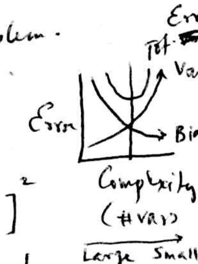
VAR. SELECTION

more X ⇒ ↓ Bias ↑ Var
few X ⇒ ↑ Bias ↓ Var.
too few X ⇒ ↑ Bias ↑ Var.

Fix:
→ High Dimensionality
→ Multicollin
→ Pred (vs) Explaination
↳ Dont include vars for pred! Want small models!

- A kind of "Data mining". Unsolved problem.
- Nothing special about selected vars
- Bias: underfitting, Var: Overfitting (future resp)



- Measure Pred Risk $R(S) = \frac{1}{n}\sum[\hat{Y}_i(S) - Y_i^*]^2$ (Sub-model; Future obs!)

Don't have future, so $R_{tr}(S) = \frac{1}{n}\sum[\hat{Y}_i(S) - Y_i]$ (Training) — use again! Snooping! Upward bias, select larger models

So penalize: $R_{tr}(S)$ + Penalty (Complexity) (C_p) — Higher is worse!
Complex Model ⇒ DOWN, UP ☺

① Mallows C_p: $\frac{2|S|\hat{\sigma}^2}{n}$ — # of sub-model preds; Est Var of FULL MODEL | ONLY IF NO CONTROL VARS!!

② AIC $= \frac{2|S|\sigma^2}{n}$ — True var (replace with full or sub-model in R)

③ BIC $= \frac{|S|\sigma^2(\log(n))}{n}$ — Same as AIC, penalize by $\left[\frac{\log(n)}{2}\right]$
└ PREFERRED, PENALIZE LARG MODELS

④ LOOCV $R_{cv}(S) \approx R_{tr}(S) + \frac{2|S|\hat{\sigma}^2(S)}{n}$ — SUB-MODEL! ∴ penalize less than Mcp!

- For GLM, use AIC, BIC (∵ "function of lack-valued function")

VSE Heuristics if p is too large: Regularized Regression (2^p submodels)
- Greedy: Settle on biggest or smallest jump in sel. process
- FOR ≠ BACK (maybe), not optimal model
  └ PREFER! Smaller models. BACK: Expensive, no-go if $p > n$; larger model for larger p

$R(S) = \text{Bias}^2[\hat{Y}_i(S)] + V[Y_i^*] + V[\hat{Y}_i(S)]$ — Overfitting
(Bias: underfitting; $V[Y_i^*]$: irreducible! Don't know!)

MSE! Can find model w/ lower MSE than full model
adding bias reduces MSE! Often...

Minimize OLS + λ(penalty)   λ balances — lack of fit; Model complexity

penalty: L0: # of non-zero βs ... Search all submodels. $\|\beta\|_0 = \#\{j : \beta_j \neq 0\}$

L_1 (LASSO) $\|\beta\|_1 = \sum|\beta_j|$ — Rewards Sparsity; Shits on Multicollin; Forces zeroes | NO CLOSED FORM!

L_2 (RIDGE) $\|\beta\|_2 = \sum\beta_j^2$ — Deal with MultiC; Comp. easy than Lasso; Good for $p > n$ | CLOSED FORM

SHRINKAGE: LASSO, RIDGE (cannot go to 0)
VAR SELECTION: LASSO (can go to 0)

Elastic Net is Both: Ridge dominates if Multicoll

- λ determined by k-fold CV (LOOCV is k = n)
- MUST scale predictors! Then interpret in terms of originals
- Less efficient than OLS (increase bias, decrease var.)
- Must redo OLS with results!
- Eval after selection — MSE (normal Reg); Classification Err (Logistic); S. of Sq. Deviances (Poisson)

L_1: Sparse Model L_2: ① Remove limit on # of vars ② Encourage group effect ③ Stabilizes L_1 path

LOGISTIC REGRESSION

ASSUME: ① Linearity: g is a linear combo of preds $g[E(Y|x)] = \beta_0 + \beta_1 \cdots \beta_p$ ② Independence: Response vars $Y_1 \cdots Y_n$ are indep. Random Vars ③ Logit function

NO ERROR TERM! PROBABILITY of SUCCESS! NOTHING TO DO w/ RESP!

$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \cdots \beta_p X_p$ or $p = \frac{e^{(-)}}{1+e^{(-)}}$ (logodds)
e^{β}: ∀ unit increase, LOG ODDS RATIO!!

- params to estimate are $\beta_0 \cdots \beta_p$ (no variance!)

Done using MLE — NO CLOSED FORM! — log(likelihood) is non-linear!
└ Good for LARGE sample sizes, else Type I Error ↑
└ Sampling dist of $\hat{\beta}$ is $\sim N(\beta, V)$ — no closed form, need software

HYP. TEST: Use Wald test (Z-test really ∵ N.D, large samples)
No t-test!

P(Type I Error) > Sign. Level for small samples...

→ ALL APPROXIMATE!! NOT GOF TESTS!

Deviance $= \log(\mathcal{L}_{reduced}) - \log(\mathcal{L}_{full})$ χ^2_p dof ($\beta_0 \cdots \beta_p x_p$ model)
[$\text{logit}(p) = \beta_0$ model]

→ Overall reg. test — H_0: Adding anything has no effect
→ Subset test — H_0: Adding "q" things has no effect (χ^2_q dof)

$CI = \hat{\beta}_j \pm Z_{\alpha/2}\sqrt{V(\beta_j)}$

LOGIT in Classification:

Classifier is $h(x_1 \ldots x_p) = \begin{cases} 1 & \text{if } \hat{p} > r \\ 0 & \end{cases}$ → Eg. 0.5

Classif. Error $= 1 - Pr(Y = h)$ DOWNWARD BIAS ∵ snooping! (Risk is upward)
Training
↳ fix w/ Cross Valid — Random subsample (Expensive, no adv over k-fold); k-fold; LOOCV
↓ Bias but ↑ Var ☹

GOF Testing: All about Residual Analysis.
Can ONLY do w/ Binary data w/ Replications! (Bernoulli vs Binomial dist)

Residuals: ① Pearson: $r_i = \frac{Y_i - n_i\hat{p}_i}{\sqrt{n_i\hat{p}_i(1-\hat{p}_i)}}$

② Deviances! ☹ $d_i = \text{Sign}(Y_i - \hat{Y}_i)\sqrt{\cdots}$ — Y_i is "saturated model" just observations! $\hat{Y}_i$ is fitted model.

- Both Pearson and Dev. have $\sim N(0,1)$ St. Norm Dist iff Assumptions hold! (Check w/ hist., qq plot)

HYP TEST: [H_0: Model fits Data] WANT BIG p-value! ONLY TIME we want this!

test stat is $D = \sum d_i^2 \sim \chi^2_{n-p-1}$

- Subset / Overall testing is about PRED POWER — Model is good at this even if does not meet assumptions!
- GOF is all about testing ↗

- Fixing Problems: ① Add Pred vars ② Transform vars ③ Outliers? ④ Maybe not appropriate for Binomial! — Corr among response Y; Heterog in Successes has not been modeled

OVERDISPERSION is caused. It means that "The variability of the prob. estimators is larger than would be implied by a Binomial Rand. Var".

To fix, maybe another s-function: ① Probit: Inverse of N.D CDF
└ Tight shape the...
② Complementary log-log
└ Good for skewed! long tails!

But Logit is still best. ① It is "canonical" ⇒ The $\hat{\beta}_p$ estimates are "fully efficient". ② Easy to interpret coeffs in terms of log-odds, cannot do with others.

GLM: $g[E(Y|X)] = \beta_0 + \beta_1 \cdots \beta_p X_p$

Y_n is a Random Var from an Exp. Dist. Family ∀ { Normal, Poisson, Bern, Bino, Gamm.

"Model some g as the Expectation of Y as a linear combo of X": $E(Y|X) = g^{-1}(\text{——})$

g is link, g^{-1} is inv. link

DIST	LINK	REG
Norm	$g(m) = m$	$m = X^T\beta$
Poisson	$g(m) = \log(m)$	$m = e^{X^T\beta}$
Bern/Bino	$g(m) = \log\frac{m}{1-m}$ (logit)	$m = \frac{e^{X^T\beta}}{1+e^{X^T\beta}}$
λ	$g(m) = 1/m$	$m = \frac{1}{e^{X^T\beta}}$

POISSON

$\log[E(Y|x)]$

(NOT) $E[\log(Y)|x]$ ← Normal!

→ For ratio and counts, if use st. reg w/ log transform and not poisson, VIOLATE CONSI VAR assumption!

→ If you must, and only for large counts, use "Variance stabilizer" $\sqrt{\mu + \frac{3}{8}}$!!

E = V for Poisson....

PDF: $P(Y=y) = \frac{e^{-\lambda}\lambda^y}{y!}$

$E(Y) = V(Y) = \lambda$

Model: $\log(\lambda) = \beta_0 \cdots \beta_p X_p$ ("log rate") → when interpreting, RATIO OF RESP RATES

- MLE, no closed form, all approxes....
- Interpret e^β: ∀ unit inc in β, the log ratio increases by β.

  Eg: "log of Expected blah" will increase by β

  "The rate ratio of blah" will inc. by e^β

MULTICOLL: Linear dep. in design matrix X. Prob. is that X^TX must be invertible for OLS to work. MC prevents this ⇒ SE of coeffs and SE of Resp are ∞ (theoretically)

→ R^2 DOES NOT CHANGE! LOL! → $\hat{\beta}_p$ is unstable w.r.t. to resp.

→ SE is artificially large → $\hat{\beta}$ has wider CI's

→ $\hat{\beta}$ less likely to be evaled as stat sig → F-stat is sig. but t-stats are not.

① Pearson's coeff or ② VIF: "Prop. increase in $Var(\hat{\beta})$ to if it were uncorrelated" (to ones more β's)

$VIF_j = \frac{1}{1-R_j^2}$ } Just Regressing against other $\hat{\beta}$s ☺

OK if $VIF_j < Max\left(10, \frac{1}{1-R^2_{model}}\right)$ or 1 (lowest)

MLR, t-Stat $= \frac{\hat{\beta}-c}{\sqrt{\hat{\sigma}^2/SXX}} = \frac{\hat{\beta}-c}{\sqrt{MSE/SXX}}$

Larger t-stat ⇒ Smaller p-value!

- Prediction outside model scope (new setting) is EXTRAPOLATION.
- Prediction, Modeling, Hyp Tests (Association) } MLR Goals

ANOVA { Bet: $k-1$; Within: $\frac{N-k}{N-1}$ } for F-stat | "

HOLDING ALL OTHERS FIXED!

Logistic, Poisson: "Null Deviance": $n-1$

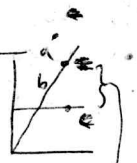
"Residual Devian": $n-p-1$

Wald test: z-stat $= \frac{\hat{\beta}}{Se(\hat{\beta})}$

Reject H_0 if $|z\text{-stat}|$ is too large ⇒ $\hat{\beta}$ is Sig.

H_0 $\hat{\beta} \le 0$, H_A $\hat{\beta} > 0$ } test for sig +ve

p-val $= P_t(Z > z\text{-val})$



a = SSE
b = SSR
"Explained by Regression"
$R^2 = \frac{SSR}{SST}$

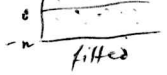
- ANOVA est. simultaneous conf Ints.
- R^2 is Coeff of Determination $= 1 - \frac{SSE}{SST} = \frac{SSR}{SST}$
- F stat $= \frac{MSR}{MSE} = \frac{SSR/p}{SSE/n-p-1}$, Adj $R^2 = \left[1-(n-1)\left(\frac{1-R^2}{n-p-1}\right)\right]$
- Cooks measures how much param change when i^{th} obs. is removed: $\frac{4}{n}$ } if > than this, investigate
- F-stat $= \frac{\text{Bet. Grp}}{\text{Within Grp}} = \frac{\text{Explained Var}}{\text{UnExplained Var}} = \frac{SSR/(k-1)}{SSE/n-k}$ } FOR ANOVA!

Causality → ISOLATION. ASSOCIATION is no ISOLATION

ASSUMPTIONS | Simple Linear LES ANALYSIS:

- Linearity (Mean zero) $E(\epsilon_i) = 0$
- Const Var $Var(\epsilon_i) = \sigma^2$
- Indep ϵ_i is iid
- Normality: $\epsilon_i \sim N(0, \sigma^2)$

SCATTERPLOT { Linearity, Cons Var, Indep }



Cons Var, Indep (Uncorrelated NOT Independent errors!)

Normality { symmetric, unimodal, no gap }

MLR: Std Res vs Var → LIN; Std Res vs Fit → CV, IND; QQ, Hist → NORM

- $\lambda = 1$ nope
- $\lambda = 0$ log(Y)
- $\lambda = -1$ $1/Y$
- $\lambda = Y_2$ $\sqrt{Y}$

USE RESID NOT RESP

REPAIR: Norm, Cons Var } Transform Y (resp) BoxCox

Linearity } Transform X

TRUE	EST
$E(\epsilon_i) = 0$	$E(\hat{\epsilon}_i) = 0$
$Var(\epsilon_i) = \sigma^2$	$Var(\hat{\epsilon}_i) = \sigma^2(1-h_{ii})$

MATRIX FORM

$E(\epsilon) = 0$	$E(\hat{\epsilon}) = 0$
$Var(\epsilon) = \sigma^2 I$	$V(\hat{\epsilon}) = \sigma^2(I-H)$

(H: hat matrix)

Est. Residuals in MLR do not have Cons Var! (Divide by Stddev and standardize)

$r_i = \frac{\hat{\epsilon}_i}{\sqrt{\sigma^2(1-h_{ii})}}$

(correlation coeff $= \rho = \sqrt{R^2}$)

Simple: one QUANT

ANOVA: one or more QUAL var

MLR: Everything.

ANOVA: pooled Var est has $(N-k)$ χ^2 dof.

MLR: $\hat{\sigma}^2 \times MSE = \frac{\hat{\epsilon}^T\hat{\epsilon}}{n-p-1} \sim \chi^2_{n-p-1}$

$MAE = (1/n)\Sigma|Y-\hat{Y}|$ } these less robust to outliers

$MSE = (1/n)\Sigma(Y-\hat{Y})^2$

$MAPE = \left(\frac{100}{n}\right)\Sigma\left(\frac{Y-\hat{Y}}{Y}\right)$

Precision Measure $\rho = \frac{\Sigma(Y-\hat{Y})^2}{\Sigma(Y-\bar{Y})^2}$

Variable Selection

① Avoid overfitting: random effects are given a chance to rear their heads ☹

② Simpler Models: Easy to explain and less chance of including insignificant factors

③ Can be illegal to use certain factors. Remember to remove other factors that correlate strongly (eg. UT and Mormon)

HOW: Greedy Algos: Consider only one thing at a time and do not consider future options.

① Forward Selection, ② Backward Elimination, ③ Stepwise regression (Combine ④)

Start w/ some or all factors → add/remove gradually

remove factors w/ $p > 0.05$ ← fit model ← "Is factor good enough?" ($p \geq 0.15$ or $p \leq 0.15$; back / forw)

Filters: p-value, R^2, AIC, BIC

Global Algos: * Need to scale data! This is Optimization!

① LASSO: $\sum_{i=1}^{n} |a_i| \leq T$ → "budget" → pick many diff. values

② Elastic Net: $\lambda \sum |a_i| + (1-\lambda) \sum a_i^2 \leq T$

③ Ridge Reg: $\sum a_i^2 \leq T$ (does not do var. selection!)

Comparison: ① Greedy are good for initial analysis but fit more random effects. [V] hi corr [V] } will just pick one if two factors hi corr

Lasso: Good for var. selection.

Ridge: Good for prediction [ ][ ] ← will push closer, so [ ] power is lost

Pred. error = f(bias, variance) ← Tradeoff bias for variance for better model.

DOE

Data is not/never available. Compare two factors by controlling for Blocking factor that creates variance.

① A/B Testing: Compare two alternatives. * Run hyp. tests after every datapoint collection! (Higher order possible)

② Factorial Design: Compare combos. Full and fractional (Subset of factors)

Balanced Design: a) Each choice same # of times b) Each pair of choices same # of times.

- Independent factors? Use Fractional w/ Regression.
- Exploration ("Gather Data") vs Exploitation ("Use it")

"How quickly are you able to deliver value?"

③ Multi-Armed Bandit: Compare alternatives. Do both explore and Exploit. Old Ladies in Vegas

→ Alternatives { A/B; Higher order A/B; MAB } Combos — fractional { Full; Frac.

PROB-BASED MODELS

iid!!! → Data cannot "clump" eg. temp over a year

① Bernoulli: "Will x happen or not?"

PMF: $P(X=1) = p$, $P(X=0) = (1-p)$ → Bern. Rand. Variable

② Binomial: "x successes out of n Bernoulli trials"

PMF: $P(X=x) = \binom{n}{x} p^x (1-p)^{n-x}$ * for large n → Normal Dist ♡

③ Geometric: "How many trials until....?"

PMF: $P(X=x) = (1-p)^x p$ or $p^x (1-p)$ * Careful defining "success"

* Check iid! If Geometric, is iid ☺

④ Poisson: "x things happening w/ AVG happen rate λ" (TIME)

PMF $= \frac{\lambda^x e^{-\lambda}}{x!} = f_x(x)$

⑤ Exponential: "If λ follows dist, it is Poisson!" "If Poisson, λ is exp. dist"

PMF $= \lambda e^{-\lambda x}$

Special case of Weibull.

⑥ Weibull: "Time until something happens" AND "Time bet. failures"

PMF: $\frac{k}{\lambda} \left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k}$

$k < 1 \Rightarrow$ "Worst things fail first" fail rate DEC w/ time

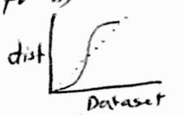
$k > 1 \Rightarrow$ opposite of ↑ eg. Tires, bulbs. Failure rate INC w/ time

$k = 1 \Rightarrow$ fail rate constant ⇒ Exponential.

- Geometric: "# of flips until bulb goes out"
- Weibull: "amt of time until bulb goes out"

Q-Q Plot: ① Compare two dataset percentiles ② Map dataset to a prob. dist. Why? a) Tests don't give visual info b) We might be interested in just the extreme values!

→ 10-percentile "Bigger than 10% of data as smaller than rest"



⇓ "More points at tails than dist"

Queueing

Queueing: Needs something resembling a queue.

Memoryless: Next state only depends on current. Follows Exp, Poisson.

↳ if not, use Weibull (eg. bulb, tires)

Queueing theory, Kendall notation is standard ⇒ "M/M/1" (Markov)

- A: Arrival Dist
- S: Serv. Dist
- C: # of Servers
- k: Queue Size
- N: Pop. Size
- D: Queue discipline (FIFO, LIFO, SIRO)

Simulation: When things dont follow prob. dist.

↳ Deterministic

↳ Stochastic < Continuous time (PDEs are used); Discrete event Stochastic Sims (DESS)

DESS: ① Build model w/ ⓐ Entities (people, bags) ⓑ Modules (queue, storage) ⓒ Actions ⓓ Resources (workers, guards) ⓔ Decision Points ② Replication: Run many times to get dist

③ Validate w/ real data * Careful can have same μ and diff σ^2

④ Use: Simulate w/ diff options. Use same randomness!

Markov-Chain Models

* Assume system is memoryless! (Based on system states) * No cycles! * Every state reachable from other.

π → State vector

$P = \{P_{ij}\}$ → Trans. Matrix w/ state trans prob p_{ij}. Solve for π^* s. $\pi^* P = \pi^*$ ← "Steady State" ← trans. dont change.

Eigen vector w/ Eigenvalue of 1 ♡ Eg. PageRank by Google™

Missing Data & Imputation

- Data wrong or missing
- Some data more than others (bias!)

Non-Estimation methods: ① Throw away (not putting in error) ② Cat. vars w/ interaction terms. (like getting two models) (also doubling vars)

Estimation Method: ① Simple: Data < numeric: Mean, Median; Categ: Mode

Easy but bias possible to under/over estimate (eg. rich ppl)

② Predictive — Complex — Use data twice — Not capturing variability ☹

Eg. w/ Regression.

③ Perturbation: Add rand. to each imputed val from some dist.

DISTURB? AVG ☺, VAR !! | DONT DISTURB AVG !!, VAR ☺

* Don't impute more than 5% of data.

MICE: Impute many factors

Errors: (Impute Error) + (Perturb Error) + (Model error)

But all data has error anyway!

ADV MODELS

What if no good models? Unknown dist? Non-param tests! — not much data — Need median — Unknown dist

Hyp. tests assume this but what if none?

① McNemar's Test (paired, ~~matched~~): Match data in resp. Throw away stuff that matches. Use binomial to compare discordant.

② Wilcoxon Signed Rank (single, paired, ~~matched~~): Assume response is cont., symmetric. "Is median different than m?" ↳ measures how !!! they are w/o distribution!

③ Mann-Whitney Test (unpaired)

④ Bayesian Models: Very little data. Est. distribution.

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} = \frac{P(B|A) \cdot P(A)}{P(B|A) \cdot P(A) + P(A^c) \cdot P(B|A^c)}$$

Combine w/ expert opinion to get

$P(A)$ = "Prior dist"

$P(A|B)$ = "Posterior" (new dist)

Single observ. can be related to larger!

Network Models: "Find highly interconnected subpopulations"

Louvain Algo: Heuristic. Maximize "Modularity"

$$\text{Modularity} = \frac{1}{2W}\sum_{i,j \text{ in same community}}\left(a_{ij} - \frac{w_i w_j}{2W}\right)$$

- a_{ij} → weight of edge (can be zero)
- $w_i w_j$ → weight of edges to node
- $2W$ → wt of all edges

If trying to minimize each node stays its own community

Neural Nets: Input → Hidden → Output "Neurons"

loop, keep adj. weights (eg. w/ Gradient descent), pick output with highest result.

They suck:
- Lots of data req. to train
- ~~Weighting~~ learning algo tough to tune so weights don't change slow or fast

Competitive Decision Making: Game Theory! Eg. Adjust model and competition reacts.

"Co-operative Game Theory" is compete + co-operate.

- Strategies
  - Pure: only one choice
  - Mixed/Random: Rock, paper, scissors
- Information
  - Perfect: know all about opp.
  - Imperfect: Don't
- Sum
  - Zero: "Benefit" = 0. One win ⇒ One loss
  - Non-zero: "Benefit" > or < 0.

Optimization models used.

OPTIMIZATION

DYNAMIC PROG. (States / Decisions)

① Basic Dyn. Prog (States / decision)
↓ Add prob. of state transitions
② Stochastic Dynamic Program.
↓ w/ discrete states and decision
③ Markov Decision Process (Prob. depends only on current state & decision)

* Optimization thinks that all data is known.
* Bellman's Eq to optimize

MATHEMATICAL MODEL PROGRAMS

* We still need models! Only solver stuff!

→ Variables: Decisions to make
→ Constraints on variables
→ Objective Function: Measure quality of Soln.
  └ * Can incl. input from other models!
→ Solution
  - "Feasible" if satisfies constraint
  - "Optimal" if "Feasible" AND best Obj. Val.

Binary Vars: Create complex models.

① Define in VARS, ② Add to OBJ FUN, ③ Link in constraint

Examples: Max. of any food is M. $x_{c_5} \le M y_{c_5}$ (if c_5 eaten, bin var must be 1)

One or the other $y_a = 1 - y_b$

If one then other is a must $x_a \le M y_b$

Both or none $y_a = y_b$

or: $y_a + y_b = 1$

Model Classification: Linear → Conv Quad → Conv → Int → Gen. Non-Linear (Easy, quick) → (Hard)

Linear: f(x) and CONS Linear $\sum a_{ij} x_i \le b_i$ ($\ge, =$) (if $\log(x)$ constraint also)

Convex Quad: f(x) convex, CONS linear. $\sum c_i x_i^2$, $\sum ax \ge b$

Convex: f(x) convex / concave, CONS convex $\sum c_i |x - 1|$, $\sum ax \le b$

Integer: f(x) linear but CONS has all or few integers. if $x \in \mathbb{Z}$ or $\in \{1, 2, 3\}$

General Non-Convex: Just not convex eg. $\sum c_i \sin x$

Network Model: flow through node / flow through arc } linear function in OBJ FUN / CONSTRAINT
- Shortest Path
- Assignment Model (workers, eg) — "eg. add workers just in case"
- Max flow model

Uncertainty: ① Model conservatively (eg. add θ)
② Scenario modeling $S_1, S_2, S_3, \ldots$ d_{is} ← demand on day i scenario s

~~[crossed out]~~

- Statistical Model: Variable is x_{ij}, Coeff is a_j
- Optimization Model: Variable is a_j, Coeff is x_{ij}

Lasso: $\sum |a_i| \le t$ | **Ridge**: $\sum a_i^2 \le t$ | **Elastic Net**: $\lambda \sum |a_i| + (1-\lambda)\sum a_i^2 \le T$

	VARS	CONS	OBJ FUN
LOGISTIC	$a_0 \ldots a_m$	none	$MAX\left[\prod p(x_i) \prod (1 - p(x_i))\right]$, $p(x_i) = \frac{1}{1+e^{-y_i}}$ (min. pred. error.)
SVM Hard	$a_0 \ldots a_m$	$(a_0 + \sum a_j x_{ij}) y_i \ge 1$	$MAX\left[\sum a_j^2\right]$ (min margin / dist bet. vectors)
SVM Soft	$a_0 \ldots a_m$	none	$\sum \max\{0, 1 - (a_0 \sum y_i)\} + \lambda \sum a_i^2$
Exp Smooth	$\alpha\ \beta\ \gamma$	$0 \le \alpha \le 1$, $0 \le \beta \le 1$, $0 \le \gamma \le 1$	$\sum_{t=1}^{n} (x_t - \hat{x}_t)^2$ (min pred. error)
ARIMA Time Series	μ, φ, θ	none	(" ")
GARCH	$\omega, \beta_i, \gamma_i$	none	$\sum_{t=1}^{n} (\sigma_t^2 - \hat{\sigma}_t)^2$
CLUSTER k-mean	z_{ik} cluster ctr, y_{ik} point	$\sum y_{ik} = 1$ (all assigned to cluster)	$MIN \sum_i \sum_k y_{ik} \sqrt[p]{\sum_j (x_{ij} - z_{jk})^p}$ (min. dist to cluster ctrs)

- R^2 is frac. of variance explained by model.
- Overfitting on training ⇒ Bad perf on Va and Test
- Data (Structured) — quantitative (# with meaning) — independent / time-series. — categorical (# w/o meaning) — Binary. [text is unstruc. data]

- **[SVM]** (Classification): the $\sum a_i^2$ is to be min to max the margin (dist. bet. lines $2/\sum a_i^2$)

optimization: $Min\left[\sum Max(0, 1 - \sum(a_0 + a_i x_i) y_i) + \lambda \sum a_i^2\right]$

Can split that: $\sum_1^{25} + 20 \sum_{26}^{30} + 1870 \sum_{31}^{47}$ → Want to avoid errors in this range more

- Scaling between [0, 1]: $Sx_{ij} = \frac{x_{ij} - min(x)}{max(x) - min(x)}$ to [b, a] (same): $Sx_{ij} \times (a-b) + b$

- Standardize: $Sx_{ij} = \frac{x_{ij} - \mu(x)}{\sigma(x)}$ ← Can be zero!

- **[KNN]** (Classification, Resp. Pred.) new point y_i

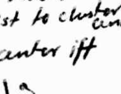
Closeness, $d = \sqrt{\sum (x_i - y_i)^2}$ weighted $d = \sqrt{\sum w_i (x_i - y_i)^2}$

- **[K Means]** (Clustering) (heuristic, Expectation Max) (-ve dist to cluster centers) (CCs)

optimization prob: "Minimize dist. to cluster center iff point is in cluster."

Can do predictive Clustering w/ voronoi diag.

Sum of d to CCs / # of CCs — marginal ROI adding more CCs ← Elbow diag.



[CUSUM] (Change Detection) Done w/ time-series data. Hyp. testing might be slow.

INC $S_t = MAX\{0, S_{t-1} + (x_t - \mu - C)\}$ → pull down sensitivity

Alarm if $S_t \geq T$ (threshold) ⇒ high val: more time to detect, less false alarm

DEC $S_t = MAX\{0, S_{t-1} + (\mu - x_t - C)\}$

[PCA] "Which factors are important?" (TLF) — Reduces correlation, Remove randomness, Rank by importance, Simpler models! (use n top PC's)

1. Scale Data, 2. Normalize mean: $\sum x_i / n = \mu_0 = 0$
3. Get an Eigenvector with sorted Eigenvalues DESC
4. Use regression to find new co-effs. for new factors

New factors are linear combos of old ones!!

k^{th} factor for i^{th} datapoint $= t_{ik} = \sum_i x_{ij} v_{jk}$

$y_i = b_0 + \sum_{k=1}^{L} b_k t_{ik}$ $(b_0 b_1 \ldots b_L)$ → not the same # as orig. model

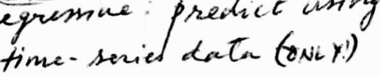
Orig factors! (implied reg. coeff) $a_j = \sum_{k=1}^{L} b_k v_{jk}$ (Do the math)

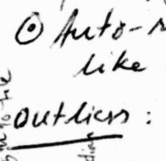
- Auto-regressive: predict using own, old values. like time-series data (ONLY Y!)

135	180	940
450	91	61
76	49	
27	226	1001
583		
	0.3	0.7

47 × 2 = 940

Outliers: point, Contextual (Something's different), Collective (many are different)





Reading Output: p-value: prob. that coefficients are zero → remove if > 0.05 (usually)

t-statistic: p-value / std err

- R^2: "How much variability does model account for?" $R^2 = 0.54$ ⇒ "My model acc. for 54% of variance. 46% was randomness or something else..."

more predictors (or less) mean more error.

Model Quality of Fit: MLE: assume have correct data, assume know variance.

If N.D, Error ~ $N(0, \sigma^2)$. Prob. of seeing z_i if y_i is given by density func (Gaussian): DF. So $MAX[\prod(DF)]$ IF IID ONLY!

⇒ $MIN(z_i - y_i)^2 = MIN[z_i - (a_0 + \sum a_i x_i)]$ → ML Value

(AIC): $AIC = 2k - 2ln(L^*)$ → penalty term. — Bal. likelihood w/ simplicity, avoid overfitting.

for ∞ points. L^* = ML value, k = # of params, n = # of data points. (LOW IS GOOD)

$AIC_c = AIC + \frac{2k(k+1)}{n-k-1}$

Compare AIC = $exp\left(\frac{AIC1 - AIC2}{2}\right) = x\%$ ⇒ Model AIC2 is x% likely to be better than Model AIC1

order of terms matters!

(BIC): Bayesian. Has higher penalty term. (⇒ Models with fewer params than AIC). Use only when more data points than params. ORDER MATTERS!

$BIC1 - 2 > 10$ (Δ): Smaller BIC "very likely" better

$6 < \Delta < 10$ "likely"

$2 < \Delta < 6$ "Somewhat likely"

$0 < \Delta < 2$ "slightly likely"

Splitting Data: Single Model: 70/30, 90/10 (Tr Ts, Tr Ts)

Compare Model: 50/70 (Tr), Split Ts, Va

- Random: may not rep. data equally
- Tr→Va→Tr→Ts→... prob. w/ missing data !! NOT in loop across!
- Rotate: Building eg. MWF in Tr set (days of week)

Tr Train: Fit params; Va Validation: Pick model; Ts Test: Est. quality

Cross Validation: (Tr + Va) / Ts. K-fold X-Validation

- Split (Tr + Va) into K sets, Tr on k-1, Va on 1
- Average errors all folds, pick best
- Train best again on (Tr + Va)!!

- Diff. random effects in training vs test sets. May get boost from high-perf model in training random effects. (Reg. SVM)

Detrending Data: Factor-based models (Reg. SVM) cannot be sensitive to trends, seasonality. So run regression on each factor and remove from prediction (response) like this

Normality Assumption: Not all data is like this

Heteroscedastic ⇒ unequal variance. Use BOX-COX to get some normality. $t(y) = \frac{(y^\lambda - 1)}{\lambda}$

→ Check with a Q-Q plot.

- SVM: Soft and Hard → more critical, life and death. dont put classifier in place that perfectly classifies. Cost of misclassifying is higher.

CONFUSION MATRICES: "True Negative" → not in category, → model is right

Sensitivity = $\frac{TP}{TP+FN}$ → Frac. of Category members correctly identified

Specificity = $\frac{TN}{TN+FP}$ → Frac. of NON Cat. members correctly identified.

- SVM soft: OK to allow for small margins to reduce class. errors in training set.

- A model tells you nothing about outliers! You need to prep data... GIGO, yo.

Exp Smoothing (Resp. Pred.)

Use for time-series data to predict ~~deviations~~ baseline response. Bit like CUSUM but not really.

Est Expect / Observ / Prev Est

$S_t = \alpha x_t + (1-\alpha) S_{t-1}$ @ time t

baseline, $0 < \alpha < 1$

"observed is real indicator" → $S_t = x_t$ ⇒ new baseline

"random luck... keep same" → $S_t = S_{t-1}$ ⇒ same, random luck

[So, if too random, keep the old (S_{t-1}) with low α
if not too random, keep the new (observed value) w/ high α ($1/2 < \alpha < 1$ eg.)

$\alpha \to 0$ for lots of randomness

$\alpha \to 1$ for not much randomness.

Trends: Eg: Spikes during Christmas time.

ADDITIVE! $S_t = \alpha x_t + (1-\alpha)(S_{t-1} + T_{t-1})$

$T_1 = 0$ $T_t = \beta(S_t - S_{t-1}) + (1-\beta)T_{t-1}$

Seasonality/Cycles: Inflate/deflate the obs. x_t,

MULTIPLICATIVE! not (baseline + trend)

$S_t = \frac{\alpha x_t}{C_{t-L}} + (1-\alpha)(S_{t-1} + T_{t-1})$ "L" cycles/period eg: BP measured every hour/day ⇒ L = 24

$C_t = \gamma\left(\frac{x_t}{S_t}\right) + (1-\gamma) C_{t-L}$ C=1 no change, C=0.9 ⇒ 10% lower | 1.1 ⇒ 10% higher

(see the effect of C_t on the observed value x_t)

Forecasting: $F_{t+1} = \alpha x_t + (1-\alpha) S_t$ | $F_{t+1} = S_t$ initially

w/ trends: $F_{t+k} = S_t + kT_t \ \forall k \in N$

w/ cycles: $F_{t+k} = (S_t + kT_t)\, C_{(t+1)-L+(k+1)}$

To find α, β, γ, optimization prob: $MIN (F_t - x_t)^2$

Good for short-term stuff

ARIMA (Resp. pred., classification): Again, time-series only! Use when data is not stationary (ie, μ, σ^2 not constant over time)

But diff. can be stationary. d-order diffs.

3-order diff: diff(diff(diff)) ← chained

(p) time periods, to predict (d)-order diffs

Basically auto-regression applied to diffs.

Uses prev. errors as predictors: (q) time-periods (???) [the order moving avg]

ARI(0,0,0) (p d q) White Noise, (0,1,0): Random walk, (p,0,0): basic auto-reg. model (0,0,q): Moving Avg model.

ARIMA(0,1,1) = EXPONENTIAL SMOOTHING! ♡♡♡

So far, to forecast values.

Need something to forecast variance.

GARCH (Variance Est.): Need variance → estimate error
↳ make a proxy for something (eg. investment risk)

"How much is forecast hi/low than truth"

GARCH(p,q)

- use variances (sq. diffs)
- use RAW! (NOT diff. of variances!)
- Is also autoregressive.

REGRESSION (Classification, resp. pred.), (Desc, Pred)

Linear ♡: $y_i = a_0 + \sum_i a_i x_i$ (pred) ‖ $\hat{y}_i$ (real) ‖ Error = $(\hat{y}_i - y_i)$

Optimization problem: $MIN\left[\sum_i (\hat{y}_i - y_i)^2\right] \forall (a_0 .. a_n), n \in N$

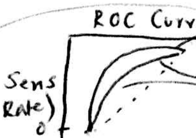
(Response is continuous)

Logistic (Classification): use for prob. values (can turn to binary classifier eg: $p > 0.7 \Rightarrow 1$)

Linear: $y_i = a_0 + \sum_i a_i x_i$, take and make $\log\left(\frac{p}{1-p}\right) = y_i$

⇒ $p = \frac{1}{1+e^{-y_i}}$, when $y_i = -\infty$, p = 0; $y_i = +\infty$, p = 1

- Pseudo R^2
- Takes longer
- No "closed form"



if = 0.5, is a guess

Prob. that logr model will give a high p to someone from YES group (in Category) than NO group

No idea of the cost of false +ve, false -ve.

Poisson: if not ND. eg: Count arrivals of people at airport line.

$f(z) = \frac{\lambda^z e^{-\lambda}}{z!}$ → NOT a param! function of something (eg. time)

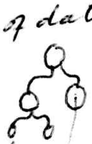
Est. $\lambda(x)$

Splines: use knots and k-splines (smooth) to fit
↳ (non-parametric)

Bayesian: Not much data // use some coeffs, rand. err to start. OR use a "broad prior dist" as seed data....

KNN: LOL: Give avg. resp. of all k-neighbors. Can weight resp. too
↳ (non-parametric)

CART: Split regression models up into tree. Model per leaf. Must have at least 5% of data in each leaf (else overfitting)



BUILD ① Take half the data: foreach(Leaf): foreach(factor): compare σ^2, split if < threshold (check size) ↳ ie, pick factor with lowest total variance.

PRUNE ② Take other half. foreach(branch-point): estimate error (w and w/o branch), prune if > threshold. Some est. of quality. "x > 2" prune and make binary lol.

Random Forest: ① Bootstrap: Pick from data randomly. Some appear once, some more than once, some not at all. Same size as dataset!
② Branch on random (1+log n) factors. NO PRUNING STEP!
ALSO can give data on imp. of each factor based on # of branches.

Regression Tree? Average the responses

Classification Tree? Most common response (mode)

Good:
- Better overall est.
- overfitting nbd. of averaging

BAD:
- Hard to explain/interpret
- Cannot point to specific model.