Erdös and Renyi (ER)

- randomly choose n out of N(N-1)/2 possible vertices
- Parameter p: each of the N(N-1)/2 possible connection is chosen with prob. p. Expected number of edges is N(N-1)/2 * p
- Degree Distribution: Poisson
- diameter: ln(N)
- clustering coef: 1/N
- homogeneous networks influenced by variance of Poission variance

Barabasi/Albert (BA)

- starting with a small number, vertices are added with m connections each
- Prob of edge depends on degree of vertex: $k_i / \sum_j k_j$
- Degree Dist. : Power law
- diameter: ln(N) / lnln(N)
- clustering coef: 1/N^0.75
- heterogeneous graphs since power law is long-tail distribution with large var
- scale free, hubs/cliques, small world

- BA more suited for web, since it reflects common phenomena like presence of hubs, the scale-free property, and small-world networks
- Tldr:
  - ER generates random graphs where edges are formed independently with a fixed prob. p between any pair of nodes
  - BA creates creates scale-free networks using preferential attachment, where new nodes are more likely to connect to already well-connected nodes (hubs)

Poisson dist: probability of x events in a given time if average rate lambda is known

$$P(x) = \exp(-\lambda) \cdot \lambda^x / x!$$

Expectation and variance lambda, both finite

Power law: $P(x) \sim c \cdot x^{-\gamma}$

$E(x^m) = c/(\gamma - m - 1)$

expectation finite for γ>2

variance finite for γ>3

long tails, probably infinite var

for every vertex v pagerank r(v) should be high if pointed to by many sites with high pagerank and selective links

$$r(v) \sim \sum_{w \in pa(v)} \frac{r(w)}{|ch(w)|}$$

$$r := \epsilon/n + (1-\epsilon) \cdot B \cdot r$$

$$r := (\epsilon/n \cdot \mathbf{1} + (1-\epsilon) \cdot B) r$$

Introduce random jumps vector r, with probability epsilon = 0.15

- GPT = generative pretrained transformer
- generative: produces text via sampling from next word probability, which is modelled by means of Markov Model (with many parameters)
- transformer: specific neural architecture, which models the probabilities; acts on time window with attention heads
- pretrained: parameters are trained huge corpora from the web via next word prediction + human-feedback reinforcement learning

Models the liklihood of a random web surfer landin on a page, combining contributions from linked pages (B * r) and random jumps (epsilon / N) to ensure stability and handle disconnected components

lower bound: given by the first summand (epsilon / N) plus normalization
upper bound is N number of nodes if all mass is centered on the page

Pages without incoming links (parents) rely only on the random jump term, so they have a low PageRank. Pages with many parents have a higher PageRank because they receive contributions from multiple terms, depending on their parents' PageRank.

Normalized cut decreases as the number of clusters increases, which can lead to trivial solutions

- cut $cut(C_1, \ldots, C_k) := \frac{1}{2} \sum_{i=1}^{k} W(C_i, V \backslash C_i)$
- ratio cut $ratiocut(C_1, \ldots, C_k) := \frac{1}{2} \sum_{i=1}^{k} W(C_i, V \backslash C_i) / |C_i|$
- normalized cut $ncut(C_1, \ldots, C_k) := \frac{1}{2} \sum_{i=1}^{k} W(C_i, V \backslash C_i) / vol(C_i)$

where $vol(A) = \sum_{i \in A} d_i$ and $d_i = \sum_j w_{ij}$ degree of i

CUT should not be used for clustering, as it favors small imbalanced clusters, NCUT more reasonable ensures that clusters are balanced and meaningfull

One cluster:
- Cut yields 0, QE yields sum of squared dist. of points to the centre
- N clusters: CUT yields sum of weights, QE yields 0 when choosing prototypes as data points itself

PCA

- find the linear projection of the data which preserves as much variation in the data as possible
- Invariant against orthogormal operation
- PCA, ICA, LDA are all deterministic, work on vectors and are linear parametric projections

ICA

- find linear projection such that output features are as independent as possible
- ICA regularizes the problem by aiming for such sources s which are maximum independent
- ICA cannot recover sources which are all Gaussian since linear mixtures of Gaussians remain Gaussian, cannot recover exact scaling of the sources and their order

LDA

- given labels, find linear projection such that variance of points with same label is minimized, variance of points in between different labels is maximized
- Maximum dim. of the LDA projected space is C-1, C being number of classes

	objective	type	effort	data type
PCA	manifold learning / reconstruction error wirh orthogonal sources	linear, parametric	O(N)	vector
ICA	reconstruction error with independent sources	linear, parametric	O(N)	vector
LDA	separation of classes	linear, parametric	O(N)	vector
MDS	distance preservation	nonparametric	O(N^2)	vector or distances
Isomap	distance preservation / manifold learning	nonparametric	O(N^3) O(N^2 log N)	vector or distances
LLE	manifold preservation	nonparametric	O(N^2)	vector
Laplacian eigenmap	correlation of graph distance and euclidean embedding viaeigenvalues of graph Laplacian	nonparamtric	O(N^2) - O(N^3)	vector or distances
t-SNE	preserve probabilities induced by distances	nonparametric	O(N^2)	vector or distances
UMAP	Preserve connections on manifold	Nonparametric	almost O(N) if fast neighbor search is used	Vector or distances

Parametric

- PCA: x → W x
- SOM: x → position of closest w in a lattice
- GTM: x → most likely position in latent space
- autoencoder: x → f(x) → f^{-1} f(x)

Non-parametric

MDS, SNE, t-SNE, MVU, LLE, Isomap, Sammon, CCA, Laplacian Eigenmap

Nearest neighbour desc.

- efficient algo to approx. nearest neighbours
- iteratively improve initial random or approx. neighbor graph by focusing on pairs of points that are likely to yield better neighbours

t-SNE:

- goal: find low-dimensional vectors such that the original dist. of pairwise points is preserved as much as possible
- match pairwise dist in original space and pairwise dist in projection space
- take gaussian in orginal space and sutdent-t in projection space
- pros: moderate original diastance can be modelled by much larger distance in projection, accuracy of large distances in projection space is not so important
- optimize $KL(p_{ij}|q_{ij})$ with gradient descent

Laplacian Eigenmap:

- deterministic
- local graph
- weighting: one for every link or heat kernel
- compute degree martrix
- compute graph laplacian
- embed via generalized eigenvectors

MDS:

- find vector to preserve distances
- Constructs a configuration of points in the desired dim.
- Minimizes the difference between the original distances and the distances in the low-dim space
- Assumes linear relationships between distances

Isomap

- input: vectors or pairwise distances
- parameters: k or epsilon for neighbourhood computation
- output: low-dimensional projection of points
- objective: preserve geodesic distances on data manifold
- Algo: Compute neighbourhood graph, compute shortest distances on neighbourhood graph, project preserving distances using Multidimensional Scaling
- effort: dominated by shortest paths in sparse graphs hence N^2logN, rest is N^2
- is non-linear
- goal: find real vectors xi such that the original manifold distances are preserved as much as possible
- works if intrinsic data manifold is two-dim and flat (swiss roll) fails if sampling is not sufficient to get proper neighbourhood graph
- non-parametric hence does not directly give you out-of-sample extension.

Locally linear embedding (LLE)

- given: real vectors
- goal: find low-dimensional vectors such that the original local data structure is preserved as much as possible
- step 1: define local neighbourhood
- step 2: find linear relation in neighbourhood
- local structure should be preserved
- step 3: find mapped points such that the linear relationships are similar
- is a deterministic dimensionality reduction method

UMAP

- tries to preserve local topological structure
- compute knn graph, weight connections by a term, cost given by cross-entropy

Method	vector	dissimilarity	deterministic	parametric
LLE	X		X	
UMAP	X	X		
PCA	X		X	X

Explain objectives of ratio cut (CUT) and quant. error (QE), NCUT (Normalized Cut)

- NCUT: addresses issue by dividing the edge weights by the total degree of each cluster, balancing the cluster. It ensures that both clusters arre meaningful and of reasonable size
- ratio cut: given weighted graph, decompose weighted graph into clusters such that the sum of weights which are pointing outside a cluster normalized by the size of the cluster is minimized
  quantization error: given vectors, find prototypes representing clusters and an assignment of data points to these prototypes such that the sum of (squared) distances of data points to their prototypes is minimum

Measure Reduction Dim. Methods

Entropy: Measures randomness or disorder of a variable. Max. entropy means the data is evenly distributed. FOr a fixed variance, entropy is hgihest if the distribution is Gaussian
Relevance: Evaluate structure of complex data, determine rather data follows gaussian dist.

Negentropy: Measures deviation of a dist. from a Gaussian dist. It is calculated as the difference between the entropy of a Gaussian dist. and the entropy of the actual data
- Used to measure non-Gaussianity

Mutual Information: Measure degree of dependence between multiple variables. If two variables are independent, their mutual information is 0. $I(t_1,\ldots,t_n) = \sum_i H(t_i) - H(t_1,\ldots,t_n)$

MI and negentropy are inversely corr. to any source, only if data is whitened

Prototype-based Models

	LVQ1	LVQ2.1	GLVQ	RSLVQ
Cost Func.	no	Yes but unbounded	Yes: correlates to margin optimization	Yes: optimize posterior probability for GMM

First step: each prototype w_j is implicitly represented via a N-dimensional vector α_j of coefficients representing $w = \sum_i [\alpha_j]_i x_i$, whereby $[\alpha_j]_i \geq 0$ and $\sum_i [\alpha_j]_i = 1$.
This allows a computation of (squared) distances: $d(x_i, w_j) = [D \cdot \alpha_j]_i - 0.5 \cdot \alpha_j^t D \alpha_j$
Relational k-means clustering results by enteriung this into the vectorial algorithm:

init α_j
repeat
compute $d_{ij} = [D \cdot \alpha_j]_i - 0.5 \cdot \alpha_j^t D \alpha_j$
determine receptive field $R_j = \{x_i \mid d_{ij} \leq d_{ik} \forall k\}$
determine coefficients $[\alpha_i]_i = 1_{x_i \in R_i} / |R_i|$

GLVQ

$$\sum \Phi \left(\frac{d(\vec{x}_i - \vec{w}_+) - d(\vec{x}_i - \vec{w}_-)}{d(\vec{x}_i - \vec{w}_+) + d(\vec{x}_i - \vec{w}_-)} \right)$$

Max. Margin with Loss function

LVQ is supervised learning

Shape:
- GMLVQ: elipsoidal shape, with arbitary rotated axes
- GLVQ: round shape, standard unit sphere
- GRLVQ: elipsoidal shape, with axis-aligned symmetry-achses
- LVQ: round shape

LVQ, GLVQ, GMLVQ are convex

GMLVQ: Λ should be positive definite, can be achieved through reparametrization: Λ = Ω*Ω^T

GLVQ uses euclidian^2 and GMLVQ uses adaptive squared euclidian

	Metric parameters
Relevance learning (GRLVQ)	Adjust feature weigthing only
Metric learning (GMLVQ)	Adjust quadratic form via adaptive linear transormation of data
Local (lGMLVQ or LGRLVQ)	Adjust individual parameters per receptive field
Low rank (LiRaMLVQ or use eigenvactors of matrix in GMLVQ)	Use a low rank matrix for the data transformation

Robust soft LVQ (RSLVQ)

data are given as Gaussian mixture over the classes:
$p(\vec{x}|W) = \sum_c \sum_{c(\vec{w}_j)=c} p(\vec{x}|\vec{w}_j) P(\vec{w}_j)$
$p(\vec{x}|\vec{w}_j) = (\sigma/2\pi)^{n/2} \exp(-\sigma/2 \|\vec{x} - \vec{w}_i\|^2)$
$P(\vec{w}_j)$ uniform distribution

probability of data labeled by c:
$p(\vec{x}_i, y_i|W) = \sum_{c(\vec{w}_j)=y_i} p(\vec{x}_i|\vec{w}_j) P(\vec{w}_j)$

optimize the ratio **Log-Likelihood-Ratio**
$\sum_i \log(p(\vec{x}_i, y_i|W)/p(\vec{x}_i|W))$

benefit: the term is bounded → no divergence

LVQ1: yields good results in typical model situations

LVQ2.1: without window does not, diverges, strange behaviour when using Early Stopping

$d^+ := d(\vec{x}^i, \vec{w}^+)$ closest prototype with label $= y^i$
$d^- := d(\vec{x}^i, \vec{w}^-)$ closest prototype with label $\neq y^i$

$\sum_{i=1}^m f_{\text{LVQ}}(d^+, d^-)$ where $f_{\text{LVQ}}(d^+, d^-) = \begin{cases} d^+ & \text{if } d^+ \leq d^- \\ -d^- & \text{otherwise} \end{cases}$

$\sum_i f_{\text{LVQ2.1}}(d^+, d^-)$ with $f_{\text{LVQ2.1}}(a, b) = \mathbf{1}_{\text{window}}(a - b)$

GLVQ: Has linear boundary and the two receptive fields are convex
GMLVQ: Similiar since choosing Λ = Ω*Ω^T. Linear transformation preserves the shape of a linear classification boundary and receptive fields are convex
LGMLVQ: Boundary has quadratic form, receptive fields can be seperated by nonlinear boundaries, might not be convex
Example of not convex receptive fields, which both have two connected components 'two cigar' example: w_1 = w_2 = (0,0)^T, Lambda_1 = (1,1)^T(1,1), Lambda_2 = (-1,1)^T(-1,1)

	k-means / NG	spectral	affinity propagation	relational NG
objective	quantization error	(normalized /ratio) graph cut	quanatizatio n error for exemplars	quantization error in kernel space
data	vectors	(sparse) graph with similarities	(sparse) similarity matrix	dissimilarity matrix
representati on	prototypes	cluster indices	exemplars	prototypes as implicit linear combination
OOS	yes	Needs additional effort	yes	yes
parameters	cluster prototypes and assignments	assignments to cluster	exemplars and assignments and number of clusters	

	silhouette	Davis Bouldin	Calinski Harabasz	gap
range	[-1,1]	>0	>0	usually >0
which k	large value	small value	large value	smallest k with large value
k=1	no	no	no	yes
data type	every distance	vectors	vectors	vectors

	method	non i.i.d.	fixed part of D
Nyström	matrix based techniques: • spectral clustering • relational NG	no	yes
patch	prototype based techniques: • relational NG • AP	yes	no

Methods for large data sets or data streams:
BIRCH (balanced iterative reducing and clustering using hierarchies), Path neural gas, Patch relational neural gas, Patch affinity propagation

k-means

goal: find prototypes so they min quantization error:

$$E = \frac{1}{2} \sum_{\vec{w}^i} \sum_{\vec{x}^j \in R(\vec{w}^i)} \|\vec{w}^i - \vec{x}^j\|^2$$

k-means++: init centroids in better way
start with one center x
take next center x with probability proportial to D(x) where D(x) = distance from already chosen centers
repeat until k prototypes are chosen

perform k-means
- unsupervised learning
- distance-based

First step: each prototype w_j is implicitly represented via a N-dimensional vector α_j of coefficients representing $w = \sum_i [\alpha_j]_i x_i$, whereby $[\alpha_j]_i \geq 0$ and $\sum_i [\alpha_j]_i = 1$.
This allows a computation of (squared) distances: $d(x_i, w_j) = [D \cdot \alpha_j]_i - 0.5 \cdot \alpha_j^t D \alpha_j$
Relational k-means clustering results by enteriung this into the vectorial algorithm:

init α_j
repeat
compute $d_{ij} = [D \cdot \alpha_j]_i - 0.5 \cdot \alpha_j^t D \alpha_j$
determine receptive field $R_j = \{x_i \mid d_{ij} \leq d_{ik} \forall k\}$
determine coefficients $[\alpha_i]_i = 1_{x_i \in R_i} / |R_i|$

Spectral Clustering

- SC requires non-negative pairwise similarities as input, missing values are ok and implicitly treated as zero. Vectorial data need to be transformed using e.g. neighborhood graphs. The number of clusters k is an input.
- Cannot handle vectorial data

Affinity propagation

- Treats missing info as -inf
- does not need number of clusters
- uses self similarities
- New cluster is build whenever the costs induced by the self-similarity of the clusters exemplar are smaller than the costs when using a smaller number of clusters measured in terms of the quant. error

Neural Gas

- Algorithm that adapts a set of prototype vectors to represent the strcuture of the input data by min. a distortion measure. Prototypes updated in a neighbourhood-based manner, with the influence of updates decreasing over time and distance from the winning prototypes

Method	cost function	parameters	out-of-sample
spectral clustering	normalized/ratio cut	assignments to cluster	no
kmeans++	quantization error	cluster prototypes and assignments	yes
affinity propagation	(exemplar based) quantization error	exemplars and assignments and number of clusters	yes

Gap statistics

Method to determine number of clusters deviation of clustering coefficient from expected value for baseline

$E(\log W_k) - \log W_k$ pick smallest k where this is significantly larger than 0

where
$W_k = \sum_{i=1}^k d_i/(2n_i)$
n_i = number of points in cluster i
d_i = sum of pairwise distances in cluster i
E refers to expectation over reference distribution

Silhouette

scaled difference of distances between clusters and within clusters. Should be large

$S_i = (b_i - a_i)/\max\{a_i, b_i\}$
where
a_i = average distance of point i to points in its cluster
$b_i = \min_j$ average distance of point i to points in cluster j

average over all points i, centered at 0, no reasonable value for k=1

Calinski Harabasz

ratio between cluster variance and within cluster variance Not suited for k=1

$\frac{SS_B}{SS_W} \cdot \frac{N-k}{k-1}$
where
SS_B = between cluster variance: $\sum_{i=1}^k n_i (m_i - m)^2$
with m_i = mean of cluster i, m = data mean, n_i size cluster i
SS_W = within cluster variance: $\sum_{i=1}^k \sum_{x \in C_i} (x - m_i)^2$

Davis Bouldin

ratio within cluster and between cluster distances, should be small

$\sum_{i=1}^k \max_{j \neq i} D_{ij}/k$
where
$D_{ij} = (\bar{d}_i + \bar{d}_j)/\bar{d}_{ij}$
$\bar{d}_i$ = average distance of points in cluster i to cluster center
$\bar{d}_{ij}$ = distance of cluster centers i and j

normalized by k, nonnegative values not reasonable value for k=1

external/internal clust. measures

Internal tries to evaluate result of clustering by using given data and clust. prescription. Does not know ideal clusters a priori. External refer to priorly given classes which are characterized by labels for each data point and evaluate in how far these are respected by the clustering

External: Adjusted Rand index (ARI) measures the similarity between pred. clusters and ground truth by counting matching pairs of points. Not suited when no labeled data
Internal: Silhouette Score: See up. Not suited for clusters of varying density or non-spherical shapes

Challenges when learning on data streams:
- Label information, what are good learning signals?
- Algorithmic challenge, how to efficiently update model for new data points
- Model selecteion challenge, how to asjust model complex. and hyper para
- Efficient memory models, how to store required information

Method	Dealing with drift	Model
iSVM	no	Kernel method
Learn++	Implicit virtual drift	Ensemble
VFDT	Window	Decision tree
JIT	Drift detection	Generic concept (eg kNN)
LVGB	Drift detection	Ensemble
ARF	Drift warning and detection	Ensemble of trees
SAM	ITTE based adaptive window	kNN
SAME	Drift detection and ITTE based adaptive window	Ensemble of kNN

Name	Type	Method
DDM	Error-based	Statistics of error
ADWIN	Error-based	Significant distance of mean error
HDDDM	Distribution-based	Hellinger distance for bins
PCA-CD	Distribution-based	Difference of PCA projections
DAWIDD	Time-dependency	Change of time characteristics in space
Shape method	Time-dependent shape fit	Detects rapid drift

Concept drift is change in the relationship between input data and the target variable over time. Common in dynamic systems.
Real concept drift: Occurs when the actual underlying dist. of the target variable changes. Means the concept of what is being predicted evolves
Virtual concept drift: Occurs when the dist. of the input data changes, but the realationship between input and target variable remains the same