- Class Notes - Intro

Like DevOps, automation & workflows in ML → MLOps but it's code + data
MLOps combines ML, DevOps & Data Engineering

Kubeflow pipelines – Visualize ML pipelines
Tensorflow transform – Check consistency
Automate unit tests & integration test → TF data validation

- Descriptive - Predictive - Prescriptive - Retrospective

Key items in ML pipeline → Monitoring
ideation (EDA) | data acquisition & exploration | R&D iterations | Validation | Serving/ Delivery

- Vertex AI – Google

New ML-problem → Optimized Marketing using RL

Characteristic	NN	SVM	Trees	MARS	k-NN, kernels
Handling mixed data	↓	↓	↑	↑	↓
Missing Values	↓	↓	↑	↑	↑
Outliers	↓	↓	↑	↓	↑
Insensitive to monotonic ∅	↓	↓	↑	↓	↓
Scalability	↓	↓	↑	↑	↓
Irrelevant inputs	↓	↓	↑	↑	↓
Linear comb^n of X	↑	↑	↓	↓	~
Interpretability	↓	↓	~	↑	↓
Predictive power	↑	↑	↓	~	↑

MARS → auto detects non-linearity & interaction & non-parametric

- Neural net issues → ↑ data, ↑ hyp. tuning ⇒ ↑ compute ↓ stats insight
- Correlation Causation
- Simpson's paradox
- IHME → outside 95% CI, 70 percent times ⇒ ↑ noise ⇒ ↓ SNR.

Curvefit – Non linear mixed effects → fixed ? confidence - ve for nested random
→ random

* For fixed effect Fisher Information matrix used to get asymptotic approx of uncertainty.
can understand Fixed vs Random from Simpson's paradox case.

Gaussian density $N(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\cdot z^2}$

$E(x) = \mu \quad E(x^2) = \mu^2 + \sigma^2$
$Var(x) = \sigma^2$

Likelihood = $\prod$ density of i instances (under independence assumption)
overall

Maximum Likelihood Estimation – $\theta_{MLE} = \arg\max_{\theta} p(X|\theta)$

Distribution

Exponential

$x \geq 0$ $\quad \lambda e^{-\lambda x}$: pdf

$x \geq 0$ $\quad 1 - e^{-\lambda x}$: cdf

mean	median	var^n
$(1/\lambda)$	$(\ln 2/\lambda)$	$(1/\lambda^2)$

Geometric pdf : $(1-p)^{k-1}p$

cdf : $1-(1-p)^k$

mean	median	var^n
$(1/p)$	$-1/\log_2(1-p)$	$(1-p/p^2)$

Normal pdf $= \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}z^2}$

μ $\quad \mu$ $\quad \sigma^2$ $\quad$ cdf $= \frac{1}{2}\left(1 + \text{erf}\left(\frac{z}{\sqrt{2}}\right)\right)$

where $\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$

Binomial ${}^nC_x\, p^x (1-p)^x$

Poisson $\lambda^x e^{-\lambda}/x!$

Log Likelihood

$$-n\lambda + \sum_{i=1}^{n} \lambda x_i = \frac{d[\ln \mathcal{L}(\lambda)]}{d\lambda}$$

$\Rightarrow$ max at $\frac{1}{\text{mean}}$

But $\bar{x}$ is unbias est^r of $1/\lambda$ $\Rightarrow$ $\hat{\lambda}_{MLE} = \left(\frac{n}{\sum x_i}\right)$ not unbiased estimator of λ

$$\frac{d \ln \mathcal{L}(p)}{dp} = \frac{n}{p} - \frac{\sum_{i=1}^{n} x_i - n}{(1-p)}$$

$\Rightarrow \hat{p}_{MLE} = \left(\frac{n}{\sum x_i}\right)$ not unbiased estimator of p

$$\ln \mathcal{L}(\mu, \sigma) = -\frac{n}{2}\ln(2\pi\sigma) + \frac{\sum (x_i - \mu)^2}{2\sigma}$$

wrt $\hat{\mu} \Rightarrow \frac{1}{\sigma}\sum(x_i - \mu) \Rightarrow \boxed{\bar{x}}$ for zero

wrt $\hat{\sigma} = -\frac{n}{2\sigma} + \frac{1}{2\sigma^2}\sum (x_i - \mu)^2 \Rightarrow$ for zero

$\hat{\sigma} = \boxed{\frac{1}{n}\sum (x_i - \bar{x})^2}$ min var unbias est (MVUE) (lil underpred)

$\hat{p}_{MLE} = \frac{k}{n}$ for k-successes

$\hat{\lambda}_{MLE} = \frac{\sum x_i}{n}$

Comments

pdf of time betⁿ events in Poisson point proc^n

$\Rightarrow$ event countⁿ & indepⁿ of constant avg rate.

Special case of gamma Continᵘˢ analogue of geometric

Approx. bias corrⁿ for MLE $= \frac{n-2}{(\bar{x})}$

to model # trials until success : discrete

Assumpⁿ: Independence, binary, constant p

Inflections at $\mu \pm \sigma$ (CP $\Rightarrow$ another normal)

Cauchy $\leftarrow$ (df = 1) Student's t $\rightarrow$ Normal (df = ∞)

Minimizing MSE on train data in regression setting yields MLE solution of assumed generative model

$B(n,p) \rightarrow N(np, npq)$ for $\uparrow n$ (CLT)

$B(n,p) \rightarrow \text{Pois}(np)$ for $\uparrow n$ & $\downarrow p$

(Bayesian) Conjugate Priors — Posterior = Likelihood * Prior ⇒ prior is CP of likelihood

★ Generally 1 hyp. param ↑

Likelihood	Conjugate Prior	Posterior Hyp.	Interpretation
Bernoulli (p)	Beta (α, β)	$\alpha+\bar{x}\cdot n,\ \beta+n-\bar{x}\cdot n$	α success, β failures
Binomial (p)	Beta (α, β)	$\alpha+\bar{x}_{\cdot n},\ \beta+\Sigma n-\bar{x}\cdot n$	—— 〃 ——
Poisson (λ)	Gamma (α, β)	$\alpha+\bar{x}\cdot n,\ \beta+n$	α total occurence in β interval
Geometric (p)	Beta (α, β)	$\alpha+n,\ \beta+\bar{x}\cdot n$	α experiment, β failures
Multinom	Dirichlet		
Normal (μ, σ^2)	Normal	complicated	
Uniform	Pareto		
Pareto	Gamma		
Exponential	Gamma	$\alpha+n,\ \beta+\bar{x}\cdot n$	

Multivariate Gaussian Distribution:

$$N(\vec{x}|\mu,\Sigma) = \frac{1}{(\sqrt{2\pi})^D\sqrt{\Sigma_{det}}}\, e^{\left\{-\frac{1}{2}(\vec{x}-\mu)^T\Sigma^{-1}(\vec{x}-\mu)\right\}}$$

* Marginal & Conditional prob. of normal are normal themselves

- ML Short new

Anova → F stat = $\frac{MSR}{MSE}$ t stat = $\left(\frac{\beta - X_{test\,val}}{SE}\right)$ Adj R^2: penalize n↑

OLS → sales ~ price$_1$ + price$_2$ + ϵ_1 — Nested impl of plm in R Check?

Fixed effects → sales ~ $p_1 + p_2$ + city fixed effects + ϵ_2

Random effects → sales ~ $p_1 + p_2$ + cfe + time fixed effects + ϵ_3

↳ can model ↑ cat. as RE ~ $N(\mu, \sigma^2)$

$p_1 + p_2$: agnostic; cfe: within city (individual); time fixed effects: (intercity + within city); ϵ_3: (idiosync error)

$$\delta^2_{b_j} = \frac{\delta^2}{(N-1)(\text{var in } X_j \text{ not ass. with other } X's)}$$

- Succint models with superior features are accurate, robust ★★

* Function Approximation — Regression

$y(x, w) = w_0 + w_1 \phi_1(x_1) + \dots + w_n \phi_n(x_n)$ to predict t (true/target)

w: prediction weights for dependent variable; x: independent variables; ϕ: basis func^n

$E(t)$ given ϕ_i is linear in $\phi \Rightarrow E(t|\phi_i) = \text{Linear Combo}(\phi_i)$

↳ Distribution assumed iid + zero mean, constant variance + Gaussian ie $N(0, \sigma^2)$

- parametric form

Assumptions

- $E(t|\phi_i)$ linear in ϕ_i
- Exogeneity in ϕ_i
- Errors random $N(0, \sigma^2)$
- Homoskedasticity
- No multicollinearity
- No serial autocorrelation

• Check for Simpson's paradox especially in panel data

Test for Normality – Kolmogorov-Smirnov (CDF) Shapiro-Wilk (better for ↓ n)

Test for multicollinearity – Breusch Pagan (test st. is Chi-sq distributed)

Test for autocorrelation – Durbin Watson (DB only checks with lag 1)

Test for (error) endogeneity – Hausman

MLE provides error metric as squared loss for best estimator

* MVUE ~ non-linear BLUE

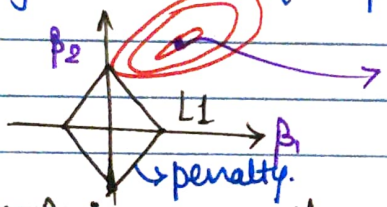
* Exact closed form sol^n ⇒ iterative approach not needed

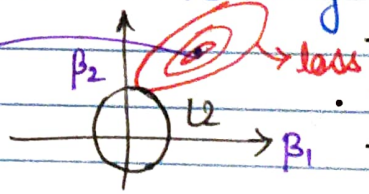
$\vec{w}^* = \underbrace{(\phi^T \phi)^{-1}}_{\text{Moore Penrose Inverse}} . \phi^T . \vec{t}$ If collinear ⇒ $\det(\phi^T\phi) \to 0$ ⇒ inverse becomes large ⇒ parameter estimates have higher uncertainty (No effect on R^2 cause info. is there)

SEs are defined as —

$$s_{b_j} = \frac{s^2}{(N-1)(\text{Variation in } X_j \text{ not associated with other } x_i\text{s})}$$

Regularization: Add penalty to whatever's causing overfit to cost f^n





- Elastic Net
- # variables
- Smoothness of function

As λ increases, chance of betas being shrunk to zero increases.

• Intercept not included case – ① No need, intercept should be close to $E(y_i)$ ② If regularized prediction of model with $y_i + c \sim \forall x_i$ would not necessarily increase by c.

• Standardize all x_i, else penalty could be overpowered.

• Deviance vs $-\log\lambda$ graphs with variance error bars & λ_{min} & λ_{1se}

• called weight decay in NN – shrinkage in stats

$\vec{w}^* = (\lambda I + \phi^T\phi)^{-1} . \phi^T . \vec{t}$ ⇒ adds numerical stability

Model evaluation → f(train, validation) ⇒ collection of solutions ⇒ CI's on βs
↳ not in Bayesian ⇒ No CIs ↳ no probab distribution

• Bias Variance Trade-off

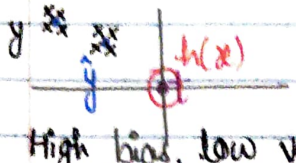
Say $t = h(x) + \varepsilon$ where $\varepsilon = N(0, \sigma^2)$ ⇒ $h(x)$ is best possible function in x

Our prediction = $y(x)$ ⇒ Expected Loss = $\int (y(x) - h(x))^2 p(x)\,dx$ (bias² + Variance)
$+ \iint (h(x) - t)^2 p(x,t)\,dx\,dt$ (irreducible)

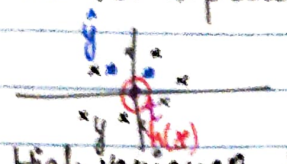
for dataset D we have

$(h(x) - E[y(x,D)])^2 + E\left[(y(x,D) - E(y(x,D)))^2\right]$

* t not function of x ⇒ double integral
* Given $h(x)$ is unique, bias² is not enclosed in over the expectation
* Given multiple models ⇒ Variance is double expectation

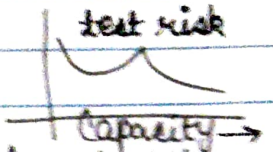


High bias, low variance



High variance, low bias

→ property of model



* Regularization due to over parameterization in neural networks

Explaining Bias-Variance trade-off → # bins in histogram.

Incremental Forward Stagewise Regression -

Residual $r \leftarrow y$, and $\forall i, \beta_i = 0$; $\forall x_i$ we have $\sum x_i = 0$ & $\sum x_i^2 = 1$ (unit norm)

$\left[\beta_i \leftarrow \beta_i + \varepsilon \cdot \text{sign}(\overset{\text{max correlation}}{\tau}(x_i, r))\right]$ ⇒ uptil r uncorrelated with all x.

& $\left[r \leftarrow r - \varepsilon \cdot \text{sign}(x_i, r) \cdot x_i\right]$ Q. shouldn't this be repeated too many times, but seems $O(n)$?

Other Regressions -

→ Partial Least Squares

→ Principal Components Regression

Logistic, Polynomial,
Quantile (Median = least abs. dev.)
Elasticnet

→ Support Vector Regression

→ Poisson Regression vs Negative Binomial vs Quasi poisson regression
($\mu = \sigma^2 = \lambda$) ($\mu \neq \sigma^2$) but Quadratic $\sigma^2 = \text{Quad}(\mu)$ $\sigma^2 = \text{Linear}(\mu)$

→ Cox Regression (Coxph → proportional hazards generally never satisfied)

→ Tobit Regression

→ Least Angle Regression (Efficient Lasso)

→ Panel Regression

Succint models with 'right' predictors are superior.

Transformation types → Log (in case of outliers in X or subsequently in residuals)
→ Standardization $\left(\frac{x-\mu}{\sigma}\right)$ → Minmax (shift to 0,1 range)
- → Box Cox transformation → $\log x$ $\forall \lambda = 0$ } find closest to gaussian
  $\frac{(x^\lambda - 1)}{\lambda}$ $\forall \lambda \neq 0$
- → Yeo Johnson (similar but separate treatment for negatives)

→ RankGauss (values → Rank → Gaussian)**
→ Freq to Rank encoding ; Target encoding ; Binary encodings

Almost linear methods –
- Additive models – GAMs
- Piecewise linear – MARS
- Locally linear – LOESS

* Universal Approximator – ∀ continuous function, member of the family exactly matches it.

Often E(w) is non-linear, not a quadratic ∴ often non-convex ⇒ SGD
Incrementally 'surf' down the cost function in weight space

$w^{(\tau+1)} = w^{(\tau)} - \eta . \vec{\nabla E}$ eta

Adaptive eta – ① Nelder-Mead (Contraction vs Expansion)
Scaling Nelder Mead to n dimensions – use (n+1) points convex poly.
Can become divergent in any edge case?

Gradient Descent → Elegant move towards nearest minima (can take time) will oscillate if eta set too high. Line Search – somehow
- adapt eta so that loss is always decreasing**

* GD faces issue with Rosenbrock: $(1-x)^2 + 100(y-x^2)^2$ (getting stuck for few inits & slow) → steep & variable change in surrounding

Conjugate Gradient – Take into account previous search direc^n (momentum?)

$S_n = \nabla X_n + \beta S_{n-1}$ where $\beta = \frac{\nabla X_n^T . \nabla X_n - \nabla X_n^T . \nabla X_{n-1}}{\nabla X_{n-1}^T . \nabla X_{n-1}}$

using search direction & component of previous search direction we have next x location as $X + = \alpha S_n$ (from line search)

* Helps in cases like Rosenbrock. Is momentum about acceleration & conjugate gradient about direction? avoids the zig-zag problem

PCA = MDS (Euclidean Dist) → XX^T
↓ eigendecompose covariance mat^x $(X^TX).\vec{v} = \lambda\vec{v}$
(Dist).$\vec{v} = \lambda\vec{v}$ ⇒ $(XX^T).\vec{v} = \lambda\vec{v}$
Both are giving same $\lambda, \vec{v}$

MDS = Principal Coordinates Analysis (PCoA)
Loss function strain – difference between actual & wanted distances**
translates to eigendecomposition in case of Euclidean

Conjugate Gradient → faster than line search, more smoothened as well.
SGD → better in online sense → streaming data → non-stationary
- Ideally decrease η with time to allow convergence (a/t)

Batch → #epochs* (one pass over all data where 1 batch taken in 1go)
SGD → faster, better for large data, ↓ prone to local minima, non-stationary is fine

• Given GD is slow ⇒ leverage 2nd order ⇒ Newton's method ⇒ Quadratic approximation of Taylor in 1d

In multi-dimension – $x_{n+1} = x_n - \nabla(f(x))\left(H(f(x))\right)^{-1}$ → Difficult to compute the Hessian & inverse issues.

$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)}$.

could move towards global minima given assumption holds true ⇒ Big steps in low curvature & vice-versa. Better to have line search SGD + leverage previous directions ⇒ Conjugate grad

Jacobian (gradients wrt $\forall x$) Hessian (curvatures wrt $\forall x$)

* Neural networks: Adaline vs Perceptron } Both are classifiers – slight difference

#weights = Σ #pairs ↳ $t - f(x)$ ↳ $t - \text{step}(f(x))$

#biases = # perceptrons (#linear combinations)

Multi-layered perceptron → if output is linear & hidden are non-linear ⇒ Universal Approximator

#total weights = (Inp+1) Hidden$_1$ + (Hidden$_1$+1) Hidden$_2$ + (Hidden$_2$+1) Output

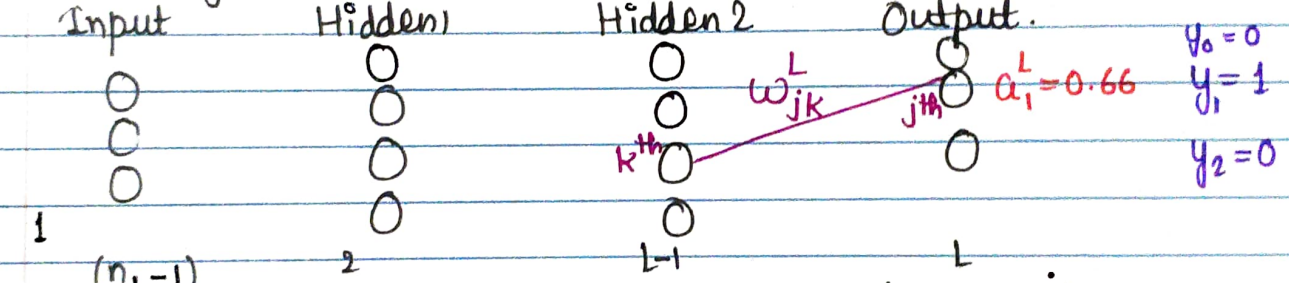
Why logistic function for sigmoid curve? $\sigma(x) = 1/1+e^{x}$ $\sigma'(x) = \sigma(x)(1-\sigma(x))$

Forward pass – Given input calculate output layer by layer

Compute error at final output layer

Backward pass – Compute gradient of weights wrt loss, layer by layer

Update all weights.



$\text{Cost} = \sum_{j=0}^{(n_L - 1)} \left(a^L_j - y_j\right)^2$ (or average)

for simplicity where $z^L_j = \sum_{k=0}^{(n_{L-1}-1)} w^L_{jk} \cdot a^{L-1}_k + b^L_j$

$a^L_j = \sigma(z^L_j)$

⇒ The chain rule becomes –

$$\frac{\partial \text{Cost}}{\partial w^L_{jk}} = \frac{\partial \text{Cost}}{\partial a^L_j} \cdot \frac{\partial a^L_j}{\partial z^L_j} \cdot \frac{\partial z^L_j}{\partial w^L_{jk}}$$

Cost sensitivity wrt all weights – if multiple paths ⇒ add those up.

$$2(a^L - y) \cdot \sigma'(z^L) \cdot (a^{L-1})$$

Given this is for two general layers, recursively repeat for all the layers. Use gradient term to update weights.

⇒ Store gradients somewhere to avoid recompute.

SGD → Man walking down the hill – slow, careful & steady
Momentum → Heavy ball rolling down the same – Newton's 1st law.
Robustness → rlm() in R or SVR → Not a function given 1-M solutions ⇒ Model joint probability
Modeling inverse problem?
* Sequential Learning with SGD → non-stationary or piecewise models ⇒ you don't want convergence
If you want convergence ⇒ η decreases with time Eg. Rmsprop.
$\eta(t) = 1/t$ ⇒ Robbins-Monroe sequence

RNNs ⇒ sequence ip → h_{t-1} → op but no memory (long) (BPT) → h_{t-1}
LSTMs memory via forget gate tanh & σ in ip gate ⇒ σ f(ip+forget) tanh ⇒ op
AEs image compression possible - denoising
VAEs write new hand written images i/p to distrib mapping then sampling KL Divergence ∫p log p/q to check
GRU reset & update gates lesser operations/ops on hidden states ⇒ faster

Markov Chain Ily state transition matrix
Rat Boltzmann machine patterns by reconstrn i/p
Convolutional NN basically rdim NN filters, stride padding etc.
sampling parameters as loss (is random $N(0,\sigma^2)$)

GANs* Graph NNs* Attention Network*

* Data Preprocessing
Boxplot: Whiskers upto extremeties or 1.5IQR outside IQR whichever nearest
Missing: Check if random/systematic else fill in conditional mean/mode/knn
Outlier detection → Prob. based ⇒ Estimate pdf(x) given $x_i \forall i$
Parzen window ⇒ kernel density Estimate ⇒ Approx f as mix of continuous(K) distributions centered at x_i with bandwidth h. K is window or kernel

$$\hat{f}_h(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

K can be Normal, poission etc.

*
Q. Difference between GAMs, KDE and (Gaussian) Mixture models?

Discrimination based: 3 sigma, Freq less than 1%, dist^n > Average dist^n k^{th} neighbour
* Trimmed mean * Huber loss instead of square loss
* Post hoc → Roughly monotonic to monotonic by isotonic regression
PyOD → COPOD