustered** and **information gained** by each clustering:

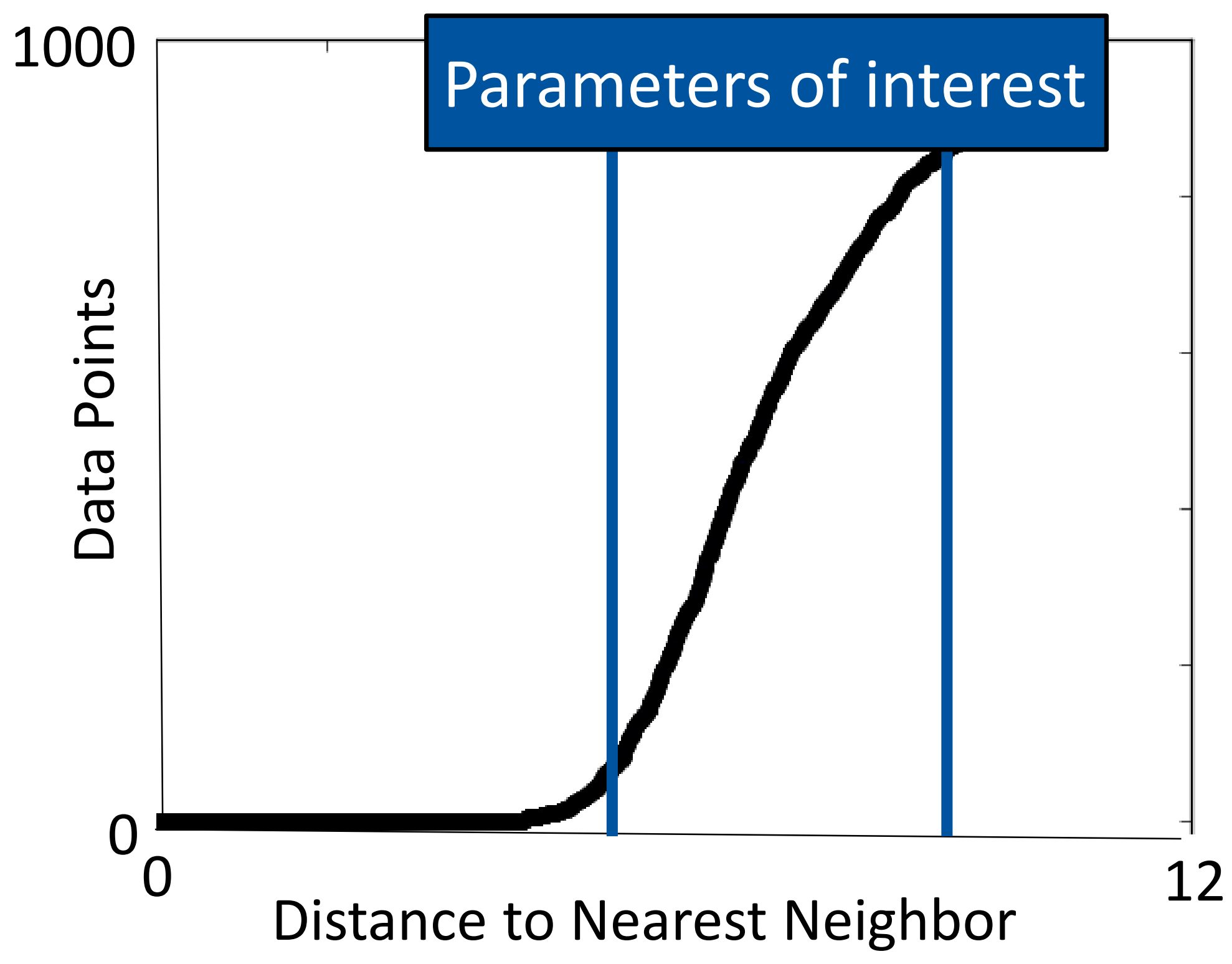
$$score = \frac{Diebetes\ Patients\ Clustered}{Total\ Diabetes\ Patients} * \frac{Informaion\ Gain}{Max\ Information\ gain}$$

- We identify an optimal parameter setting:
  - **Cosine distance, Epsilon: 0.15, Min_pts: 3**

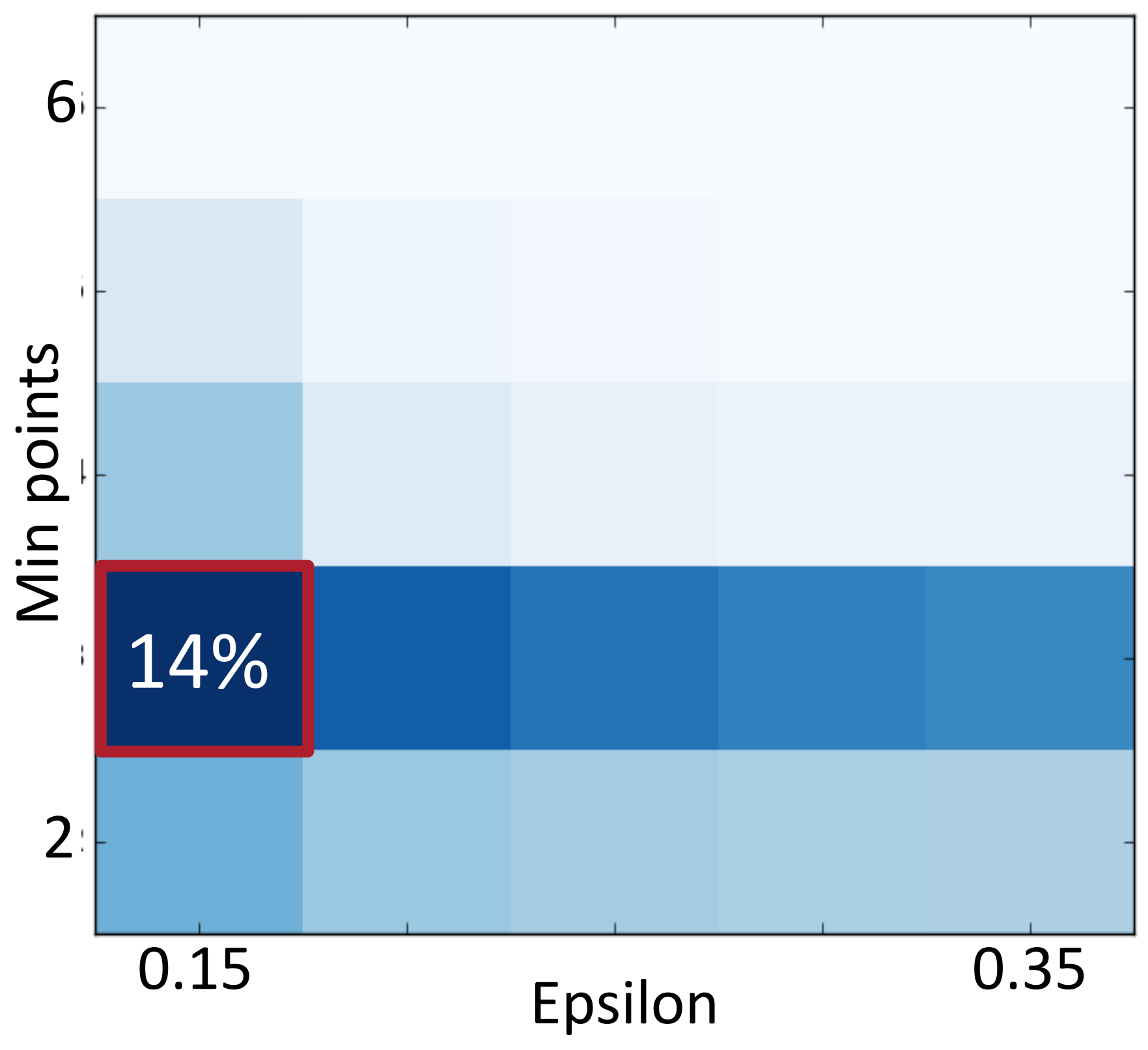
Cosine Distance



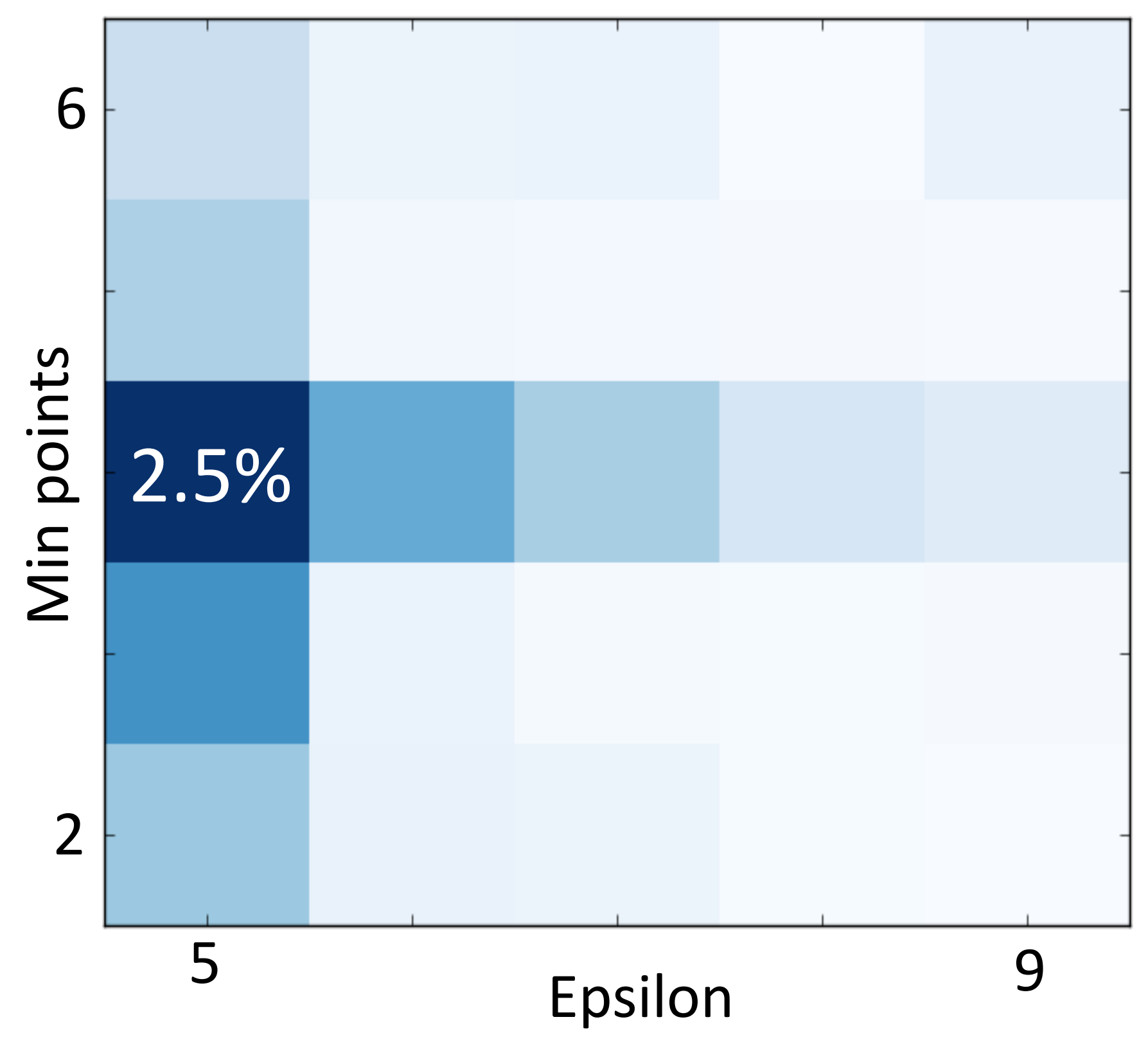
Euclidean Distance



Cosine Distance



Euclidean Distance


Min points
Epsilon