Unsupervised learning → clustering
→ density estimation

- Parametric Ulearning
  1. construct gaussian mixture models
  2. Use expectation maximization algo
- non-parametric

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

- Data
  - quantitative
    - discrete
    - continuous
      - Interval scale
      - Ration scale
  - qualitative
    - Non-numerical
      - Nominal
      - Ordinal (order present)

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 NNs?))
  → Estimating probability density function of random variable in a population from sample's help.

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

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:

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

* Parameter estimation using Bayes theorem
"prior distribution".
θ → set of parameters ($\theta = \{\mu, \sigma\}$ for gaussian distribution)
$P(\theta/\text{data})$ → posterior distribution
$P(\theta)$ → prior distribution
$P(\text{data})$ → evidence & data $= \{y_1, y_2, \cdots, y_n\}$
↳ normalizing const (helps making $\Sigma P(\theta/\text{data}) = 1$)

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 ⇒ uncertainity
→ mode ≠ MAP estimate.

"gaussian distribution is conjugate to itself wrt gaussian likelihood function."

(Latent Dirichlet Allocation algo) ✱

Markov Chain Monte Carlo methods → to Calculate posterior distribution

✱✱ Updating beliefs iteratively in real time. using bayesian inference → kalman filter.

Prior acts as a regularizer ~~here~~ in bayesian inference.

q when does MAP estimate equals MLE?
Overfitting due to Bayesian priors ⟹ 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 descriminate between categories of dependant variable.

Discriminant analysis (vs) Analysis of variance.
(vs) regression analysis
Correlation is not causation ✱

probability distribution

continuous
eg: Normal
Standard Normal

discrete
eg: 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 settle to pattern)
* if events are subj to fixed probs.,
interdependant events conform to average.

q. How can bayesian inference be used to quantify uncertainity in predictions?

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

~~Where~~ can we use these posterior distributions?
~~How~~
→ quantifying uncertainity
→ 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 occurence of extreme values

Discrete distributions:
→ Binomial
→ Poisson
→ Negative binomial
→ geometric
→ Bernoulli

Continuous
→ Normal
→ Exponential
→ logistic cauchy
→ gamma
→ chisquared

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

Joint distributions:

Discriminant analysis:

quantifies uncertainty of estimated skill of model

quantifies uncertainty in single forecast

Exponential regression: $y = \alpha e^{\beta x}$ → Can be converted to linear
power regression: $y = \alpha x^{\beta}$

confidence interval (vs) prediction interval (vs) tolerance interval

q. finding confidence interval ~~using~~ using different ~~means~~ distributions.
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 oversampling
→ bootstraping → random under sampling
→ SMOTE
→ Random oversampling

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 → $\phi_{parent} - \sum w_i \times \phi_{child}$
$w_i = \frac{n_{child}}{n_{parent}}$
→ gini index
↳ $1 - \sum_i p_i^2$

Hinge loss → SVM

grid search → best combination of hyperparameters for best fit
↳ random search

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

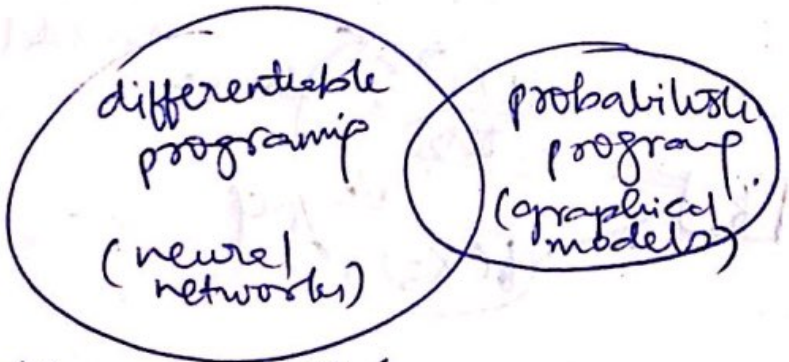
③ Using KNN algo.
→ impyute library
KDTree
KNN sensitive to outliers unlike SVM.

④ using multivariate imputation by chained eqn.
→ Multiple imputations

⑤ Using DL.

⑥ using stochastic regression/extrapolation & intrapolation

Uncertainity quantification
→ gaussian process (GP) models.
(multivariate problems)



gaussian process :
→ use prior knowledge to make predictions.
→ assign probability to diff fns possible for fitting a dataset, getting mean of prob dist to get most probable ans.
→ incorporate confidence.

multivariate gaussian distribution :

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

↳ μ → mean
Σ → covariance matrix (symmetric & +ve semidefinite)
↳ gives σ_i^2 & σ_{ij}

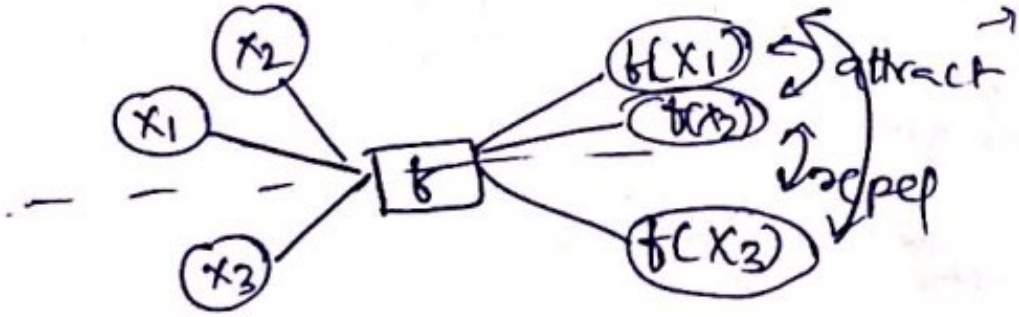
Semi supervised learning → model trained on dataset → small portion labelled data
↳ majority unlabelled data.

step 1 : cluster ~~similar~~ data into grps of similar data (unsup part)
step 2 : labelled remaining data in grp seeing the labelled data in the same grp.

manifold assumption →

adversarial ML
→ data poisoning
→ evasion attacks
→ model extracts

contrastive learning :



Few shot learning :

↳ meta learning problem

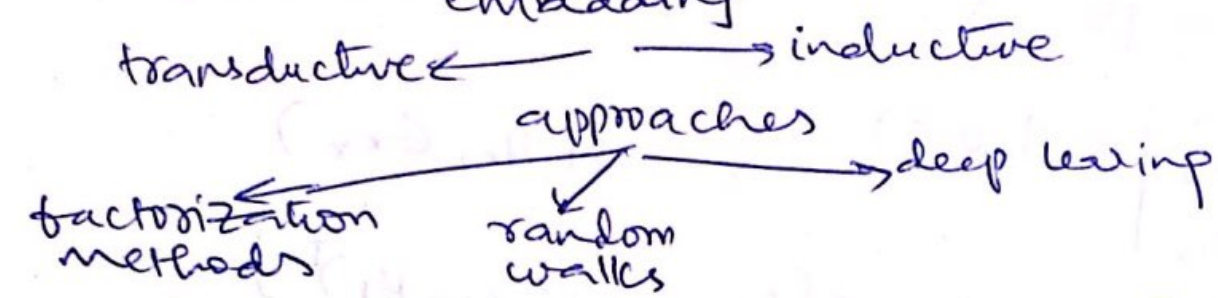
data level approach
↳ gans
↳ data augmentation
↳ using base datasets

parameter level approach
↳ limit parameter space

N-way-k-shot-Classification :
learn how to learn to classify.
Meta-learning algo

graph embedding methods & node embedding

→ learning multiple embeddings for a node



→ gaussian distribution based graph embeddig (includes uncertainty estimation)

for :
→ node classificatn
→ link prediction
→ community detectn

1) Matrix factorization :
→ using adjacency matrix → most simple method
→ using (Locally linear embedding)(LLE):

$$\epsilon_i = \sum_{j \in N_i} w_{ij} \times \epsilon_j$$

$$\phi(\epsilon) = \sum_i \left(\epsilon_i - \sum_{j \in N_i} w_{ij} \times \epsilon_j \right)^2$$

2) HOPE :

$$\phi(\epsilon) = \sum_{ij} \left(\epsilon_i \epsilon_j^T - S_{ij} \right)^2$$

S_{ij} ← similarity btw nodes i, j.
(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:

→ TF-IDF

→ Word2vec → Skip-gram, Continuous bag of words

→ GLOVE

→ BERT

→ bag of words

Gaussian process ← (set of random variables)

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \sim N(\mu, \Sigma) \qquad X \sim N(\mu_X, \Sigma_{XX})$$

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

Marginalization & conditioning

$$P_{X,Y} = \begin{bmatrix} X \\ Y \end{bmatrix} \sim N(\mu, \Sigma) = N\left(\begin{bmatrix} \mu_X \\ \mu_Y \end{bmatrix}, \begin{bmatrix} \Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_{YY} \end{bmatrix}\right.$$

$$Y \sim N(\mu_Y, \Sigma_{YY})$$

$$P_X(x) = \int_y P_{X,Y}(x,y)\,dy = \int_y P_{X/Y}(x/y)\,P_Y(y)\,dy$$

$$X/Y \sim N\left(\mu_X + \Sigma_{XY}\Sigma_{YY}^{-1}(Y - \mu_Y),\ \Sigma_{XX} - \Sigma_{XY}\Sigma_{YY}^{-1}\Sigma_{YX}\right)$$

$$Y/x \sim N\left(\mu_Y + \Sigma_{YX}\Sigma_{XX}^{-1}(x - \mu_X),\ \Sigma_{YY} - \Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XY}\right)$$

each random variable has index i.

↑ ith dimension of n dimensional multivariate distribution.

Optimization techniques

problems

- single variable fns
- multivariable fns
  - no constraints
  - both equality and inequality constraints
    - equality constraints (Lagrange multiplier method)
    - inequality constraints (Kuhn-Tucker conditions)

Numerical methods of optimization

- linear programg
- integer programg.
- quadratic programg
- non linear programg.
- Constraint satisfaction
- infinite dim optimization
- combinatorial optimization
- stochastic programg
- dynamic programg.

advanced optimization techniques

- hill climbing
- genetic algorithms (inheritance, mutation, selection, crossover/recombination)

Correlation (diff from causation)

Feature selection

Supervised methods → unsupervised methods

wrapper ↔ filter

Built - in feature selection

penalized regression models

Lasso | decision tree | randomforest

Feature selection (vs) Dimensionality reduction

univariate statistical measures

	output numerical	output categorical
input numerical	pearson's spearman's	anova kendall's
input categorial	Anova kendall's	chi squared mutual information

Q. Handling categorical data in ML

Categorial feature

nominal → one hot encodig, target encodig, binary encodig, backward difference encodig

ordinal → label encoding

Handling missing data in time series :

→ Last observation carried forward
→ Next observation carried backward
→ Linear interpolation
→ Spline interpolation
→ when seasonality present, 1. deseasonalize 2. interpolate 3. seasonalize

Outlier analysis

- univariate
- multivariate

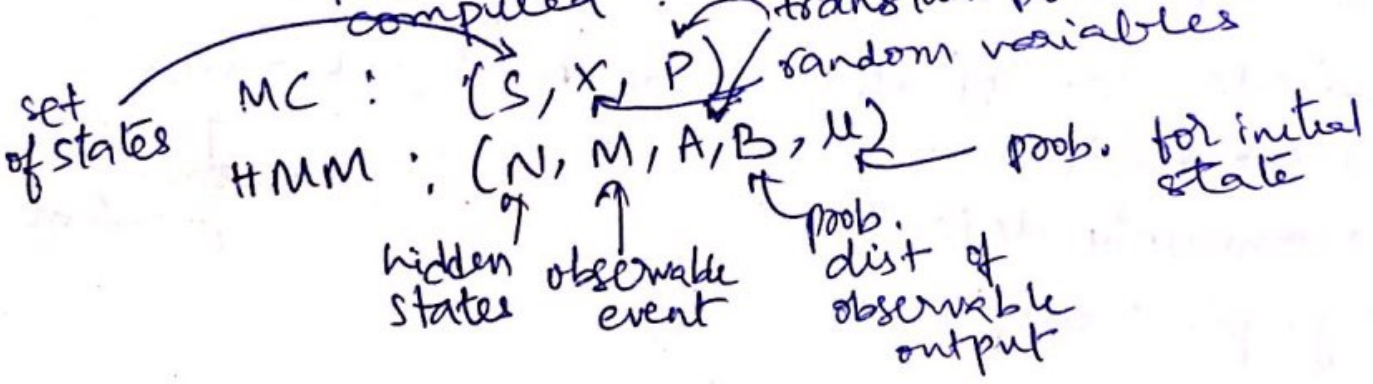
→ Normalize the data
→ Check Z-score of each datapoint
→ if $x \notin [-3, 3]$, then outlier
→ using IQR
→ ORC (outlier removal clustering → uses kmeans)

Feature selection

- using correlation
  - ↓ 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



*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

graph
- directed graph model (Bayesian network)
- undirected graph model (markov network)

dynamic graphs
- structure of graph changes with time
- node attributes can be time series

→ node classification → node attribut inference
→ link prediction → recommender systems travelling salesman problem
→ community detection
→ graph classification

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
→ betweeness centrality
→ closeness centrality
→ eigenvector centrality

gNN → utilize ~~relat~~ relationships in train NN on graphs (eg: gCN)

i. gCNN layer → $Z = \sigma(A' F W + b)$

- Z: output
- σ: activation layer
- A': graph structure (graph adjacency matrix)
- F: node features matrix
- W, b: trainable parameters

Markov chain monte Carlo:
→ class of algos for systematic random sampling from high-dimensional probability distribution
→ 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.

(Bayesians ~~vs~~ vs Frequentists)

Methods to estimate prediction interval

→ ensemble ANN
→ bayesian method
→ monte carlo method
→ bootstrap method
→ LUBE method (predicts lower & upper bounds)

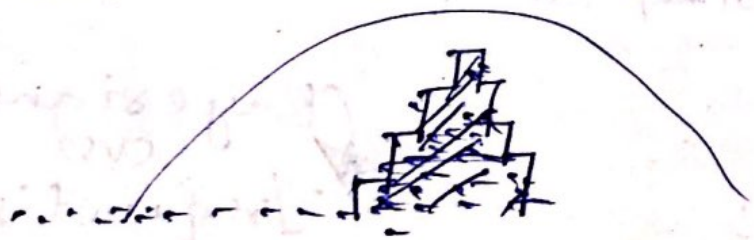
] — will be looked into deeper later.

Monte Carlo sampling
→ doesn't work well in high dimensions

① random walk
② Gibbs sampling, metropolis-hastings algos

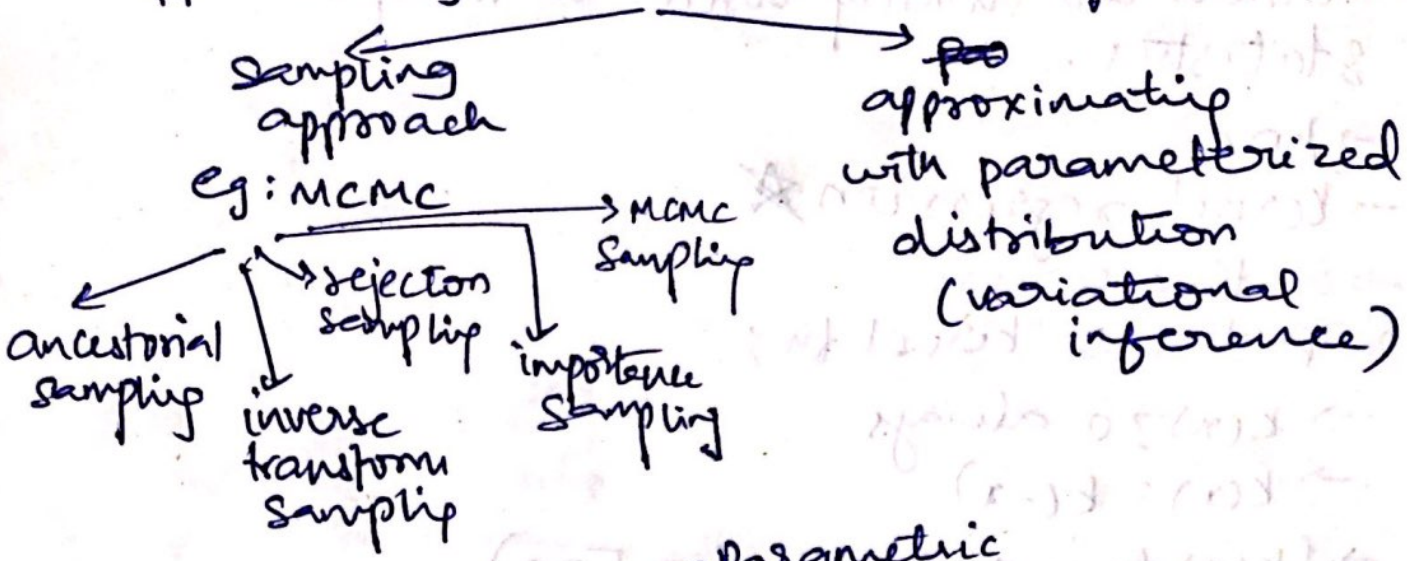
] algos based on MCMC

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 i.e posterior distribution)
prior beliefs.
→ if latest random value is better than 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
* outliers

→ Can be handled to find the perfect distribution for sample data.

transformations → log / square root / powertrans
eg: boxcox

Non parametric density estimation

↳ kernel smoothing / Kernel density estimation
giving diff weights to observations with a diff distance.

1. epanechnikov kernel : $k(x) = \frac{3}{4}(1-x^2)$ for $|x| \leq 1$
2. gaussian kernel $k(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$
3. Simply moving average kernel → $k(x) = 1/2$

Kernels are widely used in non parametric statistics.

→ KDE
→ Kernel regression ★
→ in time series

properties of kernel fn:

→ $K(x) \geq 0$ always
→ $K(x) = K(-x)$
→ $\int_x K(x)\,dx = 1$ (normalization)
→ (for many kernels $K(x) = 0 \quad x \in (-\infty, -1) \cup (1, \infty)$

* Moving average with bandwidth b:

For time series $X(1), X(2), \ldots X(n)$
where $X(i)$ is observed value a time $t = i$

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

where b is bandwidth

★ Multidimensional Scaling (MDS)

→ generating vector embedding for nodes in a graph.

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

↓
geodesic distances

used in isomap ★ dimensionality reduction.

MDS

metric (classical) (principal coordinate analysis) → model similarity/dissimilarity btw data by calculating dist btw each pair of points using geometric coordinates

non-metric → on ordinal data

multicollinearity & factor analysis ✱ (Covered will be later)

Dimensionality reduction (DR) (DR (vs) feature selection)

→ for:
1) img compression
2) remove noise
3) factor analysis →
4) take care of multicollinearity →
5) transform non-linear data into linearly seperable (Kernel PCA)

DR methods

- only keep most imp features (feature selection?)
  - → backward elimination
  - → forward selection
  - → random forests
- find combination of new features
  - Linear methods
    - → PCA
    - → FA
    - → LDA
    - → Truncated SVD
  - non-linear methods
    - ✱ Manifold learning
    - → Kernel PCA
    - → t-SNE
    - → MDS
    - → ISOMAP.

→ needs scaled data

PCA → finds a set of uncorrelated components of max variance.

LDA → finds linear comb of input features that optimizes class seperability
→ data must be normally distr
→ must contain known classes
→ max no. of comp = no. of classes − 1

"Non-linear datasets"?

Statistical methodologies for handling outliers
→ standard deviation
→ percentiles
→ z-score

Monte Carlo Simulations:
→ the crude monte carlo

$$I_{avg} = \frac{1}{b-a} \int_a^b f(x)\,dx$$

→ ancestral sampling
→ inverse transform sampling
→ rejection sampling
→ importance sampling
→ monte carlo sampling
→ MCMC sampling

Different Sampling techniques

probability sampling
→ random sampling
→ stratified sampling
→ cluster sampling
→ systematic sampling
→ multistage sampling

non-probability sampling
→ convinience sampling
→ voluntary sampling
→ snowball sampling

Linear (vs) non-linear curve.

q. why do we apply activation fn in neural networks.

Monte Carlo simulations
→ importance sampling
→ crude method
→ metropolis algorithm.
→ variational monte carlo technique

Kernel machines in machine learning

Algos capable of operating with kernels
→ kernel perceptron
→ SVMs
→ gaussian procures
→ PCA
→ canonical correlation analysis
→ ridge regression
→ spectral clustering
→ linear adaptive filters
→ etc..

Stationary distribution ~~posterior~~

$$P(A/B) = \frac{P(B/A)P(A)}{P(B)} = \frac{P(B/A)P(A)}{\sum_y P(B/A)P(A)}$$

likelihood, prior, posterior, P(B) → normalizing const

~~MCMC to estimate this~~

using a balance sheet condition,
MCMC → estimate posterior distribution without having to calculate P(B).

algos
→ metropolis hastings
→ hamiltonian monte carlo

Linear (model trees) :-

Expectation - Maximization algo & GMMs.

"latent variable" ← is inferred through math model from other variables that are observed.

latent variable models → Explain obs variables in terms of latent variable.

GMM is a generative model &

→ overcoming overfitting by density estimation.

cross validation ← → analytic criteria like Akaike info criteria (AIC) or Bayesian info criteria (BIC)

Graphical models

→ junction tree algo
→ sum-product algo
→ MCMC
→ variational inference

graph pattern

bayesian networks (directed acyclic) ← → markov networks (undirected) * need not be acyclic

factor / potential fn ϕ for markov networks.

table doesn't define prob distribution.

gives joint prob distribution on random variables

→ conditional independance.

Coditional probability dist of a graph (for each node) for bayesian network

Markov network parameter estimation → define some params that describe it's prob distribution & then use gradient descent to find param values that maximize the prob of observed data.

Bayesian network param estimation → by constructing the CPD table.

Reasoning → maximum-a-posteriori inference (MAP)

- marginal inference → $P(V=1)$ (or) $P(V=3)$ etc.
- posterior inference → $P(V_H \mid V_E = e)$
  - V_H: hidden variable
  - V_E: given evidence variable
  - given V_E, find the config that makes V_H have highest probability

Algos for solving the above 3 types of reasoning questions:

- → variable estimation
- → belief propagation
- → approximate reasoning

| exact algos
| sampling algos
| variational algos

Probabilistic graphical modelling : $P(x, y)$

→ Naive Bayes assumption : $P(y, x_1, \ldots, x_n) = P(y) \prod_{i=1}^{n} P(x_i \mid y)$

AI + RPA + IOT

Representation ← | → Learning
Inference

Generative models using probabilistic modelling
→ Sampling x from P(x) after generating P(x) or sample data.
using
→ Img denoising P(original img | noisy img) based on our prob model of what imgs look like. i.e graphical model to model the posterior distribution

Common random variables —

→ $X \sim \text{Bernoulli}(p)$ $p \in [0,1]$
Outcome of a coin flip H=1, T=0 where prob of heads is p

$$P(x) = \begin{cases} p & \text{if } x = 1 \\ 1-p & \text{if } x = 0 \end{cases}$$

→ $X \sim \text{Binomial}(n,p)$ $p \in [0,1]$
~~n is no. of hea~~
(no. of heads in n independant flips of coin with p as prob for heads)

$$P(x) = {}^{n}C_{x}\, p^{x} (1-p)^{n-x}$$

→ $X \sim$ Geometric(p), $p > 0$ (no. of flips of coin until first heads of prob p)

$$p(x) = P(1-p)^{x-1}$$

→ $X \sim \text{Poisson}(\lambda)$, $\lambda > 0$ (prob dist over non-ve int used for modelling freq of rare events) $P(x) = e^{-\lambda}\lambda^{x}/x!$

Chain rule

$$P(S_1 \cap S_2 \cdots \cap S_k) = P(S_1)\, P(S_2 \mid S_1)\, P(S_3 \mid S_2 \cap S_1) \cdots P(S_k \mid S_1 \cap S_2 \cdots \cap S_{k-1})$$

$$\text{PDF} = \frac{d}{dx}\,\text{CDF}(x)$$

→ $X \sim$ Uniform (a,b), (Equal prob density to every value btw a,b) $a<b$

$$b(x) = \begin{cases} \frac{1}{b-a}, & \text{if } x \in [a,b] \\ 0, & \text{otherwise} \end{cases}$$

→ $X \sim$ Exponential (λ), $\lambda > 0$ (decaying prob density over non-ve reals)

$$b(x) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

→ $X \sim$ Normal (μ, σ^2) (gaussian dist)

$$b(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{(x-\mu)^2}{2\sigma^2}}$$

$$\text{Cov}(X,Y) = E[(X - E(X))(Y - E(Y))]$$
$$= E[XY] - E(X)E(Y)$$

Problems based on HMM:

- Likelihood of given observed sequence (Forward algo)
- ~~Likelihood~~ of hidden seq for a given observed sequence (viterbi algo & posterior viterbi algo)
- Transition & Emission prob matrices (HMM params) training (Baum-welch & forward-backward algo) ← EM algo can be used to estimate them

Transition prop matrix : prob of transition from state a_i to a_j.

Emission prob : prob of a hidden state emitting an observation O_i.

GNN
- Recurrent GNN
- Spatial CN
- Spectral CN.

Community detection
- agglomerative methods (adding edges one by one to graph with no edges)
- divisive methods (removing edges one by one from complete graph)

→ louvain cd

"Optimizing modularity in a network is NP-Hard & have to use heuristics".

→ Suprise cd.

Etc algorithms

Node Embeddings
- Kernel-based methods,
- graph-based regularization techniques.

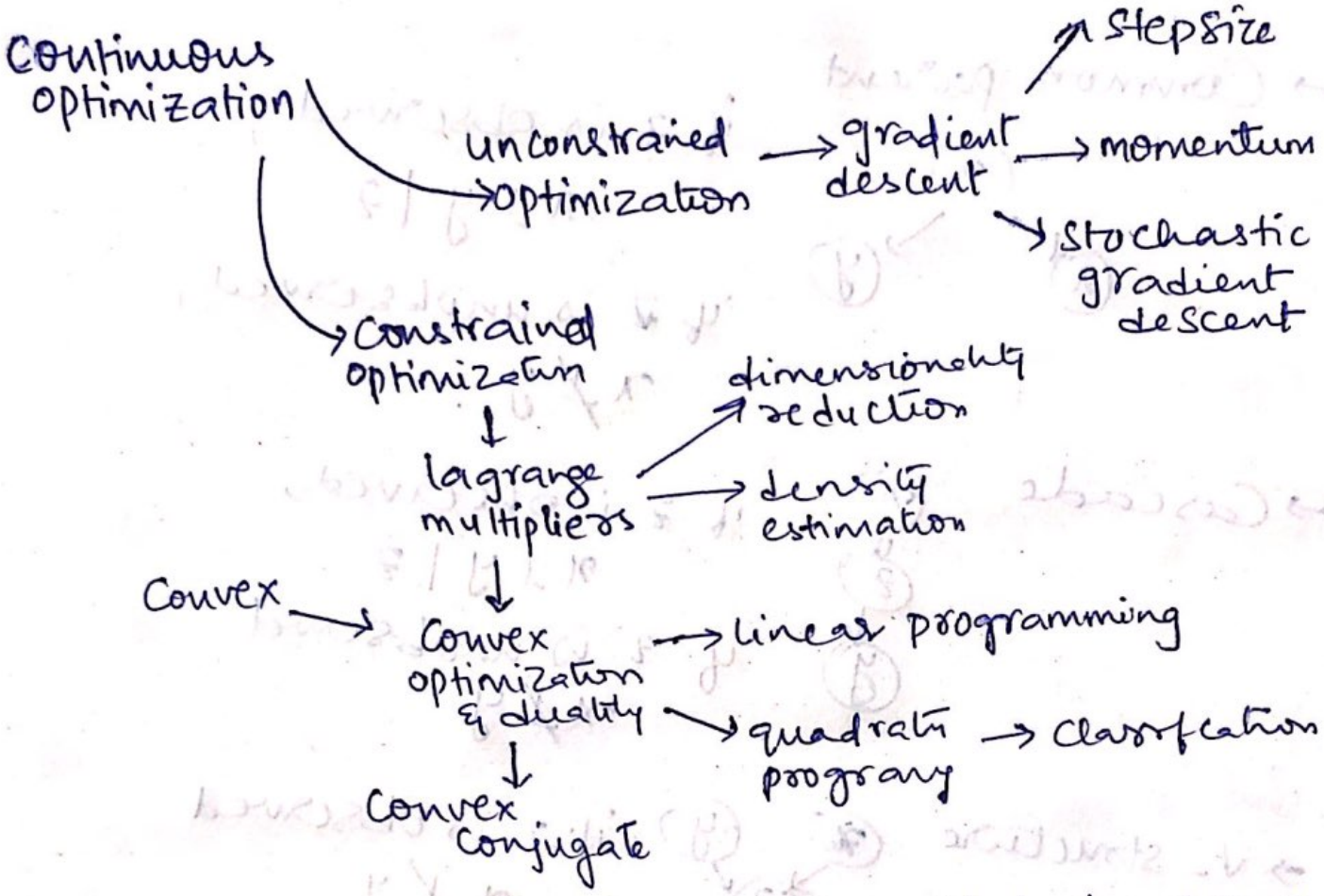
→ Density estimation with GMMs
→ Convex optimization
→ Constrained optimizatn & Lagrange multipliers
→ Empirical risk minimization
→ Regularization & Stability
→ Kernel methods, Kernels in SVM (Etc.)
→ Ranking (understn ml theory algos book)

- Online learn
- Linkage based clustering

Unsupervised learn
- discovering clusters
- discovery latent factors
- discovery graph structure
- matrix completion

→ generative models for discrete data
- bayesian
  - beta-binomial
  - dirichlet-multinomial
- Imagine ml

→ gaussian models



★ Adaptive gradient methods : update lr at each step depending on the change with last step

→ GMMs & EM algo
→ latent linear models
→ sparse linear "
→ kernels, kernel smoothy methods

Bayesian networks :

→ d-seperation notation & active path (??)

→ if $P(x, y) = P(x)P(y)$ then $x \perp y \in I(P)$

$I(P)$ denotes set of all independencies that hold for joint distr P.

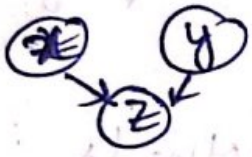
→ Common parent

if z is observed,
$x \perp y \mid z$
if z is unobserved,
$x \not\perp y$.

→ Cascade

if z is observed,
$x \perp y \mid z$
if z is unobserved
$x \not\perp y$

→ V-structure



if z is observed
$x \not\perp y$
if z is unobserved
$x \perp y$

MLE (vs) MAP (vs) Bayesian inference

max a posteriori estimation

Statistical modelling of an ex & parameter to be estimated.

MAP preferred over MLE when observed data size is small.

MAP estimation → $\underset{\theta}{\text{argmax}}\, P(\theta/D)$

where $P(\theta/D) = \frac{P(D/\theta)\,P(\theta)}{P(D)}$

"Conjugate prior distribution"

$P(\theta)$ is chosen to be conjugate prior to the $P(D/\theta)$ distribution.

eg: conj prior of binomial dist is beta dist.

MLE : $\underset{\theta}{\arg\max}\, P(D/\theta)$

MLE & MAP → point estimators

Bayesian inference → calculation of full posterior dist

$$P(\theta/D) = \frac{P(D/\theta)\,P(\theta)}{P(D)} \quad \& \quad P(D) = \int_\theta P(D \cap \theta)\, d\theta = \int_\theta P(D/\theta)\,P(\theta)\, d\theta$$

MAP → mode of posterior dist

EAP → mean of posterior dist

$$\theta_{EAP} = E(\theta/D) = \int_\theta \theta\, P(\theta/D)\, d\theta$$

generally, we use MCMC during Bayesian inference if the integral computation becomes complicated or unsolvable.