Theory

Learning From Data

Map

Paradigms:

- supervised
- unsupervised
- reinforcement
- active
- online

Theory

- VC
- bias – variance
- complexity (computation, P, NP, ...)
- bayesian

Techniques

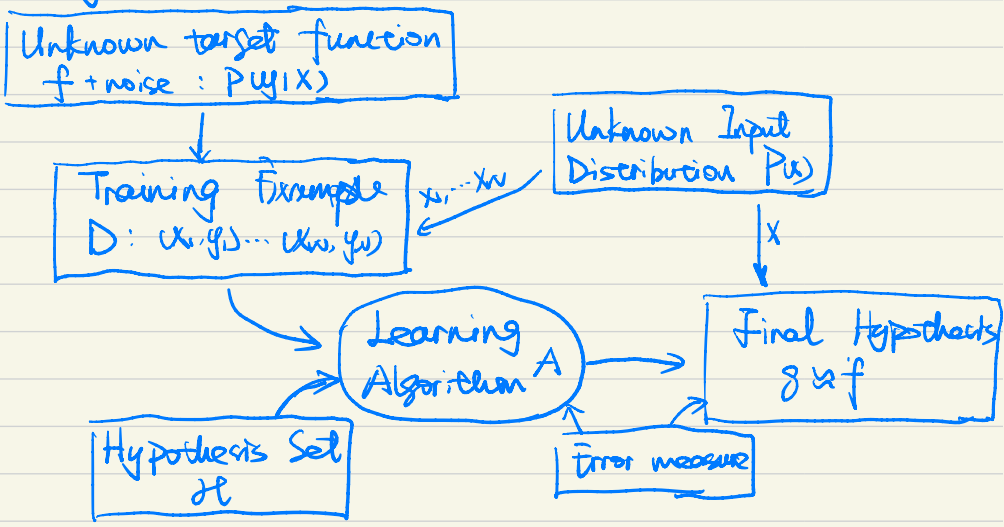
Model

- linear
- neural networks
- SVM
- nearest neighbors
- RBF
- Gaussian Process
- SVD
- graphical model

Method

- regularization
- validation
- aggregation
- input processing

Learning Problem



Learning model = hypothesis set + Learning Algorithm

Views
- Statistic: rigorous proof, idealized model in great detail
- ML: less assumption, weaker result, broader application
- Data Mining: More data analysis than prediction

Learning Feasibility

infer outside ↑ with D in a probalistic way

要了解
1. A pattern exists
2. Can't pin it down mathematically
3. We have data

Hoeffding's Inequality

Let $X_1, \cdots X_n$ be independent R.V bounded $[a_i, b_i]$ a.s.

$S_n = X_1 + \cdots + X_n$

$$\Rightarrow \begin{cases} P(S_n - E(S_n) \geq t) \leq \exp\left(-\frac{2t^2}{\sum (b_i - a_i)^2}\right) \\ P(|S_n - E(S_n)| \geq t) \leq 2\exp\left(-\frac{2t^2}{\sum (b_i - a_i)^2}\right) \end{cases}$$

In sample Error

$E_{in}(h) = \frac{1}{N}\sum_{i=1}^{N}[h(x_i) \neq f(x_i)]$

= fraction of D that f disagree with h

Out sample Error

$E_{out}(h) = P(h(x) \neq g(x))$, P is based on x

$\Downarrow \quad 0 \leq E \leq 1$

$P(|E_{in}(h) - E_{out}(h)| > \varepsilon) \leq 2e^{-2\varepsilon^2 N}$

$\Downarrow \quad \mathcal{H} = \{H_1, \cdots H_M\}$ → choose h based on E_{in}

$P(|E_{in}(h) - E_{out}(h)| > \varepsilon) \leq \sum_{h \in \mathcal{H}} P(|E_{in}(h_i) - E_{out}(h_i)| > \varepsilon)$

$(2Me^{-2\varepsilon^2 N})$ $\mathcal{H}$ complex 大 M 多大

Feasibility Question split into 2

1. Can we make sure $E_{out}(g)$ is close to $E_{in}(g)$
   Hoeffding's Inequality
2. Can we make $E_{in}(g)$ small enough
   Learning algorithm A

Error Measure

- In sample Performance over entire input space x
  $E_{in}(h) = \frac{1}{N}\sum_{i=1}^{N} e(h(x_i), f(x_i))$
- Out of sample Performance in D
  $E_{out}(h) = E_x[e(h(x), f(x))]$

In a noisy set

$P(y|x)$ is what we are looking for

$P(x)$ is the relative importance of x in gauging performance

VC Dimension

Testing:

$$P(|E_{in} - E_{out}| > \varepsilon) \le 2e^{-2\varepsilon^2 N}$$

Training:

$$P(|E_{in} - E_{out}| > \varepsilon) \le 2Me^{-2\varepsilon^2 N}$$

Effective # of Hypothesis

1. Growth Function — group h based on its behavior on D
   - A hypothesis $h: \mathcal{X} \to \{-1, 1\}$ — all input space
   - A dichotomy $h: \{x_1, \dots x_N\} \to \{-1, 1\}$
     - Number of Hypothesis $|\mathcal{H}|$ can be infinite
     - Number of dichotomy $|\mathcal{H}(x_1, \dots x_N)|$ is finite

     $$|\mathcal{H}(x_1, \dots x_N)| \le 2^N$$

   growth function count the most dichotomy of $x \in \mathcal{X}$

   $$m_{\mathcal{H}}(N) = \max_{x_1 \dots x_N \in \mathcal{X}} |\mathcal{H}(x_1, \dots x_N)| \le 2^N$$

2. Bound of Growth Function (with break point)

   Break point: If no data set of size k can be shattered by $\mathcal{H}$, then k is a break point

   Bound: If $m_{\mathcal{H}}(N) < 2^N$ for some k

   then $m_{\mathcal{H}}(N) \le \sum_{i=0}^{k-1} \binom{N}{i}$ → polynomial in N of degree k-1

   VC dim (Vapnik-Chervonenkis) of a hypothesis set $\mathcal{H}$

   $d_{VC}(\mathcal{H})$ is the largest value of N for which $m_{\mathcal{H}}(N) = 2^N$

   $$m_{\mathcal{H}}(N) \le \sum_{i=0}^{d_{VC}} \binom{N}{i}$$

   The most points that $\mathcal{H}$ can shatter

VC dimension

- $d_{vc} \geq N \iff$ there exists D of size N such that $\mathcal{H}$ shatters D
- $d_{vc} < N \iff$ No set of N points can be shattered by $\mathcal{H}$

ν degree of freedom : effective number of hypothesis

3 Replace M with $m_{\mathcal{H}}(N)$

$$P[|E_{in}(g) - E_{out}(g)| > \varepsilon] \leq 4 m_{\mathcal{H}}(N) e^{-\frac{1}{8}\varepsilon^2 N}$$

$$\Updownarrow$$

☆ **VC generalization bound : $E_{out}(g) \leq E_{in}(g) + \sqrt{\frac{8}{N} \ln \frac{4 m_{\mathcal{H}}(2N)}{\delta}}$ with $1-\delta$**

proof : estimate (Replace) $|E_{in}(g) - E_{out}(g)| > \varepsilon$

with an artificial event $|E_{in}(g) - E'_{in}(g)| > \varepsilon$

- A loose bound, but tends to be equally loose hence useful for comparing generalization performance of models
- $E_{out}(g) \leq E_{in}(g) + \Omega(N, \mathcal{H}, \delta)$

$$\Omega(N, \mathcal{H}, \delta) = \sqrt{\frac{8}{N} \ln \left(\frac{4 m_{\mathcal{H}}(2N)}{\delta}\right)}$$

$$\leq \sqrt{\frac{8}{N} \ln \left(\frac{4 (2N)^{d_{vc}} + 1}{\delta}\right)}$$

- A loose estimate of $E_{out}(g)$ as guideline
  Test sample → estimate $E_{out}(g)$
- Can extend to other Error Measure

Bias vs Variance

Approximation-Generalization tradeoff

Goal: small E_{out}, good approximation of f out of sample

complex $\mathcal{H}$ $\begin{cases} \text{better chance of approximating } f \\ \text{worse chance of generalizing out of sample} \\ \quad \text{(Have to navigate in } \mathcal{H} \text{ through } \mathcal{D} \text{ for } g\text{)} \end{cases}$

Bias vs Variance: decomposing E_{out} into:

1. How well $\mathcal{H}$ can approximate f (overall, not in current sample)
2. How well can zoom in on a good $g \in \mathcal{H}$

Apply to real value f and square error

$$E_{out}(g^{(\mathcal{D})}) = E_x[(g^{(\mathcal{D})}(x) - f(x))^2]$$

$\Downarrow$ remove dependance on $\mathcal{D}$ by expectation

$$E_{\mathcal{D}}[E_{out}(g^{(\mathcal{D})})] = E_{\mathcal{D}}[E_x[g^{(\mathcal{D})}(x) - f(x))^2]] = E_x[E_{\mathcal{D}}[g^{(\mathcal{D})}(x) - f(x))^2]]$$

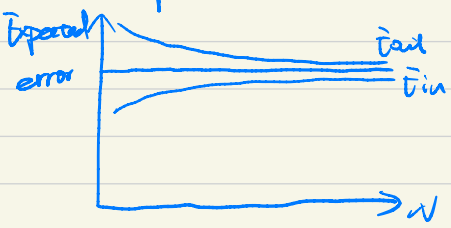
$\Downarrow\Downarrow$ average hypothesis $\bar{g}(x) = E_{\mathcal{D}}[g^{(\mathcal{D})}(x)] \approx \frac{1}{K}\sum_{i=1}^{K} g^{(\mathcal{D}_i)}(x)$ on many data set $\mathcal{D}_1 \cdots \mathcal{D}_K$ the best can get from the $\mathcal{H}$ (infinite data)

$$E_{\mathcal{D}}[(g^{(\mathcal{D})}(x) - f(x))^2] = \underbrace{E[(g^{(\mathcal{D})}(x) - \bar{g}(x))^2]}_{\text{variance}} + \underbrace{(\bar{g}(x) - f(x))^2}_{\text{bias}}$$

$$E_{\mathcal{D}}[E_{out}(g^{(\mathcal{D})})] = E_x[\text{bias}(x) + \text{var}(x)] = \text{bias} + \text{var}$$

Learning Curve

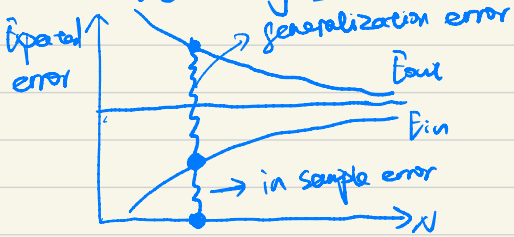
Simple Model



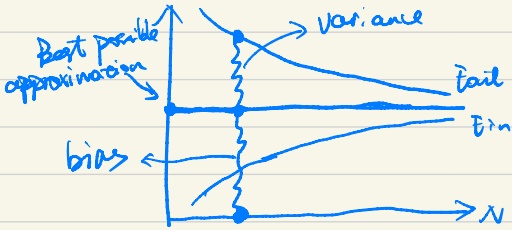
Complex Model

Match model complexity with data source not target complexity

VC analysis



bias-variance



Linear Case

$y = w^{*T} x + \text{noise}$

① Best approximation error σ^2 (= var(noise))

② Expected in sample error $\sigma^2 (1 - \frac{d+1}{N})$

③ Expected out of sample error $\sigma^2 (1 + \frac{d+1}{N})$

④ Expected generalization error $2\sigma^2 \frac{d+1}{N}$

Overfitting

Overfitting: $E_{in} \downarrow$ $E_{out} \uparrow$

fitting the data more than warranted (fitting the noise)

$$y = f(x) + \underbrace{\epsilon(x)}_{\sigma^2} \triangleq \sum_{q=0}^{Q_f} a_q x^q + \epsilon(x)$$

Overfitting:
- noise level: σ^2 + stochastic noise
- target complexity: Q_f + deterministic noise
- data set size: N -

Deterministic Noise: The part of f that $\mathcal{H}$ can't capture

Noise: something can't capture / deal with

Difference with stochastic noise.

1. depend on $\mathcal{H}$ (complex $\mathcal{H}$ → smaller)
2. fixed for a given X

Noise and bias-variance

$$E_D[(g^{(D)}(x) - f(x))^2] = E_D[(g^{(D)}(x) - \bar{g}(x))^2] + [\bar{g}(x) - f(x)]^2$$

$\Downarrow$ if $f(x)$ is noisy $y = f(x) + \epsilon(x)$, $E[\epsilon] = 0$

$$E_{D,\epsilon}[(g^{(D)}(x) - y)^2]$$

$$= E_{D,\epsilon}[(g^{(D)}(x) - \bar{g}(x))^2 + (\bar{g}(x) - f(x))^2 + (\epsilon(x))^2 + \underbrace{\text{cross term}}_{\downarrow 0 \text{ as } E[\epsilon(x)] = 0}]$$

$$\underbrace{E_{D,x}[(g^{(D)}(x) - \bar{g}(x))^2]}_{\text{var}} + \underbrace{E_x[(\bar{g}(x) - f(x))^2]}_{\text{bias}} + \underbrace{E_{\epsilon,x}[(\epsilon(x))^2]}_{\sigma^2}$$

var: impacted by both noise, capturing a model's susceptibility to being led astray by the noise

bias: deterministic noise

σ^2: stochastic noise

Not change with N

where overfit come in ← try to fit noise drag g away

Regularization

Constrain the learning algorithm to improve out-of-sample error
Necessary if model is too complicated for data amount/quality

View from VC

$E_{out}(h) \le E_{in}(h) + \Omega(\mathcal{H})$ for all $h \in \mathcal{H}$

Convert a measure $\Omega(h)$ for the complexity of an individual hypothesis
should fit better using simple h from $\mathcal{H}$

$\min_h E_{in}(h) \Rightarrow \min_h [E_{in}(h) + \Omega(h)]$

View from bias-var

sacrifice bias for smaller var

Unconstrained solution

$\min E_{in}(h) \longrightarrow w_{lin} = (X^T X)^{-1} X^T y$

Soft-order constraint / weight decay

① $\min E_{in}(h)$
$s.t \quad w^T w \le C \qquad C \uparrow \Leftrightarrow \lambda \downarrow$
$\Updownarrow$ KKT condition
② $\min E_{in}(h) + \frac{\lambda}{N} w^T w$ (augmented error) penalize large weight
$\longrightarrow w_{reg} = (X^T X + \lambda I)^{-1} X^T y$

Augmented Error

$E_{aug}(h) = E_{in}(h) + \frac{\lambda}{N} \tilde{\Omega}(h)$ regularizer
$\Uparrow$
$E_{out}(h) \le E_{in}(h) + \Omega(\mathcal{H})$

E_{aug} is a better proxy of E_{out} than E_{in}

Regularization is necessary since learning is sensitive to both stochastic and deterministic noise

Best way to constrain learning is in the direction of target

- stochastic noise : high frequency / non-smooth
- deterministic noise : non-smooth

⇓

Usually constrain learning toward smooth h hurt more to overfit noise than to hurt fit useful info

Regularizers:

1. weight decay
2. Weight elimination

$$\Omega(w) = \sum_{i,j,l} \frac{(w_{ij}^{(l)})^2}{\beta + (w_{ij}^{(l)})^2}$$

3. Early stop

choose hyper (λ, stop time) by validation

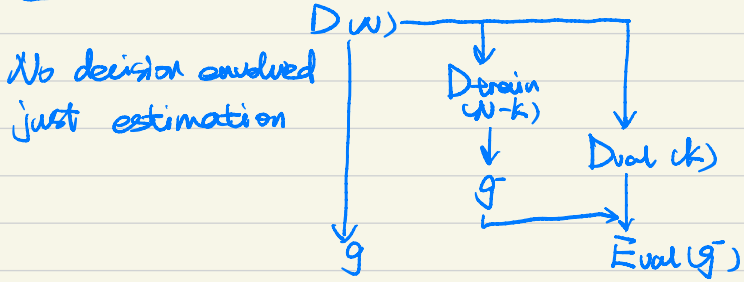
Validation

Both regularization and validation attempt to minimize E_{out}

$$E_{out} = E_{in} + \text{overfit penalty}$$

validation estimate this — regularization estimate this

Estimate E_{out} with Validation



No decision involved just estimation

$$E_{out}(g) \le E_{out}(g^-) \le E_{val}(g^-) + O\left(\frac{1}{k}\right)$$

$$E_{val}(g^-) = \frac{1}{k} \sum_{x_n \in D_{val}} e(g^-(x_n), y_n)$$

$E_{val}(E_{val}(g^-)) = E_{out}(g^-)$ ⇑ $E_{val}(e) = E_{x_n}(e) = E_{out}(g^-)$ — e depend only on x_n

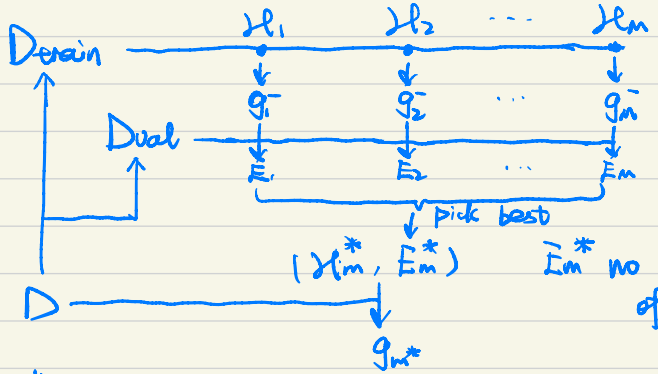
$\text{Var}(E_{val}(h)) = \frac{\sigma^2}{k}$

$E_{val}(h) = E_{out}(h) \pm O\left(\frac{1}{\sqrt{k}}\right)$

k 小: hard to exact $E_{out}(g^-)$ with $E_{val}(g^-)$

k 大: g^- is worse (less data point for training)

Model selection with validation



Decision Making involved

$(\mathcal{H}_m^*, \bar{E}_m^*)$ $\bar{E}_m^*$ no longer unbiased estimator of $\bar{E}_{out}(g_m^-)$

g_{m^*}

⇓ Hoeffding

$$\bar{E}_{out}(g_{m^*}^-) \le \bar{E}_{val}(g_{m^*}^-) + O\left(\sqrt{\frac{\ln M}{K}}\right)$$

bound is tighter than all Dtrain since $\mathcal{H}_{val} = \{g_1^- \cdots g_M^-\}$ is smaller

- Dtrain : totally contaminated
- Dval : slightly contaminated
- Dtest : totally clean

$$\bar{E}_{out}(g_{m^*}) \le \bar{E}_{out}(g_{m^*}^-) \le \bar{E}_{val}(g_{m^*}^-) + O\left(\sqrt{\frac{\ln M}{K}}\right)$$

Cross Validation

$$\bar{E}_{out}(g) \overset{\text{small } K}{\approx} \bar{E}_{out}(g^-) \overset{\text{large } K}{\approx} \bar{E}_{val}(g^-)$$

leave-one-out cross validation

① take data point (x_n, y_n) away and train with rest → g_n^-

② $e_n = \bar{E}_{val}(g_n^-) = e(g_n^-(x_n), y_n)$

③ $E_{cv} = \frac{1}{N}\sum_{n=1}^{N} e_n$

Why a good estimator?

Define $\bar{E}_{out}(N) \triangleq \bar{E}_D[\bar{E}_{out}(g)]$

$$\bar{E}_D[e_n] = \bar{E}_{D_n}\left[\bar{E}_{(x_n,y_n)}[e(g_n^-(x_n), y_n)]\right]$$
$$= \bar{E}_{D_n}[\bar{E}_{out}(g_n^-)]$$
$$= \bar{E}_{out}(N-1)$$

Cross validation can be used for model selection
For computational burden leave one out → k-fold

Theory vs Practice

Learning theory is guideline but not conclusive proof.

1. noise (stochastic + deterministic) leads to overfitting
2. Regularization helps to prevent overfitting by constraining the model, reducing the impact of the noise while still giving flexibility to fit the data.
3. Validation/CV are useful for estimating E_{out}. One important use is model selection

Kernel Method

Give 2 points x and $x' \in \mathcal{X}$

$k(x, x') = z^T z'$ for $z = \phi(x)$ in higher dimension

Compute $k(x, x')$ without ~~transforming~~ x and x'

Polynomial kernel

$\mathcal{X} = \mathbb{R}^d$, $\phi: \mathcal{X} \to \mathcal{Z}$ polynomial of order Q

$$k(x, x') = (1 + x^T x')^Q$$

$$= (1 + x_1 x_1' + x_2 x_2' + \cdots + x_d x_d')^Q$$

$$= z^T z'$$

stay here ↑ too many terms

As long as $k(x, x')$ is inner product in some $\mathcal{Z}$ space → good

Mercer's condition

$k(x, x')$ is a valid kernel iff

1. It is symmetric
2. Matrix $[k(x_i, x_j)]_{ij}$ is positive semi-definite for any $x_1, \cdots x_N$

Radial Basis Function

Each $(x_n, y_n) \in \mathcal{X}$ influences $h(x)$ based on $\|x - x_n\|$ radial

$$h(x) = \sum_{n=1}^{N} w_n \underbrace{\exp(-\gamma \|x - x_n\|^2)}_{\text{Basis Function}}$$

contribution of n on $h(x)$.

Solve for w_n

When $E_{in} = 0$: $h(x_n) = y_n$ for $n = 1, \cdots, N$

$$\sum_{m=1}^{N} w_m \exp(-\gamma \|x_n - x_m\|^2) = y_n \quad \text{for } n = 1, \cdots N$$

$$\Updownarrow$$

$$\begin{bmatrix} \exp(-\gamma\|x_1 - x_1\|^2) & \cdots & \exp(-\gamma\|x_1 - x_N\|^2) \\ \vdots & & \vdots \\ \exp(-\gamma\|x_N - x_1\|^2) & & \exp(-\gamma\|x_N - x_N\|^2) \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_N \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}$$

RBF with k centers

$N \gg k$: $u_1 \cdots u_k$ instead of $x_1 \cdots x_N$

$$h(x) = \sum_{k=1}^{k} w_k \exp(-\gamma \|x - u_k\|^2)$$

$\Downarrow$ estimate u_k with k-mean, not contaminated

$$\begin{bmatrix} \exp(-\gamma\|x_1 - u_1\|^2) & \cdots & \exp(-\gamma\|x_1 - u_k\|^2) \\ \vdots & & \vdots \\ \exp(-\gamma\|x_N - u_1\|^2) & \cdots & \exp(-\gamma\|x_N - u_k\|^2) \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_k \end{bmatrix} \approx \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}$$

$\tilde{\Phi}$ $\quad w$ $\quad y$

$$w = (\tilde{\Phi}^T \tilde{\Phi}) \tilde{\Phi}^T y$$

RBF can be derived purely on regularization

$$\sum_{n=1}^{N} (h(x_n) - y_n)^2 + \lambda \sum_{k=0}^{\infty} a_k \int_{-\infty}^{\infty} \left(\frac{d^k h}{dx^k}\right)^2 dx$$

smoothest interpolation

Learning Principles

Occam's Razor

The simplest model that fits the data is also the most plausible ☆

1. What's a simple model?
   - → Complexity of h: Minimum description length (MDL), order of polynomial
   - Complexity of $\mathcal{H}$: Entropy, VC dimension

   Link between 2 complexity

   l bits specify $f \rightarrow h$ is one of 2^l of a set $\mathcal{H}$

   exception: Looks complex but is one of few - SVM

2. Why is simpler better

   better: better out-of-sample performance

   - There're fewer simple hypotheses than complex ones
     ⇓ $m_{\mathcal{H}}(N)$
   - Less likely to fit a given data set
     ⇓ $m_{\mathcal{H}}(N) / 2^N$
   - More significance when it happens

Sampling Bias

If data is sampled in a biased way. Learning will produce a biased outcome ☆.

Data Snooping

If a data set has affected any step in the learning process its ability to assess the outcome has been compromised ☆

most common trap for practioners

When reusing data set: Trying one model after another on the same data set, you will eventually succeed

1. avoid data snooping: serict discipline
2. account for data snooping: how much data contamination

Application

High Dimmensional Data

Embedding: a relatively low-dimmensional space into which you can translate high dimmensional vectors

(嵌入)

Network Analysis

Spectral Clustering

- Spanning Tree: connect all nodes while minimizing sum of edges
- Steiner Tree: connect selected nodes while minimizing edges
  - NP-problem: need to decide which nodes to include
  - Polynomial Approximation: with spanning tree over those selected nodes

- Community Detection: detect subset of nodes that're more densly connected
- Clustering: determine subsets of points that are 'close' to each other given a similarity measure

Quality function: function that assigns a number to partition

Modularity: $Q = \frac{1}{2m} \sum_{ij} (A_{ij} - P_{ij}) \delta(c_i, c_j)$

① P_{ij}: expected edge density between i and j under null

② Erdos-Renyi: $P_{ij} = 2m/n(n-1)$

③ Configuration: $P_{ij} = k_i k_j / (2m-1)$

Modularity maximization: NP

- Louvain Method

Ideal Cluster: identify nodes that share edges more among themselves compared to outside of themselves.

$s_i = \begin{cases} 1 & : \text{group } 1 \\ -1 & : \text{group } 2 \end{cases} \Rightarrow \quad C = \min_{s \in \{-1,1\}^n} \sum_{ij} A_{ij} (1 - s_i s_j)$

n_1, n_2 size of 2 cluster s.t. $\sum s_k = n_1 - n_2$

Relaxation of clustering

① remove integer constraint for s

$$\hat{C}_1 = \min_{X \in \mathbb{R}^n} X^T L X, \quad \|X\| = 1$$

$L = D - A$ = adjacent matrix − degree matrix

⇓

spectral clustering

Time Series

Time Series Data

Time Series: a collection of realizations of distinct variable at equally spaced points in time.

R.V: $X_1 \cdots X_t \cdots X_n \quad (\Omega, P)$

$\Downarrow w$

$x_1 = X_1(w) \cdots x_t = X_t(w) \cdots x_n = X_n(w)$

① time series observations 可看作 stochastic process 下 single realization

② dependence 可帮助 predict，但也含有更多 statistical info

③ time interval 为当是 deterministic 下

Deterministic Dependence

常见体现于 mean $\mu_X(t) = E(X_t)$

① trend: $m_X(t)$

② seasonality: $S_X(t)$

Basic Stat

- Marginal Mean: $\mu_X(t) = E(X_t)$
- Marginal Variance: $Var(X_t) = E[(X_t - \mu_X(t))^2]$
- Autocovariance

$$\gamma_X(s,t) = cov(X_s, X_t) = E[(X_s - \mu_X(s))(X_t - \mu_X(t))]$$

现实世界中，R.V 只有 1 个 realization

$\Downarrow$ assumption ☆☆

stationary

✓ weak version:

$\mu_X(t) = \mu_X$, $Var(X_t) = \sigma_X^2$, $cov(X_s, X_t) = \gamma_X(s-t)$

✓ strong version:

distribution $X_{t+h}, \cdots X_{t+n+h}$ same for all t, n, h

Stochastic Dependence

$P_{X_t|X_s}$,

$\gamma_X(s,t)$ describe linear stochastic dependence

Stochastic Dependence

$P_{X_t|X_s}$

Why we need stationary

1. same distribution / stats so we can estimate from time series observations
2. Generalization of CLT / LLN when RV with large gap are kind of independent
3. Rules out trend, seasonality

$$X_t = T_t + S_t + \underbrace{Y_t}_{\text{stationary}}$$

How to get stationary

1. Remove linear regression on t → remove linear trend
2. Remove nonparametric regression on t
   i.e: kernal smoothing, polynomial, k-neighbor
   → remove trend / seasonality
3. Remove periodic regression on t → remove seasonality
4. Differenciation
   1st → remove linear trend / expanding variance
   2nd → remove polynomial trend / expanding variance
   other interval → remove seasonality
   ...
5. Transformation
   log → remove expanding variance

Empirical Estimator

$$\hat{\mu} = \bar{X}_n = \frac{1}{n}\sum_{t=1}^{n} X_t$$

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{t=1}^{n}(X_t - \hat{\mu})^2$$

$$\hat{\gamma}(h) = \frac{1}{n}\sum_{t=1}^{n-h}(X_t - \hat{\mu})(X_{t+h} - \hat{\mu}) \quad \text{for} \quad 1 \le h < n$$

$$\star\ \mathrm{var}(\hat{\mu}) = \frac{1}{n}\sigma^2 + \frac{2}{n^2}\sum_{h=1}^{n-1}(n-h)\gamma(h)$$

In order for $\mathrm{var}(\hat{\mu}) \to 0$, the second term need to decay sufficiently fast. $\Rightarrow$ estimate marginal mean from single path of realization

{ stationary
  dependence decay fast

$\Rightarrow$ can estimate expectation from mean in one realization path

Models

Autoregressive Process

AR(p): $X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p} + w_t$

- ACF decay exponentially and never 0
- stationary under conditions

Random walk

No drift: $X_t = X_{t-1} + w_t = X_0 + w_1 + \cdots + w_t$

With drift: $Y_t = \delta t + X_t$

$= \delta + Y_{t-1} + w_t$

$= \delta t + Y_0 + w_1 + \cdots + w_t$

stat:

$\mu_X(t) = E(X_t) = E(X_0)$

$\gamma_X(t) = \mathrm{Var}(X_t) = \mathrm{Var}(X_0) + t\sigma_w^2$

$\gamma_X(s,t) = \mathrm{cov}(X_s, X_t) = \mathrm{Var}(X_0) + \min(s,t)\sigma_w^2$

$\mu_Y(t) = \delta t + E(Y_0)$

$\gamma_Y(t) = \mathrm{Var}(Y_0) + t\sigma_w^2$

$\gamma_Y(s,t) = \mathrm{Var}(Y_0) + \min(s,t)\sigma_w^2$

Moving Average

MA(q) $X_t = w_t + \theta_1 w_{t-1} + \theta_2 w_{t-2} + \cdots + \theta_q w_{t-q}$

$= \sum_{h=0}^{q} \theta_h w_{t-h}$

- $\gamma(h) = \sum_{j=0}^{q-h} \theta_j \theta_{j+h} \sigma_w^2$ for $0 \le h \le q$ $\Rightarrow$ stationary

ARMA / ARIMA

ARMA(p, q) $X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p}$

$+ w_t + \theta_1 w_{t-1} + \theta_2 w_{t-2} + \theta_q w_{t-q}$

ARIMA: ARMA with differencing

Fit TS Regression

Overview

1. transform to stationary
   - log-transform
   - remove trend / seasonality
   - differentiate successively
2. check for white noise (ACF)
3. If stationary
   ① finite lag ⇒ MA(q)
      - compute ACF to get order
      - estimate with ML
      - check residual
   ② Otherwise ⇒ AR(p)
      - compute PACF to get order
      - estimate ϕ_k and residual variance $\hat{\sigma_w}$ via Yule-Walker
      - check residual
   ③ ARMA(p, q)
      - attempt to fit AR and compute residual
      - attempt to fit MA to residual
      - fit ARMA with p, q from previous steps
      - check residual

	ACF	PACF
AR(p)	decays	0 after p
MA(q)	0 after q	decays
ARMA(p,q)	decays	decays

AR(p) Parameter Estimation

AR(p): $X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p} + w_t$

- p+1 parameters to estimate: $\phi_1, \cdots \phi_p, \sigma_w^2$
- If X_t stationary $\Rightarrow$ estimate with method of moment

$$(\gamma_X(0), \gamma_X(1), \cdots, \gamma_X(p)) = f(\phi_1, \phi_2, \cdots \phi_p, \sigma_w^2)$$

$$\Downarrow$$

Yule-Walker equation

p+1 unknow and p+1 equations

$$(\hat{\phi}_1, \hat{\phi}_2, \cdots \hat{\phi}_p, \hat{\sigma}_w^2) = f^{-1}(\hat{\gamma}_X(0), \cdots \hat{\gamma}_X(p))$$

$$\Downarrow$$

continuous mapping
Delta theorem

consistent estimator

Yule-Walker:

$$\begin{cases} E(X_t X_t) = E[\phi_1 X_t X_{t-1} + \phi_2 X_t X_{t-2} + \cdots + \phi_p X_t X_{t-p} + X_t w_t] \\ E(X_{t-1} X_t) = E[\phi_1 X_{t-1} X_{t-1} + \phi_2 X_{t-1} X_{t-2} + \cdots + \phi_p X_{t-1} X_{t-p} + X_{t-1} w_t] \\ \vdots \\ E(X_{t-p} X_t) = E[\phi_1 X_{t-p} X_{t-1} + \phi_2 X_{t-p} X_{t-2} + \cdots + \phi_p X_{t-p} X_{t-p} + X_{t-p} w_t] \end{cases}$$

$$\Updownarrow$$

$$\begin{cases} \gamma_X(0) = \phi_1 \gamma_X(1) + \phi_2 \gamma_X(2) + \cdots + \phi_p \gamma_X(p) + \sigma_w^2 \\ \gamma_X(1) = \phi_1 \gamma_X(0) + \phi_2 \gamma_X(1) + \cdots + \phi_p \gamma_X(p-1) \\ \vdots \\ \gamma_X(p) = \phi_1 \gamma_X(p-1) + \phi_2 \gamma_X(p-2) + \cdots + \phi_p \gamma_X(0) \end{cases}$$

Time Series Analysis

Partial Correlation

- Partial correlation of X, Y given Z
  - regress X, Y on Z
  - $P_{XY|Z} = \text{corr}(X - \hat{X}, Y - \hat{Y})$
- Partial correlation of X_t, X_{t-h} for stationary series
  - ACF: $\rho_X(h) = \text{corr}(X_h, X_0) = \frac{E[(X_h - E(X_0))(X_0 - E(X_0))]}{\text{Var}(X_0)}$
  - PACF: $\alpha_X(h) = \text{corr}(X_h - \hat{X}_h^{\text{lin } h-1}, X_0 - \hat{X}_0^{\text{lin } h-1})$
    linear projection on $X_1, \cdots X_{h-1}$
  - Frisch-Waugh-Lovell Theorem

    $X_t = \phi_1 X_{t-1} + \cdots + \phi_h X_{t-h} + \hat{X}_t$ is best linear prediction

    $\Downarrow$

    $\alpha_X(h) = \phi_h$

Information Criterion

$AIC(k) = \underbrace{-2 \log\text{-likelihood}}_{\text{model fit}} + \underbrace{2k}_{\text{complexity penalty}}$

$BIC(k) = -2 \log\text{-likelihood} + k \ln(n)$

$BIC(k) > AIC(k)$

Gaussian Process

Gaussian Process

N-dim multivariate Gaussian

joint:

$$X = \begin{bmatrix} X_1 & X_1 \in \mathbb{R}^d \\ X_2 & X_2 \in \mathbb{R}^{N-d} \end{bmatrix}$$

$$\mu = \begin{bmatrix} \mu_1 & \mu_1 \in \mathbb{R}^d \\ \mu_2 & \mu_2 \in \mathbb{R}^{N-d} \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} \Sigma_{11} \in \mathbb{R}^{d \times d} & \Sigma_{21} \in \mathbb{R}^{d \times (N-d)} \\ \Sigma_{21} \in \mathbb{R}^{(N-d) \times d} & \Sigma_{22} \in \mathbb{R}^{(N-d) \times (N-d)} \end{bmatrix}$$

conditional:

$$\mu_{X_1|X_2} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(X_2 - \mu_2)$$

$$\Sigma_{X_1|X_2} = \Sigma_{11} - \Sigma_{21}\Sigma_{22}^{-1}\Sigma_{21}$$

$$X_1|X_2 \sim \mathcal{N}(\mu_{X_1|X_2}, \Sigma_{X_1|X_2})$$

Spacial Covariance

observed value: $y_1, \cdots y_N$

location: $x_1, \cdots x_N$

RBF: $\text{cov}(Y_i, Y_j) = k(x_i, x_j) = \exp\left[-\frac{\|x_i - x_j\|^2}{2\nu}\right]$

$$\Sigma_N = \begin{bmatrix} \text{cov}(Y_1, Y_1) & \cdots & \text{cov}(Y_1, Y_N) \\ \vdots & & \vdots \\ \text{cov}(Y_N, Y_1) & & \text{cov}(Y_N, Y_N) \end{bmatrix} = \begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_N) \\ \vdots & & \vdots \\ k(x_N, x_1) & \cdots & k(x_N, x_N) \end{bmatrix}$$

$$k_*^T = [k(x_*, x_1), \cdots k(x_*, x_N)]$$

conditional:

$$\mu_{*|1:N} = \mu_* + k_*^T K_N^{-1}(y_{1:N} - \mu_{1:N})$$

$$\sigma^2_{*|1:N} = \sigma^2_* - k_*^T K_N^{-1} k_*$$

predict:

$$\hat{y}_* = \hat{f}(x_*) = \underbrace{\mu_*}_{\text{assume } 0} + \underset{1\times N}{k_*^T} \underset{N\times N}{K_N^{-1}} \underset{N\times 1}{y_{1:N}}$$

$=$

① $\sum_1^N w_i^* y_i$ weighted average of $y_{1:N}$

② $\sum_1^N k(x_*, x_i)\alpha_i$ linear combination of N non-linear features, same α for all prediction

Gaussian Process

a collection of random variables, any finite number of which is Gaussian

$$\begin{cases} m(x) = \mu_x = E[f(x)] \\ k(x, x') = \text{cov}(f(x), f(x')) \end{cases}$$

Measurement Noise

$$y_i = f(x_i) + \epsilon_i, \quad \epsilon_i \sim N(0, \tilde{v})$$

$$\Rightarrow \begin{cases} \mu_{*|1:N} = \mu_* + k_*^T (K_N + \tilde{v} I)^{-1} y_{1:N} \\ \Sigma_{*|1:N} = k(x_*, x_*) - k_*^T (K_N + \tilde{v} I)^{-1} k_* \end{cases}$$

adjust less due to noise

Entropy

Quantify uncertainty

$$H(Y) = \sum_Y P(y) \log \frac{1}{P(y)} = -\sum_Y P(y) \log P(y)$$

$\log \frac{1}{P(y)}$: measure of surprise

$H(Y)$: expected surprise

- Joint Entropy:

$$H(Y, Z) = -\sum_{Y,Z} P(y, z) \log P(y, z) \leq H(Y) + H(Z)$$

- Conditional Entropy:

$$H(Y|Z) = H(Y, Z) - H(Z)$$

- Mutual Information

$$I(Y, Z) = H(Y) - H(Y|Z) = \sum_{y,z} P(y, z) \log \frac{P(y, z)}{P(y) P(z)}$$

$\frac{P(y,z)}{P(y)P(z)}$: distance from independance → if independent