

## Unsupervised learning

clustering

density estimation

parametric  
ULearnif

non-parametric

(1) construct gaussian mixture models

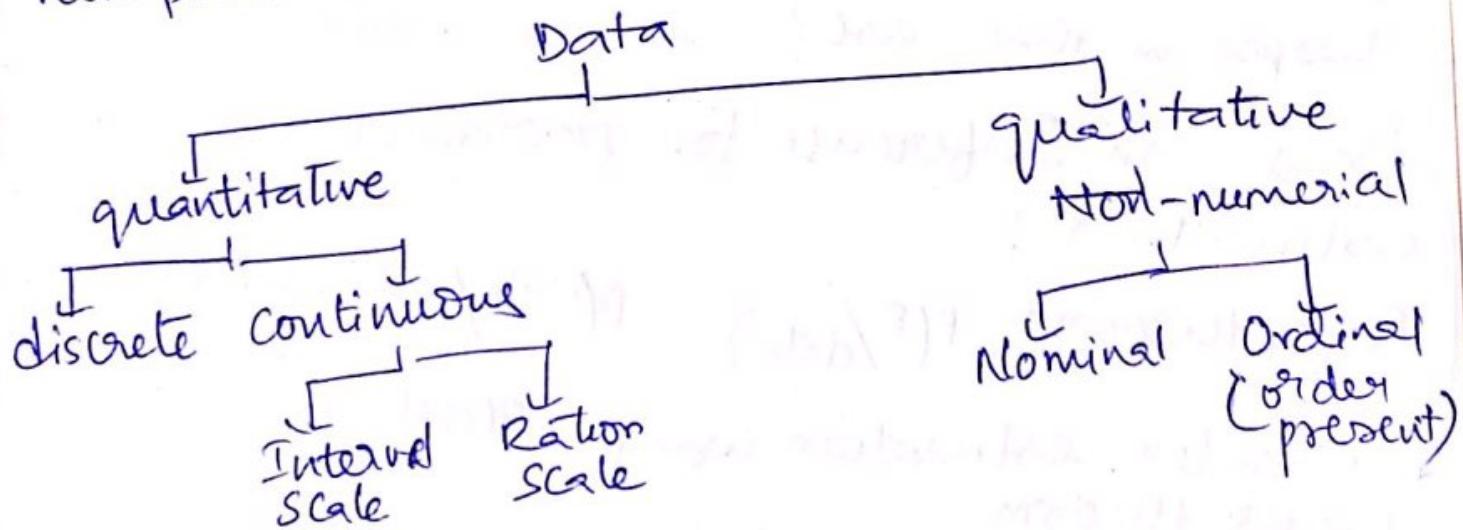
(2) Use expectation maximization algo

clustering as a mixture of gaussians

"parametric vs non-parametric distribution".

Different statistical distributions : (next page)

\* If data is ordinal/interval based, only non-parametric statistics can be used



## sequential learning

mapping input seq to output seq using state machines. hidden state seq present.

## Active learning.

Theory of rational agency: (action selection theories)

\* Density estimation ((how using deep generative NN's?))  
→ estimating probability density function  
of random variable in a population  
from sample's help.

q: Difference between probability density fn &  
probability distribution?  
+ n

q: what is maximum likelihood estimation?  
→ finding the values of parameters that result  
in best fit curve.

\* Likelihood & loglikelihood

$$L(\mu, \sigma; \text{data}) = P(\text{data}; \mu, \sigma)$$

q: When is least squares minimization same  
as max likelihood estimation? Why does it  
happen in that case? How

Bayesian Inference for parameter  
estimation:

$$\text{Bayes theorem: } P(\theta/\text{data}) = \frac{P(\text{data}/\theta) \times P(\theta)}{P(\text{data})}$$

Parameter estimation using  
Bayes theorem

"prior distribution".

$\theta \rightarrow$  set of parameters ( $\theta = \{\mu, \sigma^2\}$  for gaussian  
distribution)

$P(\theta/\text{data}) \rightarrow$  posterior distribution

$P(\theta) \rightarrow$  prior distribution

$P(\text{data}) \rightarrow$  evidence & data =  $\{y_1, y_2, \dots, y_n\}$

$\hookrightarrow$  normalizing const (helps making  $\sum P(\theta/\text{data}) = 1$ )

Can we use bayesian inference for classification problems? How? Is it used for discrete data/ continuous or both?

Different statistics from the posterior distribution & their physical significance!

- expected value = mean
- variance → uncertainty
- mode  $\neq$  MAP estimate.

"Gaussian distribution is conjugate to itself w/ gaussian likelihood function."

(Latent Dirichlet Allocation algo) \*

Markov Chain Monte Carlo methods  $\rightarrow$  to calculate posterior distribution

Updating beliefs iteratively in real time.  
using bayesian inference  $\rightarrow$  kalman filter.  
Prior acts as a regularizer ~~here~~ in bayesian inference.

q. When does MAP estimate equal MLE?

Overfitting due to Bayesian priors  $\rightarrow$  Pending

Marginalization ↴

$$P(X) = \int_y P(X|Y=y) dy.$$

What is discriminant analysis?

When is it used?

when dependant variable is categorical & predictor/independant variable is interval.

develop discriminant fn as a linear combination of independant variables to discriminate between categories of dependant variable.

Discriminant analysis vs Analysis of variance  
(vs) Regression analysis

Correlation is not causation  $\star$

probability distribution

continuous

e.g.: Normal  
standard normal

discrete  
e.g.: Binomial  
Poisson

## MCMC methods

→ monte carlo simulations

→ markov chains (are memoryless)

\* bell curve, law of large nos.

Markov → Non independant events may also conform to patterns

(In long run, dist gettle to pattern)

\* if events are subj to fixed prob.,

interdependant events conform to average.

Q: How can bayesian inference be used to quantify uncertainty in predictions?

MCMC → Random sampliy of parameters in probabilistic space to approximate the posterior distribution in Bayesian inference.

Where can we use these posterior distributions?

How

- quantifying uncertainty
- comparing models
- generating predictions

Central limit theorem & law of large nos

Covariance vs Correlation vs Causation

## Statistical distributions

q; data discrete/continuous?

q. symmetry of data & outliers scenario.

q. upper & lower limits of data

q. likelihood of occurrence of extreme values

Discrete distributions:

→ Binomial

→ Poisson

→ Negative binomial

→ geometric

→ Bernoulli

Continuous

→ Normal

→ Exponential

→ logistic Cauchy

→ gamma

→ chi-squared

q. lets say we have a distribution which is generated by combining multiple commonly known distributions. How do we find those & separate those distributions?

quantifies uncertainty of estimated skill of model

quantifies uncertainty in single forecast

Joint distributions:

Discriminant analysis:

Exponential regression:  $y = \alpha e^{\beta x}$

Can be converted to linear

power regression:  $y = \alpha x^\beta$

confidence interval vs prediction interval (w) tolerance interval

q. finding confidence interval using different means.

q. How to get confidence interval for mean given we have a sample?

Methods to get prediction intervals:

→ bootstrap resampling

→ delta method

→ bayesian method

→ mean-variance estimation method

## Resampling methods:

- cross validation → cluster based
- bootstrapping → random oversampling
- Random oversampling
- SMOTE
- under sampling

Probability mass fn (vs) probability density fn

## Hypothesis testing:

- null hypothesis
- p value & critical value
- traditional testing vs Bayesian testing

## Marginal distributions

X, y are jointly distributed random variables

Entropy → list of generalized entropies  
most widely used → Shannon's entropy

$$\Phi_1(P) = - \sum_{i=1}^n P_i \log P_i$$

Decision tree → info gain →  $\Phi_{\text{parent}} - \sum w_i \times \Phi_{\text{child}}$   
 ↓ gini index  $\hookrightarrow 1 - \sum_i p_i^2$        $w_i = \frac{n_{\text{child}}}{n_{\text{parent}}}$

Hinge loss → SVM

grid search → best combination of hyperparameters

random search → tries for best fit

Skewness, kurtosis, coefficient of variation

hyperparameter optimization

Data imputation methods: → for cross-sectional datasets  
 → for time series data

Cases:

- MCAR
- MAR
- NMAR

① Using mean/ median values :

→ poor results on encoded categorical features

② Using Mode

→ works with categorical features

→ can introduce bias

③ Using KNN algo.

→ impute library  
KDTree

KNN sensitive to outliers unlike SUM.

④ using multivariate imputation by chained eqn.

→ multiple imputations

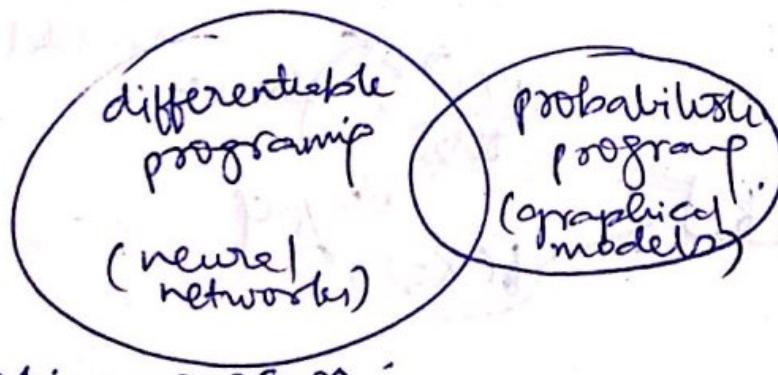
⑤ Using D2 .

⑥ using stochastic regression/extrapolation & extrapolation

Uncertainty quantification

→ gaussian process (GP) models:

(multivariate problems)



gaussian process:

→ use prior knowledge to make predictions.

→ assign probability to diff ofns possible to fit a dataset, getting mean of prob dist to get

→ incorporate confidence.

most probable ans.

multivariate gaussian distribution :

- each random variable → normal distributed
- their joint dist → also normal

→  $\mu$  → mean

$\Sigma$  → covariance matrix (symmetric & positive semidefinite)  
↳ gives  $\sigma_i^2$  &  $\sigma_{ij}$

Semi supervised learning → model trained on dataset → small portion labelled data  
/ step 1: cluster similar data into groups of similar data (unsup part)

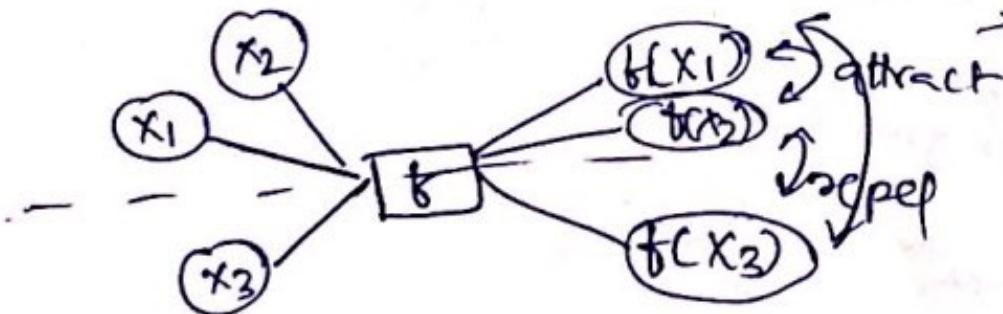
step 2: label the remaining data in grp seeing the labelled data in the same grp.

↳ majority unlabeled data.

manifold assumption →

contrastive learning:

adversarial ML  
→ data poisoning  
→ evasion attack  
→ model extraction



Few shot learning:

few-shot learning problem

data level approach → parameter level approach

↳ limit parameter space

- gan's
- data augmentation
- using base datasets to

N-way-k-shot-Classification :

learn how to learn to classify.

Meta-learning algs

graph embedding methods & node embedding

→ learning multiple embeddings for a node

embedding

transductive

approaches

deep learning

factorization  
methods

random  
walks

\* gaussian distribution based graph embedding  
(includes uncertainty estimation)

for:

→ node classification

→ link prediction

→ community detection

1) Matrix factorization:

→ using adjacency matrix → most simple method

→ using (locally linear embedding) (LLE):

$$e_i = \sum_{j \in N_i} w_{ij} \times e_j$$

$$\phi(e) = \sum_{i \in N_i} (e_i - \sum_{j \in N_i} w_{ij} \times e_j)^2$$

2) HOPE:

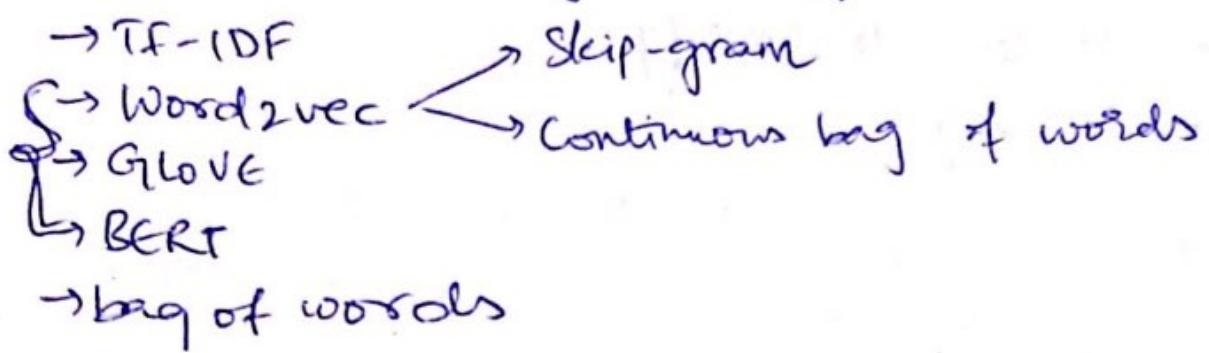
$$\phi(e) = \sum_{ij} (e_i e_j^\top - \underbrace{s_{ij}}_{\text{similarity b/w nodes } i, j})^2$$

(using Adamic/Adar  
↑ similarity)

$$A(x, y) = \sum_{u \in N(x) \cap N(y)} \frac{1}{\log(N(u))}$$

3) DeepWalk

## word embedding techniques:



Gaussian process ← (set of random variables)

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \sim N(\mu, \Sigma) \quad x \sim N(\mu_x, \Sigma_{xx})$$

$$\Sigma = \text{cov}(x_i, x_j) = E[(x_i - \mu_i)(x_j - \mu_j)^T]$$

Marginalization & conditioning

$$P_{x,y} = \begin{bmatrix} x \\ y \end{bmatrix} \sim N(\mu, \Sigma) = N\left(\begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}, \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix}\right)$$

$$y \sim N(\mu_y, \Sigma_{yy})$$

$$P_x(x) = \int_y P_{x,y}(x,y) dy = \int_y P_{x|y}(x/y) P_y(y) dy$$

$$x/y \sim N(\mu_x + \Sigma_{xy} \Sigma_{yy}^{-1} (y - \mu_y), \Sigma_{xx} - \Sigma_{xy} \Sigma_{yy}^{-1} \Sigma_{yx})$$

$$y/x \sim N(\mu_y + \Sigma_{yx} \Sigma_{xx}^{-1} (x - \mu_x), \Sigma_{yy} - \Sigma_{yx} \Sigma_{xx}^{-1} \Sigma_{xy})$$

each random variable has index i.

i<sup>th</sup> dimension of  $\xrightarrow{n \text{ dimensional multivariate distribution}}$

# Optimization techniques

problems

single variable fns

no constraints

multivariable fns

both equality and inequality constraints

equality constraints

(Lagrange multiplier method)

inequality constraints

(Kuhn-Tucker condns)

## Numerical methods of optimization

linear program

integer program

quadratic program

non linear program

stochastic program

dynamic program

combinatorial optimization

constraint satisfaction

## Advanced optimization techniques

hill climbing

genetic algorithms

(inheritance, mutation, selection, crossover / recombination)

## correlation

(diff from causation)

## Feature selection

Supervised methods

wrapper      filter

unsupervised  
methods

Built - in feature selection

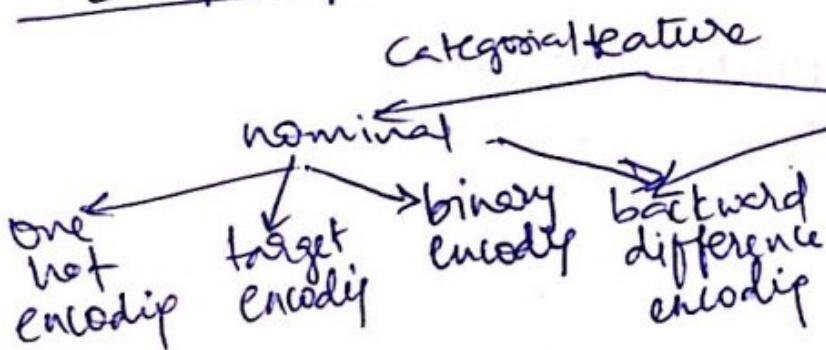
penalized regression models

lasso      decision tree  
randomforest

feature selection (vs) Dimensionality reduction

univariate statistical measures

		output numerical	output categorical
		Pearson's	anova
input numerical	output numerical	Spearman's	kendall's
	output categorical	anova	chi squared
input categorical	output categorical	kendall's	mutual information



q. Handling categorical data in ML

label encoding

Handling missing data in time series :

- Last observation carried forward
- Next observation carried backward
- Linear interpolation
- Spline interpolation
- When seasonally present, 1. deseasonalize  
2. interpolate 3. seasonalize

## Outlier analysis

univariate

multivariate

- Normalize the data
- check z-score of each datapoint
- if  $x \notin [-3, 3]$ , then outlier
- using IQR
- ORC (outlier removal clustering → uses kmeans)

## Feature selection

using correlation

↓  
 Q does it  
 depict only  
 linear dependency?  
 or a more complicated  
 function also?

using P-value  
 (not recommended),

## Markov chain models

- history of prev event is known.
- prob. of transition from 1 event to another can be measured
- prob. of future events can be computed

set of states

MC :  $(S, X, P)$  random variables  
 HMM :  $(N, M, A, B, \pi)$  prob. for initial state  
 hidden states  
 observable event  
 prob. dist of observable output

## Natural Language Understanding

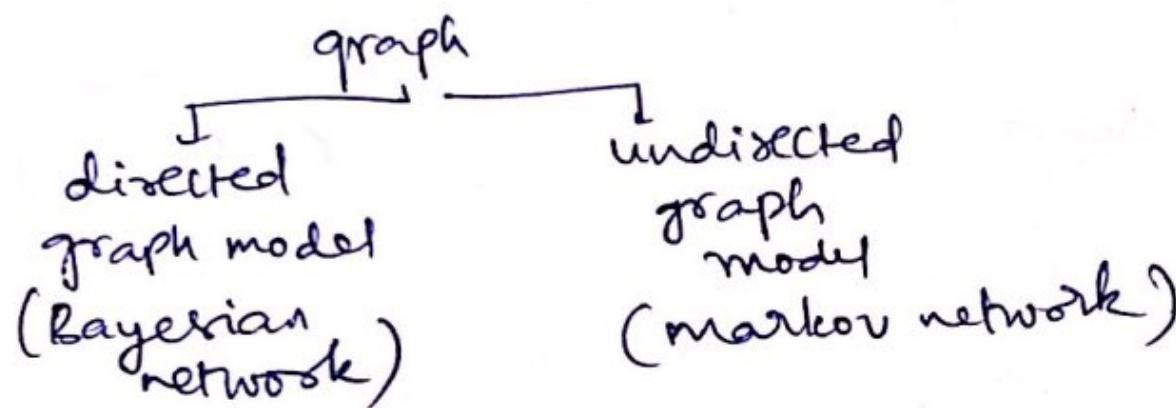
- intent recognition
- entity recognition → entity

named entities      numeric entities

# NLP, NLU, NLG

- bayesian networks
- maximum entropy
- conditional random field
- matrix factorization

## graphical models



## dynamic graphs

- structure of graph changes with time
- node classification → node attribute inference
  - link prediction → recommender systems travelling salesman problem
  - community detection
  - graph classification

node attributes can be time series

- node classification in homogeneous graph
- page rank, centrality measures : Baseline models

manual feature engineering to augment vocabulary-based feature vectors with graph-related node features.

\* Methods to calculate quantitative values of structural position of a node in a graph.

centrality measures  
examples

- degree centrality
- betweenness centrality
- closeness centrality
- eigenvector centrality

GNN → utilize ~~scalar~~ relationships in training NN on graphs (eg: GCN)

$$\text{GCNN layer} \rightarrow z = \sigma \left( A' F w + b \right)$$

Diagram illustrating the components of a GCNN layer:

- ↑ output
- ↑ activation layer
- ↑ graph structure (graph adjacency matrix)
- ↓ node features (matrix)
- ↓ trainable parameters

Markov chain monte Carlo :

- class of algos for systematic random sampling from high-dimensional probability distributions
- drawing samples where next sample is dependent on prev sample
- Gibbs sampling & Metropolis Hastings algo.
- when exact inference is intractable from a complicated probability distribution

(Bayesian vs,  
Frequentist)

# Methods to estimate prediction interval

- ensemble ANN
- bayesian method ] will be cooked into deeper later.
- monte carlo method
- bootstrap method
- LUBE method (predictly lower & upper bounds)

## Monte Carlo sampling

→ doesn't work well in high-dimensions

\*① random walk

\*② Gibbs sampling, metropolis - hastings algo } algs based on MCMC

## MCMC :

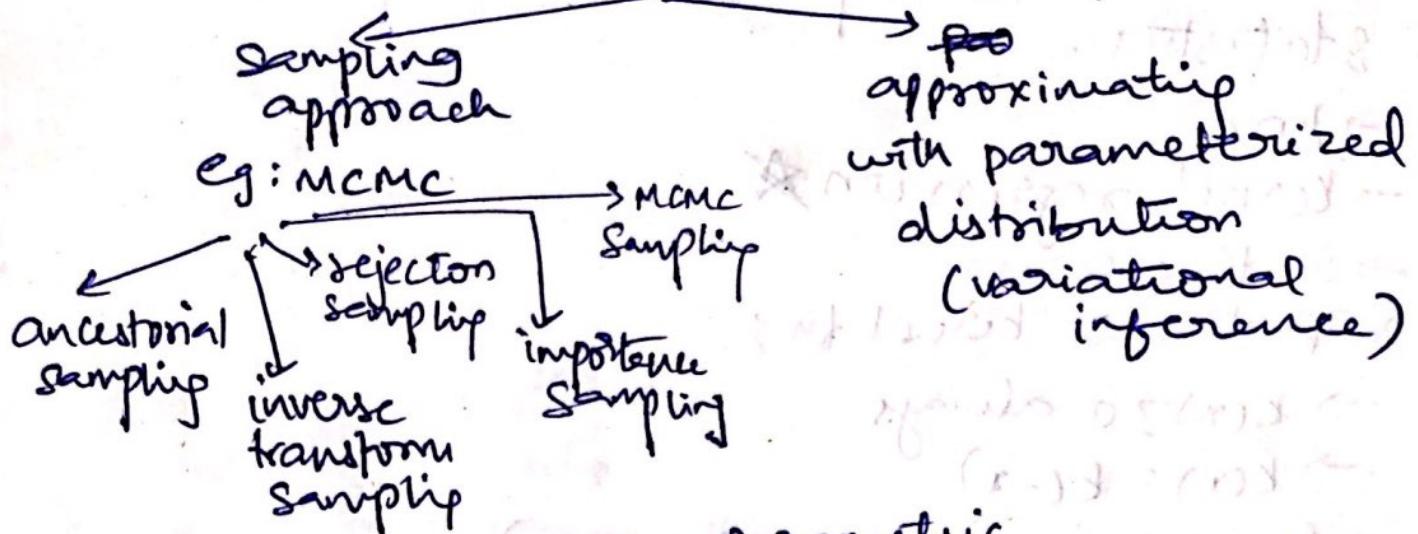
- pick a random parameter value to consider
- generate random values continuously
- (but subject to some rule that determines that it's a good value for the parameter)
- give pair of values for the parameter, can compute which is better.  
(how likely each value explains data given prior beliefs.)  
i.e. posterior distribution
- if latest random value is better, then prev one in the chain, add it with a certain probability value determined by how much better it is.



## contrastive divergence (CD)

useful for training unstructured graphical models like RBMs

# Approximating inference value in high dimensions



- Density estimation
  - parametric
  - non parametric
- using deep generative neural networks ↗
- (→ can run a statistical test to check if a random sample fits a distribution)
- \* skewness } → can be handled to find the perfect distribution for sample data.
- \* outliers }

transformations → log / square root / power transforms

~~not~~ eg: boxcox

Non parametric density estimation ~~AKA~~

Kernel smoothing / Kernel density estimation

1. epanechnikov

$$\text{kernel : } k(n) = \begin{cases} \frac{3}{4}(1-n^2) & \text{for } |n| \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

giving diff weights to observations with a diff distance.

$$2. \text{ gaussian kernel } k(n) = \frac{1}{\sqrt{\pi}} e^{-n^2/2}$$

$$3. \text{ simple moving average kernel } \rightarrow k(n) = \frac{1}{2}$$

Kernels are widely used in nonparametric statistics.

→ KDE

→ Kernel regression \*

→ in time series

properties of kernel fn:

→  $k(x) \geq 0$  always

→  $k(x) = k(-x)$

→  $\int k(x) dx = 1$  (normalization)

→ (for many kernels)  $k(x) = 0 \quad x \in (-\infty, -1) \cup (1, \infty)$

\* Moving average with bandwidth  $b$ :

For time series  $x(1), x(2), \dots, x(n)$

where  $x(i)$  is observed value at time  $t=i$

$$MA(i) = \frac{1}{\min(i+b, n) - \max(i-b, 1) + 1} \sum_{k=\max(i-b, 1)}^{\min(i+b, n)} x(k)$$

where  $b$  is bandwidth

\* Multidimensional Scaling (MDS)

→ generating vector embedding for nodes in a graph.

by positioning nodes of graph in ~~n-dim~~ space while preserving the distances between them

geodesic distances

used in isomap

dimensionality reduction

## MDS

metric  
(classical)  
(principal  
coordinate  
analysis)

non-metric

on ordinal data

→ model  
similarity/dissimilarity  
btw data by calculating  
dist btw each pair  
of points using  
geometric coordinates

multicollinearity & factor analysis ~~(Covered later)~~  
Dimensionality reduction(DR) ( DR (rs) feature select)

→ too:

1) img compression

2) remove noise

① factor analysis →

② take care of multicollinearity →