9 Unsupervised Learning

Unsupervised learning works with _____ data (no ____), making evaluation more difficult.

Density Estimation

Density estimation models the probability _____ function of the (unknown) distribution from which data were drawn.[1]

- Recall: We used a _____ model for density estimation in §7 to help with one-class classification: `gm = mixture.GaussianMixture(n_components=1)` estimates the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ of the multivariate normal distribution $N_D(\boldsymbol{\mu}_D, \boldsymbol{\Sigma}_{D\times D})$, which has density function

  $$f_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\mathbf{x}) = \frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)}{\sqrt{(2\pi)^D|\boldsymbol{\Sigma}|}}\,.$$

  For density estimation, use the same code to estimate $f_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\mathbf{x})$ given $\{\mathbf{x}_i\}$. (Just omit the _____ used to decide whether a new $\mathbf{x}$ is in the one class.)

- Recall: We used a *nonparameteric* model in _____ regression, a supervised learning method in §7. We used N Gaussians centered at $\{\mathbf{x}_i\}$ to make weighted averages of y's.

  In *kernel density estimation* (KDE), use the _____ of the same N Gaussians to estimate the probability density function $f(\mathbf{x})$ which generated the unsupervised (no y) examples $\{\mathbf{x}_i\}$.

  Consider the 1D case. Our kernel model is

  $$\hat{f}_b(x) = \frac{1}{Nb}\sum_{i=1}^{N} k\left(\frac{x-x_i}{b}\right),$$

  where b is a hyperparameter controling the underfit-overfit tradeoff and $k(x) \geq 0$ is a kernel with $\int_{-\infty}^{\infty} k(x)dx = 1$.

  As in §7, we use a Gaussian kernel, $k(z) = \frac{1}{\sqrt{2\pi}}\exp\left(\frac{-z^2}{2}\right)$.

  We can rewrite the model as

  $$\hat{f}_b(x) = \_\_\_\_\_$$

  which is the _____ of the N Gaussians $\{N(\mu = \_\_\_\_, \sigma = \_\_\_\_)|i = 1, \ldots, N\}$.[2]

[1] These methods improve upon just using a (density) _____ .

[2] The notation of Burkov and Wikipedia conceals the essential point that the model is an _____ of N Gaussians centered at $\{x_i\}$. Possibly they emphasize we are passing $\frac{x-x_i}{b}$ to a parameterless kernel.

Python

- To estimate $N_D(\boldsymbol{\mu}_D, \boldsymbol{\Sigma}_{D\times D})$, use `from sklearn import mixture` and `gm = mixture.GaussianMixture(n_components=1)` as in §7. Then `gm.fit(X)`, `gm.means_`, `gm.covariances_`, and `np.exp(gm.score_samples(X))` work as before. To learn more:
  - User guide: `https://scikit-learn.org/stable/modules/mixture.html`
  - Reference manual:
    `https://scikit-learn.org/stable/modules/generated/sklearn.mixture.GaussianMixture.html`
- For KDE:
  - `from sklearn.neighbors import KernelDensity`
  - `kde = KernelDensity(bandwidth=1.0, kernel='gaussian')`
    - b=`bandwidth` is the bandwidth
    - `kernel` is one of `'gaussian'` (the default), `'tophat'`, `'epanechnikov'`, `'exponential'`, `'linear'`, `'cosine'`; see Example link below for their shapes and effects
  - `kde.fit(X)` fits the model to the data.
  - `kde.score_samples(X)` gives log-likelihood of each `x` in `X`, so `np.exp(kde.score_samples(X))` gives $\hat{f}_b(x)$.

  To learn more:
  - User guide: `https://scikit-learn.org/stable/modules/density.html`
  - Reference manual:
    `https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KernelDensity.html`
  - Example:
    `https://scikit-learn.org/stable/auto_examples/neighbors/plot_kde_1d.html`[3]

[3] Click on "launch binder" to run it online. Change "N = 100" to "N = 10" to see kernels.

Clustering

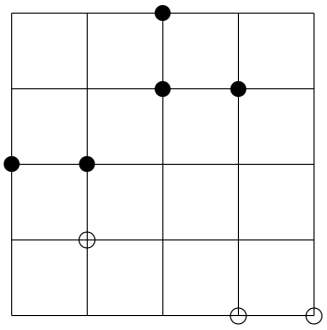
Clustering finds groups of _____ unlabeled examples and assigns a cluster ID to each example.[4] It is used in, e.g., exploratory data analysis, market segmentation, social network analysis, recommender systems, and stock sector analysis.

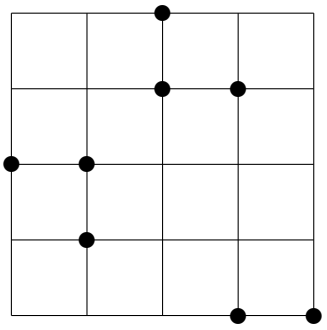
- *k-means clustering* maps each unlabeled example $\mathbf{x}$ to a cluster ID.
  - Choose the number of clusters _____.
  - Randomly choose one example to start each cluster as its _____ $\mathbf{c}$.
  - Label each example $\mathbf{x}$ with the centroid to which it is _____.
  - Recompute each centroid as the _____ of the examples labeled with it.
  - Repeat the last two steps until centroids _____.

  k is a hyperparameter typically decided by an educated guess.

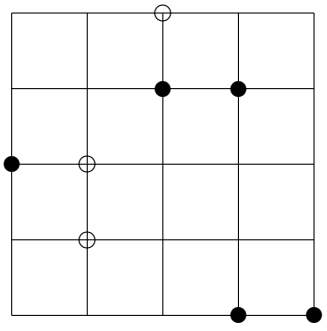
  k-means seeks to minimize _____, the sum of squared distances of examples to their respective centroids. Avoid poor results by running it _____ times.

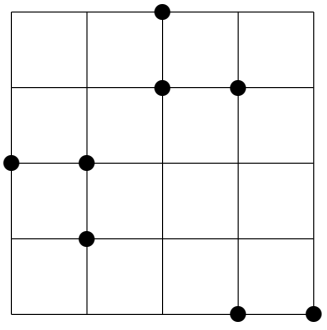
  e.g. Run k-means with $k = 3$, starting with the three unfilled points as cluster centers.

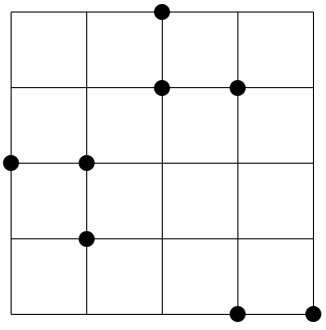




  e.g. Run k-means with $k = 3$, starting with the three unfilled points as cluster centers.







[4] Clustering labeled examples is not classification, as it _____ $\{y_i\}$.

Python

- `from sklearn.cluster import KMeans`
- `kmeans = KMeans(n_clusters=8, n_init=10, random_state=0)`:
  - `n_clusters` is the number of clusters to be found
  - `n_init` is the number of times k-means is run, each with different centroid seeds
  - `random_state=0` determines centroid initialization
- `kmeans.fit(X)` computes the clusters
- `kmeans.labels_` gives the labels (cluster IDs) of each $\mathbf{x}$ in the training `X`
- `kmeans.cluster_centers_` gives coordinates of the cluster centers
- `kmeans.predict(X)` gives the closest cluster for each $\mathbf{x}$ in `X`

To learn more:

- User guide: `https://scikit-learn.org/stable/modules/clustering.html#k-means`
- Reference manual:
  `https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html`
- Examples:
  2D: `https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_plusplus.html`
  3D: `https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_iris.html`

- *DBSCAN*[5] is a _____ *-based clustering* algorithm that puts $\mathbf{x}$ in a cluster if it is _____ to many points in that cluster.

  Hyperparameters:

  - ϵ is a distance threshold (_____).
  - n is the _____ number of examples in a cluster.

  Definitions:

  - $\mathbf{x}'$ is a _____ of $\mathbf{x}$ if its distance to $\mathbf{x}$ is $\leq \epsilon$.
  - $\mathbf{x}$ is a _____ example if its neighborhood size is at least n.
  - An _____ (or *noisy* example) has no neighbors.

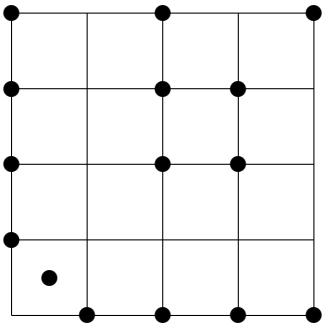
  Algorithm:

  - For each unexamined core example:
    - Make its neighborhood a _____.
    - _____ add core-example neighbors of this cluster's examples.
    - Add non-core example neighbors (_____ recursively) of cluster examples.
  - Call remaining examples _____.

[5] "DBSCAN" refers to "Density-based spatial clustering of applications with noise."

DBSCAN builds clusters with an _____ shape. (k-means builds hyper-_____ clusters.) Choosing ϵ and n is not easy. Increasing n or decreasing ϵ requires higher density to make a cluster. DBSCAN cannot effectively handle clusters of _____ density.

e.g. Run DBSCAN with $\epsilon = 1$ and $n = 2$:



Python

- `from sklearn.cluster import DBSCAN`
- `db = DBSCAN(eps=0.5, min_samples=5, metric='euclidean')`
  `eps` is ϵ, `min_samples` is n, and `metric` options include those we used in k-NN
- `db.fit(X)` computes the clusters
- `db.labels_` gives labels (cluster IDs) of each `x` in the training `X`; noisy examples get `-1`
- `db.core_sample_indices_` gives indices of core samples
- For each $k \neq -1$ in `db.labels_`, we can find neighbors in cluster k:

```
is_in_cluster_k = (db.labels_ == k)
is_core_sample = np.zeros(shape=db.labels_.shape)
is_core_sample[db.core_sample_indices_] = True
is_neighbor_of_cluster_k = (is_in_cluster_k & ~is_core_sample)
```

To learn more:

- User guide: `https://scikit-learn.org/stable/modules/clustering.html#dbscan`
- Reference manual:
  `https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html`
- Example:
  `https://scikit-learn.org/stable/auto_examples/cluster/plot_dbscan.html`

- *HDBSCAN* improves upon DBSCAN and can handle clusters of _____ density. (It drops ϵ. Details are omitted.) Try it _____.

  To learn more:

  - User guide: `https://scikit-learn.org/stable/modules/clustering.html#hdbscan`
  - Reference manual:
    `https://scikit-learn.org/stable/modules/generated/sklearn.cluster.HDBSCAN.html`

There is a _____ comparison of many clustering methods at `https://scikit-learn.org/stable/modules/clustering.html`.

Dimensionality Reduction

Dimensionality reduction maps $\mathbf{x}$ into a vector with _____ features to reduce correlation among features, reduce noise, visualize data (we see only 2D or 3D), and facilitate interpretable models.

- *Principal component analysis* (PCA) fits a new coordinate system to $\{\mathbf{x}_i\}$ where each new _____ is called a *principal component* (PC):
  - Each PC is a _____ *vector* (length 1).
  - The first PC is the direction of the _____ of the data $\{\mathbf{x}_i\}$. (It is the _____ axis of a "minimal" ellipsoid enclosing the data.)
  - For $i > 1$, the ith PC is orthogonal[6] to the first $i - 1$ PCs and in the direction of the _____ greatest variance in the data.

  A helpful picture is Figure 7 on p. 15 of

  `https://www.dropbox.com/s/y9a7b0hzmuksqar/Chapter9.pdf?dl=0`.

  To do dimensionality reduction, we choose some number p of dimensions ($0 < p < D$) and _____ each $\mathbf{x}_i$ onto the first p PCs, transforming the D-dimensional $\mathbf{x}_i$ into a smaller p-dimensional example. Burkov omits details.

  Benefits of PCA:

  - PCA does _____ while retaining most of the information, saving memory, disk space, and computation time.
  - PCA can mitigate the _____ *of dimensionality*: as D increases, the "_____" of the feature space increases faster than the available data, which become _____. Many elementary models/algorithms/insights are not designed for sparse data.

    e.g. The number of D-digit binary numbers in $\{0, 1\}^D$ is _____. The number of D-digit decimal numbers in $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}^D$ is _____.

    e.g. We need $N = 10^D$ points to sample each unit interval/square/cube/hypercube from $[0, 10]^D$. ___ Draw $[0, 10]^D$ and unit hypercubes for each $D \in \{1, 2, 3\}$.

  - PCA does *feature* _____. (It creates several important new features as linear combinations of original features. This is not feature _____.)
  - The first _____ PCs often account for most of the data variability, so even high-dimensional data can be visualized in 2D or 3D.

[6] Two vectors $\mathbf{a}$ and $\mathbf{b}$ are *orthogonal* if $\mathbf{a} \cdot \mathbf{b} = 0$. In 2D, orthogonal means _____.

Python

```
from sklearn.decomposition import PCA
```

- `pca = PCA(n_components=None, random_state=0)` keeps `n_components` PCs; using `None` keeps all D components
- `pca.fit(X)` learns `n_components` PCs from $\{\mathbf{x}_i\}$ in `X`
- `pca.components_` gives PCs (axes/directions of maximum variance in the data)
- `pca.explained_variance_ratio_` gives % of variance explained by each PC
- `pca.transform(X)` applies dimensionality reduction to each $\mathbf{x}$ in `X`

To learn more:

- User guide: `https://scikit-learn.org/stable/modules/decomposition.html`
- Reference manual:
  `https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html`

Perspective

"Burkov has undertaken a very useful but impossibly hard task in reducing all of machine learning to 100 pages. He succeeds well in choosing the topics–both theory and practice–that will be useful to practitioners, and for the reader who understands that this is the _____ 100 (or actually 150) pages you will read, not the _____, provides a solid introduction to the field."

—Peter Norvig, Research Director at Google and author of *Artificial Intelligence: A Modern Approach*