Linear Regression:

ML Pipeline: fct Class $\mathcal{F}$, Training loss ℓ, Optim. Method

Multiple Lin Reg: loss fcts: $\ell_{sq}(f(x),y)=(y-f(x))^2$, $\ell_{abs}(f(x),y)=|f(x)-y|$, $\ell_{huber}=\begin{cases}\frac12(f(x)-y)^2 & \text{if } |f(x)-y|\le\delta\\ \delta(|f(x)-y|-\frac12\delta) & \text{if } |f(x)-y|\ge\delta\end{cases}$,

$\ell_\tau(f(x),y)=\tau\max\{0,y-f(x)\}+(1-\tau)\max\{0,f(x)-y\}$, $\tau\in[0,1]$

↳ $\tau\in[0,\frac12)\Rightarrow f$ underestimates, $\tau\in(\frac12,1]\Rightarrow f$ overestimates

LinReg Estimator: $\hat w=\arg\min_{w\in\mathbb{R}^d}\|y-Xw\|_2^2=(X^TX)^{-1}X^Ty$ ($d\le n \Leftrightarrow X^TX$ invertible)

↳ $\hat y=X\hat w=X(X^TX)^{-1}X^Ty=\Pi_X y$ ("projection onto span(X)")

↳ $d>n\Rightarrow X^TX$ not investible $\Rightarrow \hat w$ not unique bec. $\dim(\mathrm{Ker}(\Phi^T\Phi))>0$

↳ Min Norm Solution: $\hat w=\min_{w\in\mathbb{R}^d}\|w\|_2$ s.t. $y=Xw$

$\Rightarrow \hat w=X^T(XX^T)^{-1}y=(X^TX)^+X^Ty$ where $(X^TX)^+=X^T(XX^T)^{-1}(X^T)^{-1}$

NonLinear LS: $\hat w=\arg\min_{w\in\mathbb{R}^p}\|y-\Phi w\|_2^2=(\Phi^T\Phi)^{-1}\Phi^Ty$

$\Phi^T\Phi$ investible $\Leftrightarrow \mathrm{rank}(\Phi)=p$, $\vec\nabla^2\mathcal{L}(w)=\Phi^T\Phi \Rightarrow$ pos.def. $\Rightarrow$ unique min

Remarks: $\vec w=\binom{w_0}{w_1}$ & $\vec x=\binom{x}{1}\Rightarrow w_0^*=\bar y-w_1^*\bar x$, $w_1^*=\frac{Cov(x,y)}{Var(x)}$

Optimization: Iterative Approach:
⊕: Comp. cheap if ℓ, f compl. ⊖: might not conv.

GD: $w^{t+1}=w^t-\eta\vec\nabla_w\mathcal{L}(w^t)$

Conv. of GD for LinReg: GD converges for $\min_{w\in\mathbb{R}^d}\frac12\|y-Xw\|_2^2$ if $\eta<\frac{2}{\lambda_{max}(X^TX)}$, $\eta_{opt}=\frac{2}{\lambda_{max}+\lambda_{min}}$, ill / well conditioned if $\lambda_{max}\gg\lambda_{min}$ / $\approx$

Other Grad.-Based Methods: 2^{nd} Order: use Hessian of $\mathcal{L}$

Momentum: $w^{t+1}=w^t+\alpha(w^t-w^{t-1})-\eta\vec\nabla\mathcal{L}(w^t)$ ($\alpha=\beta\cdot\eta$, $\beta\in[0,1)$)

Adaptive: $w_i^{t+1}=w_i^t-\frac{\eta}{\sqrt{\delta_i^t+\gamma}}\frac{\partial\mathcal{L}}{\partial w_i}(w_i^t)$, $\delta_i^t=(w_i^t-w_i^{t-1})^2$

Stochastic GD: random select 1 Datapoint & use for GD ✓ ($\vec\nabla\mathcal{L}=\vec\nabla_w\ell(f(x_i),y_i)$)

(Mini)Batch SGD: $S\subseteq\{1,\dots,n\}$: $\vec\nabla\mathcal{L}_S(w)=\frac{1}{|S|}\sum_{i\in S}\vec\nabla_w\ell(f_w(x_i),y_i)$ (← $|S|=1$)

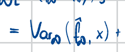
↳ In practice: $D=S_1\sqcup\dots\sqcup S_n$ & train Model on S_i's

↳ $|S_i|$ large $\Rightarrow$ Var ↓ & Comp. Compl. ↑

Remarks: $w^k=\underbrace{(\mathbb{1}-\eta X^TX)^k}_{=A}w_0+\eta\underbrace{\left(\sum_{j=1}^{k-1}(\mathbb{1}-\eta X^TX)^j\right)X^T}_{=B}y$

↳ $A=V(\mathbb{1}-\eta\Lambda)V^T$ (Proj Mat to Span(X)), $B=V\begin{pmatrix}0\\\frac{1}{\sigma_i}\end{pmatrix}_{ii}U^Ty$ (0 if $\sigma_i=0$, $\frac{1}{\sigma_i}$ if $\sigma_i\ne0$)

Convexity: $f(\lambda x+(1-\lambda)y)\le\lambda f(x)+(1-\lambda)f(y)\Leftrightarrow\nabla^2f(x)$ p.s.d. $\forall x$

$\Leftrightarrow f(y)\ge f(x)+\langle\nabla f(x),y-x\rangle\Leftrightarrow$ 

Lemma: (i) f,g conv., $\alpha,\beta\ge0\Rightarrow\alpha f+\beta g$ conv., (ii) $\max\{f,g\}$ conv. if f,g conv. ✓

(iii) $f\circ g$ convex if $\begin{cases}f \text{ convex \& } g \text{ affine or}\\ f \text{ (convex \&) non-decreasing}, g \text{ convex}\end{cases}$

(iv) f convex, diff., x^* stat point $\Rightarrow x^*$ global min (if f strongly conv. $\Rightarrow x^*$ unique minimizer)

Strongly Convex with $m>0\Leftrightarrow f(\lambda x+(1-\lambda)y)\le\lambda f(x)+(1-\lambda)f(y)-\frac m2\lambda(1-\lambda)\|x-y\|^2$

Strongly Convex $\overset{\not\Leftarrow}{\Rightarrow}$ strictly convex, $\nabla^2f-m\mathbb{1}$ p.s.d. $\Leftrightarrow\lambda(\nabla^2f)\ge m$

Model Evaluation & Selection:

Estim Err: $\ell(\hat f(x),f^*(x))$, **Pred. Err:** $\ell(\hat f(x),y)$

Gener. Err.: Average Pred Error: $\frac1n\sum_{i=1}^n\ell(\hat f(x_i),y_i)$

Estim of GE: Empirical Risk: $\mathcal{L}(f;D)=\frac1n\sum_{i=1}^n\ell(f(x_i),y_i)$

$\Rightarrow\hat f_D=\arg\min_{f\in\mathcal F}\mathcal{L}(f;D)$, GE too optimistic if train & eval on D

$D=D_{train}\sqcup D_{test}$ (evaluate GE on D_{test} & train on D_{train}), $|D_{train}|>|D_{test}|$: better for model training, but test error is not very accurate. $|D_{train}|\approx|D_{test}|$: worse for training but better estim of GE.

Cross Validation: If $D=D_{train}\sqcup D_{test}\Rightarrow\hat f$ still depends on D_{test} (bec. choose $\hat f$ s.t. GE small) $\Rightarrow$ Cannot give estim of GE

Model Select. w/ val. set: $D=\underbrace{D_{train}\sqcup D_{val}}_{D_{use}}\sqcup D_{test}$

↳ D_{train} for training, use D_{val} to check for lowest error among $\hat f$, D_{test} to estim GE

K-fold CV: $D_{use}=D_1\sqcup\dots\sqcup D_K$. Use $D_{use}\setminus D_i$ for train. & calc. val. loss on D_i. Repeat $\forall\mathcal{M}_r$ & $D_i\Rightarrow$ obtain avg $\mathcal{L}$ $\forall\mathcal{M}_r$

$\mathcal{L}_i^{(\mathcal M)}:=\mathcal{L}(\hat f_{\mathcal M,i};D_i)=\frac{1}{|D_i|}\sum_{(x_i,y_i)\in D_i}\ell(\hat f_{\mathcal M,i}(x),y)$

$\Rightarrow$ Crossvalidation Error: $CV_K(\mathcal M)=\frac1K\sum_{i=1}^K\mathcal{L}_i^{(\mathcal M)}$

$\Rightarrow$ Select $\mathcal M_r$ with lowest $CV_K(\mathcal M_r)\Rightarrow$ train $\mathcal M_r$ on whole D_{use}

$\hat f=\hat f_{\mathcal M^*,D_{use}}$ & estim GE as $\mathcal{L}(\hat f;D_{test})=\frac{1}{|D_{test}|}\sum_{(x_i,y_i)\in D_{test}}\ell(f(x_i),y_i)$

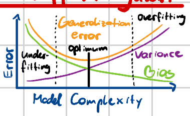
LeaveOneOut CV: $K=|D_{use}|$, $|D_i|=1\Rightarrow$

K↑: Submodels closer to full model BUT decr. prec. of $\mathcal{L}_K^{(m)}\approx GE$

Bias-Variance Tradeoff & Regular.

$\mathcal F$ too simple $\Rightarrow$ underfitting

$\mathcal F$ too complex $\Rightarrow$ overfitting



BV-Tradeoff: $\hat f_D=\mathcal M(D)$

Model Bias: $Bias_D^2(\hat f_D,x):=\left(\mathbb{E}_D[\hat f_D(x)]-f^*(x)\right)^2$ (Ground truth)

$Bias_D^2(\hat f_D)=\mathbb{E}_x[Bias_D^2(\hat f_D,x)]\Rightarrow$ Simple Model: large Bias, Complex Model: small Bias

Model Variance: $Var_D(\hat f_D,x):=\mathbb{E}_D[(\hat f_D(x)-\mathbb{E}_D[\hat f_D(x)])^2]$

$=\mathbb{E}_D[(\hat f_D(x))^2]-(\mathbb{E}_D[\hat f_D(x)])^2\Rightarrow Var_D(\hat f_D)=\mathbb{E}_x[Var_D(\hat f_D,x)]$

Approx: $\mathbb{E}_D[\hat f_D]\approx\bar f=\frac1J\sum_{j=1}^J\hat f_j$, $Var_D(\hat f_D)\approx\frac1J\sum_{j=1}^J(\hat f_j(x)-\bar f(x))^2$

↳ Simple Model: small Var, Complex Model: large Var

BV-Decomp.: $\mathbb{E}_D[\mathcal{L}(\hat f_D;P_{X,Y})]=Var_D(\hat f_D)+Bias_D^2(\hat f_D)+\sigma^2$ ($var(\varepsilon_i)$, irred. err.)

$\mathbb{E}_D[(\hat f_D(x)-f(x))^2]=Var_D(\hat f_D,x)+Bias_D^2(\hat f_D,x)$

Lasso Regr.: $\min_{w\in\mathbb{R}^d}\|y-Xw\|_2^2$ s.t. $\|w\|_1\le R$ (small R $\Leftrightarrow$ large λ)

$\Leftrightarrow\min_{w\in\mathbb{R}^d}(\|y-Xw\|_2^2+\lambda\|w\|_1)\Rightarrow$ If $X^TX=\mathbb{1}_d\Rightarrow[\hat w_\lambda]_j=sign(w_j^{LS})\cdot\max\{|w_j^{LS}|-\lambda,0\}$

$\Rightarrow$ Lasso induces Sparsity (β_j's $=0$). If input dim's highly correlated (Columns of X) $\Rightarrow$ Lasso might not be stable

Ridge Regr.: $\min_{w\in\mathbb{R}^d}\|y-Xw\|_2^2$ s.t. $\|w\|_2\le R\Leftrightarrow\min_{w\in\mathbb{R}^d}(\|y-Xw\|_2^2+\lambda\|w\|_2^2)$ (shrinks all params)

↳ $\hat w_{ridge}=(X^TX+\lambda\mathbb{1})^{-1}X^Ty=\frac{1}{1+\lambda}\hat w^{LS}$, $Bias^2(w_{ridge})=\|\lambda(\Phi^T\Phi+\lambda\mathbb{1})^{-1}w^*\|^2$

$Bias_D^2(\hat f_D)=\|V\mathrm{diag}(\frac{\lambda}{\sigma_i^2+\lambda})V^Tw^*\|_2^2$ (true value w^*), $Var_D(\hat f_D)=\sigma_\varepsilon^2\sum_{i=1}^d\frac{\sigma_i^2}{(\sigma_i^2+\lambda)^2}$

$\Rightarrow\lambda\to0$: $\begin{cases}Bias=0\\Var=\sum_i^d\frac{\sigma_\varepsilon^2}{\sigma_i^2}\end{cases}$, $\lambda\to\infty$: $\begin{cases}Bias\to\|w^*\|_2^2\\Var\to0\end{cases}\Rightarrow\begin{cases}\lambda \text{ large}\\ \Rightarrow B\uparrow, V\downarrow\end{cases}$

Choice of λ: GE reaches min for a λ^*: find λ^* with CV

Remark: $\|w\|_2\le\|w\|_1\le\sqrt{\|w\|_0}\,\|w\|_2$

Classification: labels are discrete, Binary Cl.:

Binary Cl: $f:\mathbb{R}^n\to\{-1,1\}$, if $\hat f:\mathbb{R}^n\to\mathbb{R}$ (Regr.) $\Rightarrow\hat y=sign(\hat f(x))$ (Classif.)

0-1-ℓ: $\ell_{0-1}(\hat y,y)=\mathbb{1}_{\hat y\ne y}=\begin{cases}1&\hat y\ne y\\0&\hat y=y\end{cases}=\mathbb{1}_{\hat y\cdot y<0}$ non-cont. & non-convex

↳ Decision Boundary: $\{x\in\mathbb{R}^d:\hat f(x)=0\}$

Featurized lin Classif.: $\phi:\mathbb{R}^d\to\mathbb{R}^p$, $\mathcal F_\phi=\{f(x)=w^T\phi(x)\}$

↳ Decision Boundary: $\{x\in\mathbb{R}^d:w^T\phi(x)=0\}$

Surrogate Loss fct for training:

Surrogate ℓ for train. & ℓ_{0-1} for val: ℓ_{0-1} non convex & non diff

Accuracy (used for D_{val}): $\frac{1}{|D|}\sum_{(x,y)\in D}\ell_{0-1}(sign(f(x)),y)$. Surrogate loss should be: convex, differentiable, only depend on $z:=y\cdot f_w(x)$, decreasing & bound from below ($\ell\ge0$) $\Rightarrow\ell_{exp}(z)=e^{-z}$, $\ell_{log}(z)=\log(1+e^{-z})$

$\frac{\partial}{\partial z}\ell_{exp}$ is unbounded but $\frac{\partial}{\partial z}\ell_{log}\in(-1,-\frac12)\Rightarrow\ell_{log}$ better

Logistic Regression: $\mathcal{L}(w)=\frac1n\sum_{i=1}^n\log(1+e^{-y_i\cdot w^T\phi(x_i)})$

Max-Margin Solution & Logistic Regression:

ℓ_{log} is convex & diff. $\Rightarrow$ possible to find minimizer $\hat w$ if(!) the data is linearly separable. $\Rightarrow\mathcal{L}\to0$ but $\hat w$ not unique

Max Margin Sol: $\hat w_{MM}=\arg\max_{\|w\|_2=1}margin(w)=\arg\max_{\|w\|_2=1}\min_{1\le i\le n}y_i\langle\hat w,x_i\rangle$

Hard SVM: $\hat w_{SVM}=\left(\min_{w\in\mathbb{R}^d}\|w\|_2 \text{ s.t. } y_i\langle\hat w,x_i\rangle\ge1\ \forall i=1,\dots,n\right)$

Remarks: (i) $y\langle\vec w,\vec x\rangle=|\langle\vec w,\vec x\rangle|=\frac{|\langle\vec w,\vec x\rangle|}{\|w\|_2}=\|x\|_2\cos(\theta)$ (correct class., $\|w\|_2=1$, $\theta=\measuredangle(\vec w,\vec x)$)

(ii) For linearly separable D: $w_{MM}=\frac{w_{SVM}}{\|w_{SVM}\|_2}$

(iii) For linearly sep. D: $\mathcal{L}_{logreg}(w)$ convex & GD with $\eta=1$ satisfies $\lim_{t\to\infty}\frac{w^t}{\|w^t\|_2}=\vec w_{MM}$ $\quad margin(\hat w)=\frac{1}{\|\hat w\|}$

Soft-Margin-Sol: D not linearly sep ($\exists_i: y_i\langle\vec w,\vec x_i\rangle<0$ & $\ker(X)=\emptyset$) $\Rightarrow\exists$ global minimizer $\hat w$ of $\min_w\mathcal{L}_{linreg}(w)$

GD with $\eta=\frac{4}{\lambda_{max}(X)}$ satisfies $\lim_{t\to\infty}w^t=\hat w$

Def $\vec\xi=(\xi_1,\dots,\xi_n)\in\mathbb{R}^n\Rightarrow$ Require $y_i\langle\vec w,\vec x_i\rangle\ge1-\xi_i$

Optim Problem: $\min_{w\in\mathbb{R}^d,\xi\in\mathbb{R}^n}\left(\|w\|_2^2+\lambda\sum_{i=1}^n\xi_i\right)$ s.t. $\begin{cases}y_i\langle\vec w,\vec x_i\rangle\ge1-\xi_i\\\xi_i\ge0\ \forall i\end{cases}$

$=\min_{w\in\mathbb{R}^d}\left[\|w\|_2^2+\lambda\sum_{i=1}^n\underbrace{\max(0,1-y_i\langle\vec w,x_i\rangle)}_{\text{hinge loss}}\right]$

Remarks: $\#\{\xi_i>0\}=\#\{$wrongly classif pts$\}+\#\{x: x^Tw\in[-1,1]\}$

$\xi_i=0\ \forall$ correctly classified pts outside $x^Tw=\pm1$ margin

Multiclass Classif.: Train K diff. linear Models

$\hat f_1,\dots,\hat f_K\in\mathcal F_{lin}$ $\forall$Cl. $\Rightarrow\hat y(\vec x)=\arg\max_{1\le k\le K}\hat f_k(\vec x)$

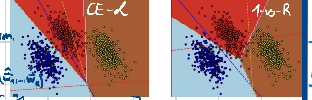
1 vs Rest: Train K binary Classifiers: $\hat y=\begin{cases}-1&y=k\\1&y\ne k\end{cases}$

Cross-Entropy: fit $\hat f_1,\dots,\hat f_K$ simultaneously ($\vec f(x)=(f_1(x),\dots,f_K(x))$)

$\ell_{CE}(\vec f(x),y)=-\log\left(\frac{e^{f_y(x)}}{\sum_{i=1}^Ke^{f_i(x)}}\right)\Rightarrow\hat{\vec f}=\arg\min_{f\in\mathcal F}\left[\sum_{i=1}^n\ell_{CE}(\vec f(x_i),y_i)\right]$

If f_i can be param. by $\vec w_i\in\mathbb{R}^d$, def $\hat w=(\hat w_1,\dots,\hat w_K)$

↳ $\hat w=\arg\min_{w_1,\dots,w_K}\left[\sum_{i=1}^n\ell_{CE}(f_w(\vec x_i),y_i)\right]$



Approx of GE: $GE\approx\frac{1}{|D_{test}|}\sum_{(x,y)\in D_{test}}\mathbb{1}_{\hat y(x)\ne y}$

Other evaluation Metrics for Classifiers.

Assign the "important" Class to -1 & other Cl. +1

H_0: $\hat y(x)=-1$, H_1: $\hat y(x)=1$

	$y=1$	$y=-1$
$\hat y=1$	TP	FP (T1 err)
$\hat y=-1$	FN (T2 err)	TN

If FN & FP not equally severe: $GE=C_{FN}\frac{\#FN}{\#\{y=+1\}}+C_{FP}\frac{\#FP}{\#\{y=-1\}}$

Minimizing this error only depends on $\delta=\frac{C_{FN}}{C_{FP}}\Rightarrow$ Usually $\delta>1\Rightarrow\{FP\downarrow, FN\uparrow\}$

$FNR=\frac{\#FN}{\#\{y=+1\}}=P(\hat y(x)=-1\,|\,y(x)=1)$, $FOR=\frac{\#FP}{\#\{\hat y=+1\}}$

$FPR=\frac{\#FP}{\#\{y=-1\}}=P(\hat y(x)=1\,|\,y(x)=-1)$, $Accuracy=\frac{\#TP+\#TN}{n}$

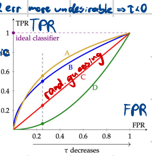
TPR = Sensitivity = power = recall $=\frac{\#TP}{\#\{y=+1\}}$ (1-FOR)

TNR = Specificity $=\frac{\#TN}{\#\{y=-1\}}$, Precision $=\frac{\#TP}{\#\{\hat y=+1\}}$

F1-Score $=\frac{2}{\frac{1}{recall}+\frac{1}{precision}}$, want large, want small

$\hat y_\tau(x)=sign(\hat f(x)-\tau)=\begin{cases}+1&\text{if }\hat f(x)>\tau\\-1&\text{if }\hat f(x)<\tau\end{cases}$. T1 err more severe $\Rightarrow\tau>0$, T2 err more undesirable $\Rightarrow\tau<0$

ROC Curve: Receiver Operating Characteristic



AreaUnderTheROCCurve (AUROC): measure of Classifier Perform. $\in[0,1]\Rightarrow\begin{cases}\text{rand.Cl.}: \frac12\\\text{perf.Cl.}: 1\end{cases}$

Distribution Shifts & Robustness: Minority group: $\max\left\{\frac{1}{|g_1|}\sum_{(x,y)\in g_1}\mathbb{1}_{\hat y\ne y},\frac{1}{|g_2|}\sum_{(x,y)\in g_2}\mathbb{1}_{\hat y\ne y}\right\}$

Fairness: Equal Odds: $\forall y,s,s'\in\{-1,1\}: P(\hat y=1|Y=y,S=s)=P(\hat y=1|Y=y,S=s')$ (→ All groups same FPR & FNR)

Demographic Parity: $\forall s,s'\in\{-1,1\}: P(\hat y=1|S=s)=P(\hat y=1|S=s')$ (→ P of pos pred doesn't vary between groups)

Equal Opportunity: $P(\hat y=1|S=s,Y=1)=P(\hat y=1|S=s',Y=1)$ (→ Only demand: every group has same P to get $\hat y=1|y=1$)

Remark: Equal Op. & Dem. Parity: accuracy-fairness-tradeoff

Kernels:

Kernelization: Assume $\vec w=\Phi^T\alpha\in\mathbb{R}^p$, $\alpha\in\mathbb{R}^n$

$\hat w\leadsto\hat\alpha=\min_{\alpha\in\mathbb{R}^n}\frac1n\sum_{i=1}^n\ell(y_i,\sum_{j=1}^n\alpha_jk(x_i,x_j))=\min_{\alpha\in\mathbb{R}^n}\frac1n\sum_{i=1}^n\ell(y_i,(K\alpha)_i)$ ($O(p\cdot n)\gg O(n^2)$ if $p\gg n$)

Replace $\langle\phi(x_i),\phi(x_j)\rangle=k(x_i,x_j)$, Def $K\in\mathbb{R}^{n\times n}$, $K_{ij}=k(x_i,x_j)$

Valid Kernel: (i) Symmetric: $k(x,x')=k(x',x)$ (ii) K p.s.d.

Thm: 1) $\phi:\mathbb{R}^d\to\mathbb{R}^p$, $\tau:\mathbb{R}^p\to\mathbb{R}^q\Rightarrow k(x,x')=\langle\tau(\phi(x)),\tau(\phi(x'))\rangle$ valid ker.

2) k_1,k_2 valid $\Rightarrow k(x,x')=k_1(x,x')+k_2(x,x')$ & $k((x,y),(x',y'))=k_1+k_2$ valid

3) k_1,k_2 valid $\Rightarrow k(x,x')=k_1(x,x')\cdot k_2(x,x')$ & $k((x,y),(x',y'))=k_1\cdot k_2$ valid

Validity Check: 1) Symmetry, 2) valid comb., 3) easily not psd, 4) feature map

Examples: g Taylor series with coeff ≥0: $k(x,x')=g(\langle x,x'\rangle)$ valid

Polynomial Kernel: $k(x,x')=(1+\langle x,x'\rangle)^m\Rightarrow\phi(x)\in\mathbb{R}^p$ ($x\in\mathbb{R}^d$), $p\sim\begin{cases}O(d^m)&\text{bad } d\to\infty\\O(m^d)&m\to\infty\end{cases}$

Gaussian Kernel: $k(x,x')=\exp\left(-\frac{\|x-x'\|_p^\alpha}{\tau}\right)$ ($\mathcal H$ is ∞-dim), τ bandwidth, $\alpha=2$ & $p=\begin{cases}2: \text{Gauss}\\1: \text{Lapl.}\end{cases}$

Thm: $\mathcal{L}$ strongly convex & smooth, $\hat w=\lim_{t\to\infty}w^t\in\{\sum_{i=1}^n\alpha_i\phi(x_i):\alpha\in\mathbb{R}^n\}\subset\mathbb{R}^p$

$\Rightarrow\exists\hat\alpha\in\mathbb{R}^n:\hat w=\Phi\hat\alpha$, $\hat\alpha$ global minimizer of kernelized obj. fct.

In practice: Kernelized OLS: $\min_{w\in\mathbb{R}^d}\frac1n\|y-Xw\|_2^2\to\min_{\alpha\in\mathbb{R}^n}\frac1n\|y-XX^T\alpha\|_2^2$ ($K=XX^T\to\hat\alpha=K^{-1}y$, $\hat w=X^T\hat\alpha=X^TKy$)

Kern. Ridge Reg: $\frac1n\|y-Xw\|_2^2+\lambda\|w\|_2^2=\frac1n\|y-XX^T\alpha\|_2^2+\lambda\alpha^TXX^T\alpha=\frac1n\|y-K\alpha\|_2^2+\lambda\alpha^TK\alpha$ ($K=XX^T$ $\Rightarrow\alpha^*=(K+\lambda\mathbb{1})^{-1}y$ $\Rightarrow y_{new}=K_{new}^T\alpha$)

Classification: $\ell(y_i,\hat f(x_i))=\log(1+e^{-y_i\langle\vec w,\vec x_i\rangle})\Rightarrow\hat\alpha=\arg\min_{\alpha\in\mathbb{R}^n}\frac1n\sum_{i=1}^n\log\left(1+\exp\left(-y_i\sum_{j=1}^n\alpha_jk(x_i,x_j)\right)\right)$

↳ $\hat y(x)=sign\left(\sum_{j=1}^n\hat\alpha_jk(x,x_j)\right)$

Remarks: False: $\forall k\ \exists\phi: k=\langle\phi(.),\phi(.)\rangle$,

$k=\frac{\max}{\min}$ not valid, k can be negative, $k=\min\ \forall x,y>0$ valid, $k=\max$ not valid

$k=\frac{\min}{\max}\ \forall x,y>0$ valid, $k=\cos(x-y)$ valid, $k=(x^Tx-1)^2$ not valid

$k=|A\cap B|$ valid, $k=2|A\cap B|-|A|-|B|$ not valid

Neural Networks

Idea: learn φ as well $\to\phi_j(x,\theta)=\varphi(\theta^Tx)$

with φ a non linear activation fct, e.g.: $\varphi_{id}(z)=z$, $\varphi_{sigm}(z)=\frac{1}{1+e^{-z}}$, $\varphi_{tanh}(z)=\tanh(z)=\frac{e^z-e^{-z}}{e^z+e^{-z}}$, $\varphi_{ReLU}(z)=\max(0,z)$

Thm: $f: S\to\mathbb{R}$ ($S\subset\mathbb{R}^n$ & compact) cont. fct & φ any sigmoidal fct ($\sigma(t\to\infty)=1$, $\sigma(t\to-\infty)=0$). Then f can be uniformally approx by $\hat f(x)=\sum_{j=1}^mw_j^{(2)}\varphi((w_j^{(1)})^Tx+w_{j0}^{(1)})$

Backward propagation: Optim Problem is non-convex! → Grad based M.

Optimization in a general NN:

Setup: $W=[W^{(1)},\dots,W^{(L)}]\Rightarrow$ Goal: $\vec\nabla_W\ell=(\vec\nabla_{W^{(1)}}\ell,\dots,\vec\nabla_{W^{(L)}}\ell)$

Def: $\delta:=2(f(x;W)-y)$, $z^{(\ell)}=W^{(\ell)}v^{(\ell-1)}$, $v^{(\ell)}=\varphi(z^{(\ell)})$

1) $(\vec\nabla_{W^{(L)}}\ell)^T=\frac{\partial\ell}{\partial W^{(L)}}=\frac{\partial\ell}{\partial f}\odot\frac{\partial f}{\partial W^{(L)}}$ (Note: $\begin{bmatrix}\alpha_1\\\vdots\\\alpha_K\end{bmatrix}\odot\begin{bmatrix}\tilde x_1^T\\\vdots\\\tilde x_K^T\end{bmatrix}=\begin{bmatrix}\alpha_1\tilde x_1^T\\\vdots\\\alpha_K\tilde x_K^T\end{bmatrix}\in\mathbb{R}^{K\times m}$)

Note: $\frac{\partial f}{\partial W}=\frac{\partial f}{\partial vec(W)}=\begin{bmatrix}\partial f_1/\partial w_{11}&\partial f_1/\partial w_{12}&\cdots&\partial f_1/\partial w_{Km}\\\vdots&&&\vdots\\\partial f_K/\partial w_{11}&\partial f_K/\partial w_{12}&\cdots&\partial f_K/\partial w_{Km}\end{bmatrix}\in\mathbb{R}^{K\times Km}\Rightarrow\begin{bmatrix}v^{(L-1)}\\\vdots\\v^{(L-1)}\end{bmatrix}\in\mathbb{R}^{K\times m}$ ($W\in\mathbb{R}^{K\times m}$)

↳ $f=W^{(L)}v^{(L-1)}\Rightarrow\frac{\partial f}{\partial W^{(L)}}=\begin{bmatrix}\vec v^{(L-1)}\\\vdots\\\vec v^{(L-1)}\end{bmatrix}\in\mathbb{R}^{K\times n_{L-1}}$ Note: $\vec v^{(L-1)}$ calculated in Forw. pass

↳ $(\vec\nabla_{W^{(L)}}\ell)^T=\frac{\partial\ell}{\partial W^{(L)}}=\delta\cdot\begin{bmatrix}\vec v^{(L-1)}\\\vdots\\\vec v^{(L-1)}\end{bmatrix}$

2) $(\vec\nabla_{W^{(L-1)}}\ell)^T=\frac{\partial\ell}{\partial f}\cdot\frac{\partial f}{\partial v^{(L-1)}}\cdot\frac{\partial v^{(L-1)}}{\partial z^{(L-1)}}\cdot\frac{\partial z^{(L-1)}}{\partial W^{(L-1)}}=\delta\cdot W^{(L)}\cdot\frac{\partial v^{(L-1)}}{\partial z^{(L-1)}}\cdot\begin{bmatrix}v^{(L-2)}\\\vdots\\v^{(L-2)}\end{bmatrix}$

$\frac{\partial\vec v^{(L-1)}}{\partial z^{(L-1)}}=\begin{bmatrix}\varphi'(z_1^{(L-1)})&0&\cdots&0\\0&\varphi'(z_2^{(L-1)})&&\vdots\\\vdots&\cdots&\ddots&\varphi'(z_{n_{L-1}}^{(L-1)})\end{bmatrix}$

3) $(\vec\nabla_{W^{(L-2)}}\ell)^T,\dots,(\vec\nabla_{W^{(1)}}\ell)^T=?$ Note: $\frac{\partial z^{(\ell+1)}}{\partial z^{(\ell)}}=\frac{\partial z^{(\ell+1)}}{\partial v^{(\ell)}}\cdot\frac{\partial v^{(\ell)}}{\partial z^{(\ell)}}$

$\Rightarrow(\vec\nabla_{W^{(\ell)}}\ell)^T=\frac{\partial\ell}{\partial W^{(\ell)}}=\frac{\partial\ell}{\partial f}\cdot\frac{\partial f}{\partial z^{(L-1)}}\cdot\frac{\partial z^{(L-1)}}{\partial z^{(L-2)}}\cdot\ldots\cdot\frac{\partial z^{(\ell+1)}}{\partial z^{(\ell)}}\frac{\partial z^{(\ell)}}{\partial W^{(\ell)}}$

Optimization & training techniques: Convergence depends on initial. SGD better than GD bec. it can escape local min.

Vanishing & Exploding Grad.: Optim fails if $\|\vec\nabla_w\mathcal{L}\|_2\to\begin{cases}\infty\\0\end{cases}\Rightarrow$ control $\varphi'(z)$

$\sigma(z)=\frac{1}{1+e^{-z}}\Rightarrow\sigma'(z)=\frac{e^{-z}}{(1+e^{-z})^2}$ $\begin{cases}⊕\text{ easy to compute}\\⊖\text{ only good if } z\approx0\end{cases}$, $ReLU(z)=\max(0,z)\to\varphi'(z)=\begin{cases}0&z<0\\1&z>0\end{cases}$ $\begin{cases}⊕\text{ easy der.}\\⊖\text{ not def @ } z=0\end{cases}$

Random Weight init: For 1 layer NN ideal: $w\sim\mathcal N(0,\frac2d)$ [d = input dim]

Heuristics: 1) Xavier Init: $w_i\sim\mathcal N(0,\frac{1}{n_{in}})$ or $w_i\sim\mathcal N(0,\frac{2}{n_{in}+n_{out}})$ (good if $\varphi(z)=\tanh(z)$), 2) He et al: $w_i\sim\mathcal N(0,\frac{2}{n_{in}})$ (good if $\varphi(z)=ReLU(z)$)

Learning Rate & weight optim: Ideally: $\eta_t\xrightarrow{t\to\infty}0$, better: $\frac{1}{\eta_t}\propto\frac{|\vec\nabla_w\mathcal{L}(w^t;D)|}{\|w^t\|_2}$

Regularization in NNs: 1) Penalty term: $\arg\min_w\sum_{i=1}^n\ell(W;x_i,y_i)+\lambda\|w\|_2^2$

2) Early Stopping: e.g. Stop if val loss increases in training

3) Dropout: Randomly omit units with prob. 1-p in each iteration of SGD

↳ Train remaining subnetwork $\Rightarrow$ For testing use all units, adjust $z_j^{(\ell)}=p\cdot\vec w_j^{(\ell-1)}\cdot v^{(\ell-1)}$

4) Batch Normalization: Correct for mean drift → Standardize batches (not the units!)

Algor: 1) $B:=\{v_{i_1},\dots,v_{i_k}\}$ for 1 unit ($S=\{i_1,\dots,i_k\}$ subset of I), 2) learnable params: β,γ, 3) $v_j\mapsto\bar v_j=BN(v_j;\dots)$

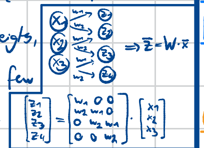
$\mu_S=\frac{1}{|S|}\sum_{j\in S}v_j$, $\sigma_S^2=\frac{1}{|S|}\sum_{j\in S}(v_j-\mu_S)^2\Rightarrow\hat v_j=\frac{v_j-\mu_S}{\sqrt{\sigma_S^2+\varepsilon}}$, $\varepsilon>0\Rightarrow\bar v_j=\gamma\hat v_j+\beta$

4) Output $\bar v_j$ $\forall j\in S$

Convolutional Neural Networks

Images can be represented as {$n\times m$ matrix if greyscale; $n\times m\times 3$ tensor if color}

Idea: Units share same weights, & each unit connected to few units in the next layer



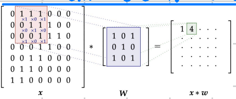
$\Rightarrow z = W\cdot x$

1D Conv: $w\in\mathbb{R}^k$, $x\in\mathbb{R}^d \Rightarrow w * x = z \in \mathbb{R}^{k+d-1}$, $z_j = \sum_{i=\max\{1,j-k+1\}}^{\min\{j,d\}} w_i x_{j-i+1}$

CNN: ANN with $v^{(l)} = \varphi(w^{(l)} * v^{(l-1)})$ layers, but may also contain fully connected layers

Remark: Weights are called filters.

Multidimensional Convolution:



Consider $n\times m\times 3$ RGB Image with m filters of size $f\times g$.

Padding: Add a frame (couple of white rows/cols) to the image $\Rightarrow$ new dim: $(n+2p)\times(m+2p)$

Stride: Stepsize of filter. Must match dim of filter: $(n+2p-f)\,\%\, s \overset{!}{=} 0$, $\frac{n+2p-f}{s}$ if ...

Output Dim: $\left(\frac{n+2p-f}{s}+1\right)\times\left(\frac{m+2p-g}{s}+1\right)\times m$

Clustering (Unsupervised Learning)

Unsupervised learning: learning without y_i's.

Goal: extract meaningful patterns. {Dim. Red $\leftrightarrow$ Regression; Clustering $\leftrightarrow$ Classific.}

Standard Clustering: group D into clusters based on similarity

Hierarchical Clust.: 1) Form individual Clusters $\forall$ D-points 2) Iteratively merge pairs of clusters that are closest

K-Means Clustering & Lloyd's Heuristic

each K-means iter. requir. amount of memory polynomial in n, d, K

Each Cluster represented by its center $\mu_j \in \mathbb{R}^d$

each K-means iter. performs nbr of additions & multiplications polynomial in n,d,K

Def: Clusterassignment: $z_i = \arg\min_{j\in\{1,\dots,K\}} \|x_i - \mu_j\|_2$

Goal: learn μ_j in training $\to$ assign z_i in testing based on μ_j

$\mu = (\mu_1,\dots,\mu_k)^T = \arg\min \sum_{i=1}^n \min_j \|x_i-\mu_j\|_2^2$ non-convex & np-hard

Lloyd's Heuristic: Iteratively update z_i & $\mu_j = \frac{1}{n_j}\sum x_i$

Def: $z^{(t)} = (z_1^{(t)},\dots)$, $\mu^{(t)} = (\mu_1^{(t)},\dots)$, $n_j^{(t)} = \#\{i : z_i = j\}$

1) Init $\mu^{(0)}$, 2) Repeat: $z_i^{(t)} = \arg\min_{j\in\{1..K\}} \|x_i - \mu_j^{(t-1)}\|_2$, $\mu_j^{(t)} = \frac{1}{n_j^{(t)}}\sum_{i: z_i=j} x_i$ $\forall j = 1..K$, $t\leftarrow t+1$ until conv.

Convergence Analysis & Limitations:

Thm: $\hat{R}(\mu,z) = \sum_{i=1}^n \|x_i - \mu_{z_i}\|_2^2 \Rightarrow \hat{R}(\mu^{(t)},z^{(t)}) \ge \hat{R}(\mu^{(t)},z^{(t+1)}) \ge \hat{R}(\mu^{(t+1)},z^{(t+1)})$

↳ Lloyd's Algorithm guaranteed to converge.

But only to local min. ⊖: Strongly dep. on init $\mu^{(0)}$; Selecting K open problem; Can't model clusters of non-spherical shape

Init Schemes & K-Means++:

Unif. Distr.: $p_j(x_i) = \frac{1}{n}$ → ⊖ bad for unbalanced cl.

Furthest point: $\mu_1^{(0)} \sim \mathcal{U}(D) \Rightarrow \mu_{j+1} = \arg\max_{x_j\in D} \|x_j - \mu_m\|$ (furthest from all μ) ⊖ sensitive to outliers

K-Means++: 1) $\mu_1^{(0)} = x_i \in D$: $i\sim\mathcal{U}(\{1,\dots,n\})$, 2) $\forall j = 2,\dots,K$: $\mu_j = x_i \in D$, $i\sim p(i) \propto \min_{m\in\{1,\dots,j-1\}} \|x_i - \mu_m\|_2^2$ (prob ∝ dist to nearest cl. center → far away w. high prob.)

Choosing K: Increasing k always yields better results!

Heuristics: Elbow Heur., Adding Regul.: $\hat{R}(\cdot) + \lambda\cdot K$

PCA

Goal: $x_1,\dots,x_n \in \mathbb{R}^d \leadsto z_1,\dots,z_n \in \mathbb{R}^k$, $k \ll n$

k=1: wlog $\bar{x}_i = 0\ \forall i$. Goal: $\forall x\in\mathbb{R}^d$ find $z\in\mathbb{R}$ & $w\in\mathbb{R}^d$: $x \approx w z$

Minimize reconstr. err.: $\min_{z,w} \|x - zw\|_2^2$ s.t. $\|w\|_2 = 1$

$w^*, z^* = \arg\min_{\|w\|_2=1, z\in\mathbb{R}^n} \sum \|z_i w - x_i\|_2^2 \Rightarrow w = \arg\min \dots$, $z_i = w^T x_i$

↳ $w^* = \arg\max_{\|w\|_2=1} \sum (w^T x_i)^2 = \arg\max w^T \Sigma w$, $\Sigma = \frac{1}{n}X^T X$

Σ p.s.d & sym $\Rightarrow \Sigma = \sum_{i=1}^d \lambda_i v_i v_i^T$, $\lambda_1 \ge \dots \ge \lambda_d \ge 0$ EV

$w^* = \arg\max_{\|w\|_2=1} w^T\Sigma w = v_1$ [$\max \sum \alpha_i^2 \lambda_i$ & $\sum \alpha_i^2 = 1 \to \alpha_1 = 1$]

Remarks: w PCA $\to -w$ also sol., If $\lambda_1 = \dots = \lambda_m > \lambda_{m+1} \ge \dots$: Any convex comb. of v_i's also optim. $\Rightarrow$ not unique

$1\le k\le d$: $W, z_1,\dots,z_n^* = \arg\min_{W\in\mathbb{R}^{d\times k}, W^TW = I_k, z\in\mathbb{R}^k} \sum \|x_i - W z_i\|_2^2$

↳ $W^* = [v_1|\dots|v_k]$, $z_i^* = W^T x_i$ $\Rightarrow \lambda_j$ = {empirical var along j-th PC; var explained by j-th PC}

Optimal k: Use CV. Heuristic: k st 95% of var explained

SVD: $X = USV^T \to n\cdot\Sigma = VDV^T \to$ first k EV of Σ = first k SV of X → first k cols of V

PCA & K-Means: Both Matrix factor. Probl.

PCA: $W^*, z_1^*,\dots,z_n^* = \arg\min_{W\in\mathbb{R}^{d\times k}: W^TW = I_k,\ z\in\mathbb{R}^k} \sum \|x_i - Wz_i\|_2^2$

k-M: $W^*, z_1^*,\dots,z_n^* = \arg\min_{W\in\mathbb{R}^{d\times k},\ z_1,\dots,z_n\in E_k} \sum \|x_i - Wz_i\|_2^2$, $E_k = \{e_1,\dots,e_k\}$

Kernel PCA

$x\in\mathbb{R}^d \mapsto \varphi(x)\in\mathbb{R}^p \mapsto z = W^T\varphi(x)$, $W\in\mathbb{R}^{p\times k}$

$\Rightarrow w = \sum \alpha_i\varphi(x_i) \Rightarrow \alpha^* = \arg\max_{\alpha\in\mathbb{R}^n} [\alpha^T K^T K \alpha]$ s.t. $\alpha^T K\alpha = 1$

$\Rightarrow \alpha^{(i)} = \frac{1}{\sqrt{\lambda_i}} v_i$ for k=1..., $z = \sum_j \alpha_j^{(i)} k(x, x_j)$

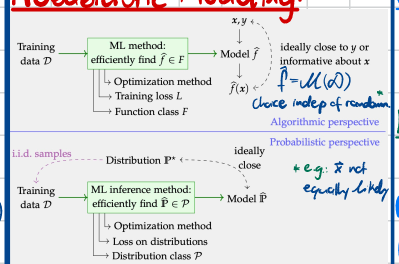
Autoencoders

Encoder: $x \mapsto z = W^T x$ ($\mathbb{R}^d \to \mathbb{R}^k$), Decoder: $z \mapsto \hat{x} = Wz$

Goal: find subspace minimizing reconstr. error: $\min_W \sum \|x_i - f(x_i;W)\|_2^2$

Idea: learn Identity fct: $x \approx f(x;\theta) = f_{dec}(f_{enc}(x;\theta_{enc});\theta_{dec})$

Probabilistic Modeling



$\hat{f} = \mathcal{U}(D)$, choice indep of randomness. + e.g. $\hat{x}$ not equally likely

$D = \{(x_1,y_1),\dots,(x_n,y_n)\} \overset{iid}{\sim} P_{X,Y} \Rightarrow P_D = \prod P_{x_i,y_i}$

Histogram: $\hat{p}_h(x) = \frac{1}{n\cdot h}\sum_{i=1}^n 1\{x_i \in B_k\}$ (h: bin width) ⊕ no implicit assumptions on P; ⊖ unknown, needs large |D|

Kernel Density Estimator: $\hat{p}_{KDE}(x) = \frac{\sum k(x,x_i)}{\int \sum k(x_i,x)\,dx}$

Statistical Inference:

Paradigms: Frequentist vs Bayes. Bayes Viewpoint:

Strong prior belief $\leadsto$ (model) θ^* unobserved rand. var. $\sim P_\theta$

$p(\theta|D) = \frac{p(D|\theta)p_\theta(\theta)}{p(D)}$, $p(D) = \int p(D|\theta)p_\theta(\theta)\,d\theta$

⊕ Incorporate prior belief, precision known even if only small sample

⊖ prior info can lead to bias, Computationally hard

Maximum Likelihood Estimation:

$D = \{(x_i,y_i)\}$, $(x_i,y_i) \overset{iid}{\sim} P^\theta$, $\theta\in\Theta$

$\hat{\theta}_{MLE} = \arg\max_{\theta\in\Theta} p(D|\theta) \overset{iid}{=} \arg\max \prod p(x_i,y_i;\theta) = \arg\min_{\theta\in\Theta} \sum -\log(p(x_i,y_i;\theta))$

Def: KL Divergence: measure of "difference" between ref distr. P & another distr. Q:

$D_{KL}(P\|Q) = \mathbb{E}_{x\sim P}\left[\log\left(\frac{p(x)}{q(x)}\right)\right] = \int p(x)\log\left(\frac{p(x)}{q(x)}\right)dx$, $=0$ iff $P=Q$

Remarks: $D_{KL}(P\|Q) \ne D_{KL}(Q\|P)$, $D_{KL} \ge 0$

Maximum A Posteriori Estimation:

$\hat{\theta}_{MAP} = \arg\max_{\theta\in\Theta} p(\theta|D) = \arg\max p(D|\theta)p(\theta) \overset{iid}{=} \arg\max p(\theta)\prod p(x_i,y_i|\theta) = \arg\min \sum -\log(p(x_i,y_i|\theta)) - \log(p(\theta))$

If $\theta\sim\mathcal{U}(\Theta) \Rightarrow p(\theta) = const \Rightarrow \hat{\theta}_{MAP} = \hat{\theta}_{MLE}$

Bayes Optimal Predictors:

Goal: estim $Y|X=x$

$f^*(x) = \arg\min_{a\in\mathcal{Y}} \mathbb{E}[\ell(a,Y)|X=x] = \arg\min_{a\in\mathcal{Y}} \int p(y|x)\ell(a,y)\,dy$

Probabilistic Regression:

Regr. w/ MLE:

Gaussian Noise: $y = f(x;\theta) + \varepsilon$, $\varepsilon \perp x$, $\varepsilon\sim\mathcal{N}(0,\sigma^2)$ (fixed & known)

↳ $Y|X=x \sim \mathcal{N}(f(x;\theta),\sigma^2) \Rightarrow \hat{\theta}_{MLE} = \arg\min_{\theta} \sum (y_i - f(x_i;\theta))^2$

Heavy Tailed Noise: $\varepsilon\sim$ student t, $\nu,\sigma > 0$

$p(y|x;\theta) = \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\pi\nu}\,\Gamma(\frac{\nu}{2})}\left(1 + \frac{(y-f(x;\theta))^2}{\nu\sigma^2}\right)^{-\frac{\nu+1}{2}}$

↳ $\hat{\theta}_{MLE} = \arg\min_\theta \sum \log\left(1 + \frac{(y_i - f(x_i;\theta))^2}{\nu\sigma^2}\right) \Rightarrow$ regr. with $\ell = \log(1+\dots)$

Regr w/ MAP: Gaussian Prior: $\theta^* \sim \mathcal{N}(0,\sigma_p^2 I)$

$\hat{\theta}_{MAP} = \arg\min_\theta \sum (y_i - f(x_i;\theta))^2 + \frac{\sigma^2}{\sigma_p^2}\|\theta\|_2^2$

↳ Ridge Reg with $\lambda = \frac{\sigma^2}{\sigma_p^2} \to$ {large: D noisy; small: D accurate}

Laplace Prior: $\theta_i \sim Laplace(0,b) \Rightarrow p(\theta) = \frac{1}{2b}e^{-\frac{|\theta|}{b}}$ ($Var(\theta) = 2b^2$)

$\hat{\theta}_{MAP} = \arg\min_\theta \sum (y_i - f(x_i;\theta))^2 + \frac{2\sigma^2}{b}\|\theta\|_1 \Rightarrow$ Lasso with $\lambda = \frac{2\sigma^2}{b}$

Bay. Opt. Cl. using ℓ_{sq}: $f^*(x) = \mathbb{E}[Y|X=x]$

Remarks: MAP w/ Gaussians: $P_{Y_i} = \mathcal{N}(\mu,1)$, prior: $\mu\sim P_\mu = \mathcal{N}(\mu_0,1) \Rightarrow$ Posterior: $P_{\mu|D} = \mathcal{N}\left(\frac{\sum x_i + \mu_0}{n+1}, \frac{1}{n+1}\right)$

Probabilistic Classification: Discrim. & Gener.

Discriminative Model: model cond. prob $Y|X$ ($P_{Y|X}$); $p(x,y) = p(y|x)p(x)$

Generative Model: Model $P_{X,Y}$ by $P_{X|Y}$ & P_Y

Classification w/ Discriminative Models:

Focus on binary Cl.: $y\in\{-1,1\}$. Assume $y\cdot f(x;\theta)\in F$

Model $p(y|x)$ by $\sigma(z) = \frac{1}{1+e^{-z}}$ for $Y|X=x \sim Ber(\sigma(f(x;\theta)))$

$p(y|x;\theta) = \sigma(y\cdot f(x;\theta)) = \frac{1}{1+e^{-y\cdot f(x;\theta)}} \Rightarrow \hat{y} = \arg\max_{y\in\{\pm1\}} p(y|x;\theta)$

MLE: $\hat{\theta}_{MLE} = \arg\min_{\theta\in\Theta} \sum_{i=1}^n \log(1 + e^{-y_i f(x_i;\theta)}) \Rightarrow$ log reg

MAP: Gauss prior: $\theta^*\sim\mathcal{N}(0,\sigma_p^2 I)$, Laplace Prior: $p(\theta) = \frac{1}{2b}e^{-\frac{|\theta|}{b}}$

GP: $\hat{\theta}_{MAP} = \arg\min \sum \log(1+e^{-y_i f(x_i;\theta)}) + \frac{1}{2\sigma_p^2}\|\theta\|_2^2$ (Ridge w/ $\lambda = \frac{1}{2\sigma^2}$)

LP: $\hat{\theta}_{MAP} = \arg\min \sum \log(1+e^{-y_i f(x_i;\theta)}) + \frac{1}{b}\|\theta\|_1$ (Lasso w/ $\lambda = \frac{1}{b}$)

Classification w/ generative Models:

Aim: $p(y|x) = \frac{p(x,y)}{p(x)} \propto p(x|y)p(y) \Rightarrow$ estim $p(y)$ & $p(x|y)$

Gaussian Naive Bayes Model: $P(Y=y) = p_y$

"Naive": each dim indep.: $p(x_1,\dots,x_d|y) = \prod_{j=1}^d p(x_j|y)$

$X_j|Y=y \sim \mathcal{N}(\mu_{y,j},\sigma_{y,j}^2) \Rightarrow \Sigma_y = diag(\sigma_{y,1}^2,\dots)$

MLE: $\hat{p}_y = \frac{n_y}{n}$, $\hat{\mu}_{y,j} = \frac{1}{n_y}\sum x_{i,j}$, $\hat{\sigma}_{y,j}^2 = \frac{1}{n_y}\sum (x_{i,j} - \hat{\mu}_{y,j})^2$

↳ $\hat{y} = \arg\max_y \left[\log(\hat{p}_y) - \frac{1}{2}\sum_{j=1}^d \left(\log(\hat{\sigma}_{y,j}^2) + \frac{(x_j - \hat{\mu}_{y,j})^2}{\hat{\sigma}_{y,j}^2}\right)\right]$

↳ Binary Cl.: $\hat{y} = sign\left(\log\left(\frac{p(x|+1)}{p(x|-1)}\right)\right)$

Fischer's LDA: (F) LDA: $\Sigma_{+1} = \Sigma_{-1}$, ($P(Y=1) = 0.5$)

Special Case of GNB: $\sigma_{+,j}^2 = \sigma_{-,j}^2 = \sigma_j^2 \Rightarrow \hat{y} = sign(w^T x + w_0)$

$f(x) = \log\left(\frac{p(+1|x)}{p(-1|x)}\right) = \sum_{j=1}^d \dots$

Bayes Opt. Cl.: $f^*(x) = \arg\min_{a\in\mathcal{Y}} \mathbb{E}[\ell(a,Y)|X=x]$

0-1 L: $\ell_{0-1}(f(x),y) = 1_{f(x)\ne y} = \{0\ f=y;\ 1\ f\ne y\} \Rightarrow \mathbb{E}[\cdot] = 1 - p(a|x)$

↳ $f(x) = \arg\max_{a\in\mathcal{Y}} p(a|x)$

Asym 0-1 L: $\ell(f(x),y) = c_{f(x)}1_{f(x)\ne y} = \{0\ f=y;\ c_{f(x)}\ f\ne y\}$

↳ $\mathbb{E}[\ell(a,Y)|X=x] = (1-p(a|x))c_a \Rightarrow f(x) = \arg\min_a (1-p(a|x))c_a$

Binary Cl.: $\ell(f(x),y) = \{0\ f(x)=y;\ c_{FN}\ f(x)=-1 \ \&\ y=1;\ c_{FP}\ f(x)=1\ \&\ y=-1\}$

$\mathbb{E}[\ell(-1,Y)|X=x] = c_{FN}\cdot P(Y=1|X=x)$, $\mathbb{E}[\ell(1,Y)|X=x] = c_{FP}\cdot P(Y=-1|X=x)$

↳ $f(x) = \{-1\ \ p(1|x)c_{FN} < p(-1|x)c_{FP};\ 1\ \ p(1|x)c_{FN} > p(-1|x)c_{FP}\}$

0-1 L w/ Abstention: $\ell(f(x),y) = \{0\ f(x)=y;\ 1\ f(x)\ne y;\ c\ f(x)=r\}$

↳ $\mathbb{E}[\ell(a,Y)|X=x] = \{1-p(a|x)\ a\in\mathcal{Y};\ c\ a=r\} \Rightarrow f(x) = \{\arg\max p(a|x)$ if $\max p(a|x) > 1-c$; r if $\max p(a|x) \le 1-c\}$

Remark: $c\in[0,\frac{1}{2}]$, K = #possible labels

Binary Cl.: $f^*(x) = \{-1\ \ p(+1|x) < c;\ r\ \ p(+1|x)\in[c,1-c] \Leftrightarrow c < \min\{p(1|x),p(-1|x)\};\ 1\ \ p(+1|x) > 1-c\}$

Gaussian Mixture Models:

Gaussian Bayes Cl. (GDA): $P(Y=y) = p_y$,

$P(x|y) = \mathcal{N}(x;\mu_y,\Sigma_y)$ (Multivariate Gaussian) $\Rightarrow \hat{p}_y = \frac{n_y}{n}$, $\hat{\mu}_y = \frac{1}{n_y}\sum x_i$

$\hat{\Sigma}_y = \frac{1}{n_y}\sum_{i:y_i=y}(x_i - \hat{\mu}_y)(x_i - \hat{\mu}_y)^T$

Clustering w/ GMMs:

$P(x|\theta) = \sum_{j=1}^K w_j\mathcal{N}(x;\mu_j,\Sigma_j)$

$(w_{1:K},\mu_{1:K},\Sigma_{1:K}) = \arg\min -\sum_{i=1}^n \log\sum_j w_j\mathcal{N}(x_i;\mu_j,\Sigma_j)$

↳ Non Convex Probl. → SGD → find $\theta = (w,\mu,\Sigma)$ algorith.

Hard EM: E-Step: $z_i^{(t)} = \arg\max_z P(z|x_i,\theta^{(t-1)}) = \arg\max P(x_i|z,\theta^{(t-1)})\cdot P(z|\theta^{(t-1)})$, $D^{(t)} = \{(x_i,z_i^{(t)})\}_{i=1}^n$

M-Step: $\theta^{(t)} = \arg\max_\theta P(D^{(t)}|\theta)$

↳ $\gamma_j(x) = 1_{\{z(x)=j\}}$, $w_j = \frac{1}{n}\sum\gamma_j(x_i)$, $\mu_j = \frac{\sum\gamma_j(x_i)x_i}{\sum\gamma_j(x_i)}$

$\Sigma_j = \frac{\sum\gamma_j(x_i)(x_i-\mu_j)(x_i-\mu_j)^T}{\sum\gamma_j(x_i)}$

Soft EM: E-Step: $\gamma_j^{(t)}(x_i) = \frac{w_j^{(t-1)}P(x_i|\Sigma_j^{(t-1)},\mu_j^{(t-1)})}{\sum_l w_l^{(t-1)}P(x_i|\Sigma_l^{(t-1)},\mu_l^{(t-1)})}$

M-Step: $w_j^{(t)} = \frac{1}{n}\sum\gamma_j^{(t)}(x_i)$, $\mu_j^{(t)} = \frac{\sum\gamma_j^{(t)}(x_i)x_i}{\sum\gamma_j^{(t)}(x_i)}$

$\Sigma_j^{(t)} = \frac{\sum\gamma_j^{(t)}(x_i)(x_i-\mu_j^{(t)})(x_i-\mu_j^{(t)})^T}{\sum\gamma_j^{(t)}(x_i)}$

EM-Kochrezept: 1) write down $p(z;\theta)$ & $p(x|z;\theta)$

2) $p(z,x;\theta) = p(x|z;\theta)p(z;\theta)$ & $p(x;\theta) = \sum_z p(x|z;\theta)p(z;\theta)$

3) $p(z|x;\theta) = \frac{p(z,x;\theta)}{p(x;\theta)} = \frac{p(z,x;\theta)}{\sum_z p(z,x;\theta)}$

4) E-Step: soft: calc $q(z) = p(z|x,\theta)\ \forall z$; hard: find $z^* = \arg\max q(z)$ (given θ find z_i)

5) M-Step: soft: $\theta = \arg\max\ \mathbb{E}_q[\log(p(z,x;\theta))]$; hard: $\theta = \arg\max \log(p(z^*,x;\theta))$

Types of GMMs: spherical: $\Sigma_k = \sigma_k^2 I \to K$ params

Diag: $\Sigma_k = diag(\sigma_{k,1}^2,\dots,\sigma_{k,d}^2) \to Kd$ params, Tied: $\Sigma_1 = \dots = \Sigma_K \to \frac{d(d+1)}{2}$ params

Full: $w_k, \Sigma_k\ \forall k$ separately: $\frac{d(d+1)}{2}\cdot K$ params for covmat

Remark: $\Sigma_k = \sigma^2 I\ \forall k \to$ k-Means (emp. var. shared)

Init: $w_k^{(0)} = \frac{1}{K}$, $\mu_k^{(0)}$: rand. init / k-means++, Σ_k: spherical

Gaussian-Mixture Bayes Classifier:

$P(y|x) = \frac{1}{z}p(x|y)p(y) = \frac{1}{z}p(y)\sum_j w_j^{(y)}\mathcal{N}(x;\mu_j^{(y)},\Sigma_j^{(y)})$

↳ $\hat{y} = \arg\max_y P(y|x) = \arg\max_y p(x|y)p(y)$

Semi-Supervised Learning (partially missing y_i's)

learn from large amount of unlabeled D & small labeled D

EM for SSL: E-Step: labeled pts: $\gamma_j^{(t)}(x_i) = 1_{\{j=y_i\}}$

unlabeled pts: $\gamma_j^{(t)}(x_i) = P(Z=j|x_i,\mu^{(t-1)},\Sigma^{(t-1)},w^{(t-1)})$

M-Step: $w_j^{(t)} = \frac{1}{n}\sum\gamma_j^{(t)}(x_i)$, $\mu_j^{(t)} = \frac{\sum\gamma_j^{(t)}(x_i)x_i}{\sum\gamma_j^{(t)}(x_i)}$

$\Sigma_j^{(t)} = \frac{\sum\gamma_j^{(t)}(x_i)(x_i-\mu_j^{(t)})(x_i-\mu_j^{(t)})^T}{\sum\gamma_j^{(t)}(x_i)}$

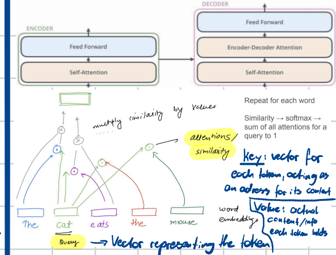
Convergence of EM: $P(x_{1:n}|\theta^{(t)}) \ge P(x_{1:n}|\theta^{(t-1)})$

LLMs:

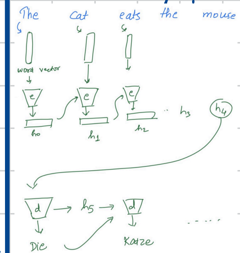
RNNs: output of 1 step is input of next step

Transformer: encoder-decoder architecture. For pred next word look at all input words & translated words all at once → attention!

Structure: Self Attention (look at other words in sentence), Feed Forward (Applied indep to each word), Encoder-Decoder attention layer in Decoder (focus on part of input)



Seq to Seq Model: Go over source sentence 1 by 1 → give current input + last hidden state to encoder → get new hidden state. ⊖ long information flow



Additional Remarks:

- LDA & LogReg share the decision Boundary
- Sample lengths of multivariate Gaussian concentrates around $\sigma\times\sqrt{d}$ (d = dim)
- LLMs: Positional encoding, Self Attention
- Attention: Capturing long-range dependencies between Input Sequences
- Transformer: Multi-head attention, Enc-decoder, Positional encoding, Feed forward NN, residual & skip layer
- Self Att. ops. input to MLP: Values, Keys, Queries
- Query: Vector representing the token
- Self Att.: Focus on specific part of the input seq independently of their position.
- $\forall D$: $|D| < \infty$, $y\in\{\pm1\}$, $x_i \ne x_j\ \forall i\ne j \Rightarrow \exists\theta$: D lin. sep.
- Given kernel $K\in\mathbb{R}^{n\times n}$. It is possible to find $\varphi: \{x_1,\dots,x_n\}\to\mathbb{R}^n$: $K_{ij} = \langle\varphi(x_i),\varphi(x_j)\rangle$
- $A = \begin{pmatrix}a & b\\ c & d\end{pmatrix} \to A^{-1} = \frac{1}{ad-bc}\begin{pmatrix}d & -b\\ -c & a\end{pmatrix}$
- $w\sim\mathcal{N}(0,\frac{2}{p_k}) \to w\sim\mathcal{N}(0,1)\Rightarrow$ [illegible]