Probabilities

Expectation / Var / Covar

$\mathbb{E}[X]=\int_\Omega xf(x)dx=\int_\omega x\mathbb{P}[X=x]dx$

$\mathbb{E}_{Y|X}[Y]=\mathbb{E}_Y[Y|X]$

$\mathbb{E}_{X,Y}[f(X,Y)]=\mathbb{E}_X\mathbb{E}_{Y|X}[f(X,Y)|X]$

$\mathbb{V}(X)=\mathbb{E}[(X-\mathbb{E}[X])^2]=\mathbb{E}[X^2]-\mathbb{E}[X]^2$

$\mathrm{Cov}(X,Y)=\mathbb{E}[(X-\mathbb{E}[X])(Y-\mathbb{E}[Y])]$

Distributions

$\mathcal{N}(x|\mu,\sigma^2)=\frac{e^{-(x-\mu)^2/(2\sigma^2)}}{\sqrt{2\pi\sigma^2}}$

$\mathcal{N}(x|\boldsymbol{\mu},\boldsymbol{\Sigma})=\frac{e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}}$

$\mathrm{Exp}(x|\lambda)=\lambda e^{-\lambda x}$,

$\mathrm{Ber}(x|\theta)=\theta^x(1-\theta)^{(1-x)}$

Sigmoid: $\sigma(x)=1/(1+e^{-x})$

$\mathrm{unif}(a,b): x\in[a,b]?\frac{1}{b-a}:0$

Optimization

Gradient Descent

$\theta^{new}\leftarrow\theta^{old}-\eta\nabla_\theta\mathcal{L}$

Convergence isn't guaranteed.

Less zigzag by adding momentum:

$\theta^{(l+1)}\leftarrow\theta^{(l)}-\eta\nabla_\theta\mathcal{L}+\mu(\theta^{l}-\theta^{(l-1)})$

- Mini-batch: SGD

Newton's Method

Use 2nd order derivation. (Hessian)

$\theta^{new}\leftarrow\theta^{old}-(\nabla_\theta\mathcal{L}/\nabla^2_\theta\mathcal{L})$

$H=\nabla^2_\theta\mathcal{L}$ has to be p.d (convex func).

Bias-Variance tradeoff

$\mathrm{Bias}(\hat f)=\mathbb{E}[\hat f]-f$

$\mathrm{Var}(\hat f)=\mathbb{E}[(\hat f-\mathbb{E}[\hat f])^2]$

$|\mathcal{Z}|\downarrow\uparrow\quad|\mathcal{F}|\uparrow\downarrow\Rightarrow$ Var $\uparrow\downarrow$ Bias $\downarrow\uparrow$

Pred. error = var + b^2 + n

$\mathbb{E}_D\mathbb{E}_{Y|X=x}(\hat f(x)-Y)^2=$

$\mathbb{E}_D(\hat f(x)-\mathbb{E}_D(\hat f(x))^2+(\mathbb{E}_D(\hat f(x))$

$-\mathbb{E}(Y\mid X=x))^2+\mathbb{E}(Y-\mathbb{E}(Y\mid X=x))^2$

0.1 Loss-Functions

0-1 Loss: Piecewise cont, not diff

$\mathcal{L}^{0-1}(y,c(x))=(c(x)=y)\ ?\ 0:1$

Hinge Loss:

$\mathcal{L}^{hinge}(y,c(x))=\max(0,1-w^Txy)$

Perceptron Loss:

$\mathcal{L}^{perc}(y,c(x))=yw^Tx<0\ ?\ -yw^Tx:0$

exponential Loss:

$\mathcal{L}^{exp}(y,c(x))=\exp(-yc(x))$

Logistic Loss:

$\mathcal{L}^{log}(y,c(x))=\log(1+\exp(-yc(x)))$

Risks and Losses

Conditional Expected Risk

$R(f,X)=\int_\mathbb{R}\mathcal{L}(Y,f(X))\mathbb{P}(Y|X)dY$

Total Expected Risk $R(f)=$

$\mathbb{E}_X[R(f,X)]=\int_\mathcal{X}R(f,X)\mathbb{P}[X]dX$

$=\int_\mathcal{X}\int_\mathbb{R}\mathcal{L}(Y,f(X))\mathbb{P}[X,Y]dXdY$.

Empirical Risk Minimizer (ERM) $\hat f$:

$\hat f\in\mathrm{argmin}_{f\in\mathcal{C}}\hat R(\hat f,Z^{train})$

$\hat R(\hat f,Z^{train/test})=\frac{1}{n}\sum_{i=1}^n Q(Y_i,\hat f(X_i))$

$Z^{train}=(X_1,Y_1),\dots,(X_n,Y_n)$

$\mathbb{P}[X|Y]=\frac{\mathbb{P}[X,Y]}{\mathbb{P}[Y]}=\frac{\mathbb{P}[Y|X]\mathbb{P}[X]}{\mathbb{P}[Y]}$

Math and Basics

Some gradients

f	∇_xf	f	df/dx
$\|x\|_2^2$	$2x$	a^Tx	a
$\|x\|_1$	$sng(x)$	x^Ta	a
x^TAx	$(A+A^T)x$	σ	$\sigma(1-\sigma)$
x^Tx	$2x$		

$\nabla_\beta(y-X\beta)^T(y-X\beta)=2(X^TX\beta-X^Ty)$

Positive semi-definite matrices M

$\forall x\in\mathbb{R}^n: x^TMx\ge0\Leftrightarrow$

all eigenvalues of M are pos: $\lambda_i\ge0$

Kernels

Similarity based reasoning

$K(\mathbf{x},\mathbf{x}')$ pos.semi-def. (all EV ≥0)

Gram Matrix $K=K(\mathbf{x}_i,\mathbf{x}_j)$, $1\le i,j\le n$

$K(\mathbf{x},\mathbf{x}')=\phi(\mathbf{x})^T\phi(\mathbf{x}')$, $K(\mathbf{x},\mathbf{x}')=K(\mathbf{x}',\mathbf{x})$

$K(\mathbf{x},\mathbf{x}')=K_1(\mathbf{x},\mathbf{x}')K_2(\mathbf{x},\mathbf{x}')$

$K(\mathbf{x},\mathbf{x}')=\alpha K_1(\mathbf{x},\mathbf{x}')+\beta K_2(\mathbf{x},\mathbf{x}')$

$K(\mathbf{x},\mathbf{x}')=K_1(h(\mathbf{x}),h(\mathbf{x}'))\quad h:\mathcal{X}\to\mathcal{X}$

$K(\mathbf{x},\mathbf{x}')=h(K_1(\mathbf{x},\mathbf{x}'))\quad h$: poly/exp

Kernel Function Examples:

$K(\mathbf{x},\mathbf{x}')=\mathbf{x}^T\mathbf{x}'\quad K(\mathbf{x},\mathbf{x}')=(\mathbf{x}^T\mathbf{x}'+1)^p$

RBF(Gauss): $K(\mathbf{x},\mathbf{x}')=e^{-\|\mathbf{x}-\mathbf{x}'\|_2^2/h^2}$

Sigmoid: $K(\mathbf{x},\mathbf{x}')=\tanh(\alpha\mathbf{x}^T\mathbf{x}'+c)$

1 Density Estimation with Parametric Models

1.1 Maximum Likelihood (MLE)

Likelihood: $\mathbb{P}[\mathcal{X}|\theta]=\prod_{i\le n}p(x_i|\theta)$

Find: $\hat\theta\in\mathrm{argmax}_\theta\,\mathbb{P}[\mathcal{X}|\theta]$

Procedure: solve $\nabla_\theta\log\mathbb{P}[\mathcal{X}|\theta]\equiv0$

Consistent: converges to best θ_0.

1.2 Maximum A Posteriori (MAP)

Assume prior $\mathbb{P}(\theta)$

Find: $\hat\theta\in\mathrm{argmax}_\theta P(\theta|\mathcal{X})=$

$=\mathrm{argmax}_\theta P(\mathcal{X}|\theta)P(\theta)$

Solve $\nabla_\theta logP(\mathcal{X}|\theta)P(\theta)=0$

1.3 Bayesian density learning

Prior Knowledge of $p(\theta)$,

Find Posterior Density: $p(\theta|\mathcal{X})$.

$\mathcal{X}^n=\{x_1,\cdots,x_n\}$

$p(\theta|\mathcal{X}^n)=\frac{p(x_n|\theta)p(\theta|\mathcal{X}^{n-1})}{\int p(x_n|\theta)p(\theta|\mathcal{X}^{n-1})d\theta}$

1.4 Frequentist (Fisher): ML estimation

1. Define parametric model (e.g. $\mathcal{N}(\theta,1)$)
2. Define the likelihood as function of parametric model (prob of the observations given the parameter θ), e.g. $\mathbf{P}(y_1,\dots,y_n\mid\theta)=\prod_{i\le n}\mathbf{P}(y_i\mid\theta)=\prod_{i\le n}\mathcal{N}(y_i,\theta,1)$
3. estimator maximizes $\hat\theta_{ML}=\arg\max_\theta\mathbf{P}(y_1,\dots,y_n\mid\theta)$ (log-likelihood)

1.5 Properties of ML Estimators:

- Consistent ($\theta_{ML}\to\theta_0$) as $n\to\infty$
- Equivariant: $\hat\theta_{ML}$: θ, $g(\hat\theta_{ML})$: $g(\theta)$, g invertible
- Asymptotically normal: $1/\sqrt{n}(\theta_{ML}-\theta_0)$ converges to rv with distribution $\mathcal{N}(0,J^{-1}(\theta)I(\theta)J^{-1}(\theta))$
- Asymptotically efficient: θ_{ML} minimizes $\mathbb{E}[(\theta_{ML}-\theta_0)^2]$. I.e. $\mathbb{E}[(\theta_{ML}-\theta_0)^2]=\frac{1}{I_n(\theta_0)}$

Rao Cramer Bound

There exists no estimator such that $\mathbb{E}[(\hat\theta^*-\theta_0)^2]=0$, $\mathbb{E}[(\hat\theta-\theta_0)^2]\ge\frac{1}{I_n(\theta_0)}$, $\hat\theta$ unbiased $I_n(\theta_0)=-\mathbb{E}[\frac{\partial^2\log[\mathcal{X}_n|\theta]}{\partial\theta^2}]$

Efficiency $e(\theta_n)=\frac{1}{\mathrm{Var}[\hat\theta_n]I_n(\theta)}$

$e(\theta_n)=1$ (efficient) $\lim_{n\to\infty}e(\theta_n)=1$ (asym. efficient)

Stein estimator For finite samples might be better sol (ML estimators not nec. efficient).

2 Linear Regression

- Optimal solution for regression $\arg\min_f\mathbb{E}(Y-f(X))^2$ given by $f^*(x)=\mathbb{E}(Y|X=x)$
- **Statistical learning theory:** Directly minimize empirical risk $\arg\min_f\sum_{i=1}^n(y_i-f(x_i))^2$

(1) Ordinary least squares

$Y=\beta_0+\sum_{j=1}^dX_j\beta_j=X^T\beta$,

β_0=bias, $X,\beta\in\mathbb{R}^{d+1}$.

- Minimization through gradient descent or closed form

Closed Form

$RSS(\beta)=\sum_{i=1}^n(y_i-x_i^T\beta)=$

$(\mathbf{y}-\mathbf{X}\beta)^T(\mathbf{y}-\mathbf{X}\beta)$, $\hat\beta=(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$

Prediction

$\hat{\mathbf{y}}=\mathbf{X}\hat\beta=\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$,

(2) Ridge regression:

$\min(\mathbf{y}-\mathbf{X}\beta)^\top(\mathbf{y}-\mathbf{X}\beta)$ s.t. $\sum_{j=1}^d\beta_j^2\le t$

$\Rightarrow(\mathbf{y}-\mathbf{X}\beta)^\top(\mathbf{y}-\mathbf{X}\beta)+\lambda\|\beta\|^2$

$\mathbf{X}\hat\beta^{ridge}=\sum_{i=1}^d\mathbf{u}_j\frac{d_j^2}{d_j^2+\lambda}\mathbf{u}_j^T\mathbf{y}$

- $\frac{d_j^2}{d_j^2+\lambda}$ small for small SV.
- $d_j\to1$ for large SV.
- Suppresses contributions of small Evals (remove multicoli. blowing up variance).

(3) LASSO

$\hat\beta^{LASSO}=\mathrm{argmin}_\beta(\mathbf{y}-\mathbf{X}\beta)^\top(\mathbf{y}-\mathbf{X}\beta)$

subject to $\sum_{j=1}^d|\beta_j|\le s$.

Rewrite as $(\mathbf{y}-\mathbf{X}\beta)^\top(\mathbf{y}-\mathbf{X}\beta)+\lambda|\beta_j|$

- Large λ will set some coefficients equal to 0 $\to$ sparse solution (model selection)

(4) Bayesian Linear Regression

Define prior distribution over β

$p(\beta|\Lambda)=\mathcal{N}(\beta|\mathbf{0},\Lambda^{-1})\propto e^{-\frac{1}{2}\beta^T\Lambda\beta}$,

$\Lambda=\Sigma^{-1}$ precision mat. Favors β=0.

Posterior: Given observed $\mathbf{X},\mathbf{y}$

$p(\beta|\mathbf{X},\mathbf{y},\Lambda)=\mathcal{N}(\beta|\mu_\beta,\Sigma_\beta)$

$\mu_\beta=(\mathbf{X}^T\mathbf{X}+\sigma^2\Lambda)^{-1}\mathbf{X}^T\mathbf{y}$

$\Sigma_\beta=\sigma^2(\mathbf{X}^T\mathbf{X}+\sigma^2\Lambda)^{-1}$

Bayesian lr with gaussian prior = ridge for $\Lambda=\lambda\mathbb{I}_d,\sigma=1$

3 Non-Linear Regression

3.1 Feature Transformations

$f(X)=\sum_{m=1}^M\beta_mh_m(X)$, $h_m(X):\mathbb{R}^d\mapsto\mathbb{R}$, $1\le m\le M$

- Determine Boundaries e.g. $|x_1|+|x_2|<1\Rightarrow|x_1|+|x_2|-1<0$. Use $\phi(X)=|x_1|+|x_2|-1$, $w=1$
- 2 boundaries: multiply 2 equations

3.2 Gaussian Process Regression

joint Gaussian over all outputs

$\mathbf{y}=f(X)+\varepsilon\quad\varepsilon\sim\mathcal{N}(\varepsilon|0,\sigma\mathbb{I}_n)$,

$f(X)\sim GP(m(X),k(X,X'))$

$m(X)=\mathbf{0}$ if $f(X)=X\beta$

Prediction

$P(\begin{bmatrix}\mathbf{y}\\ y_*\end{bmatrix})=\mathcal{N}(\mathbf{y}|m(X),\begin{bmatrix}\mathbf{C_n}&\mathbf{k}\\ \mathbf{k^T}&c\end{bmatrix})$

$p(y_*|\mathbf{x}_*,\mathbf{X},\mathbf{y})=\mathcal{N}(y_*|\mu_*,\sigma_*^2)$

$\mu_{y_*}=\mathbf{k}^T\mathbf{C}_n^{-1}\mathbf{y}\quad\mathbf{C}_n=\mathbf{K}+\sigma^2\mathbb{I}$

$\sigma_*^2=c-\mathbf{k}^T\mathbf{C}_n^{-1}\mathbf{k}\quad c=k(x_*,x_*)+\sigma^2$

$\mathbf{k}=k(x_*,\mathbf{X})\quad\mathbf{K}_{ij}=k(x_i,x_j)$

4 Classification

$A=\frac{\#correct}{all}$, $R=\frac{TP}{TP+FN}$, $P=\frac{TP}{TP+FP}$

4.1 Discriminative / Generative Models

Discriminative models: model decision boundary between classes $p(y|x)$. E.g. HMM, Naive Bayes

Generative model: explicitly model the distribution of each class. $p(x,y)$. E.g. Perc., SVM, trad. NNs.

4.2 Classifiers

Probabilistic Generative Classifier

(1) Assume distribution of labels $p(Y|\theta)$ and $p(X|Y=y)$,

(2) MLE over joint likelihood $P(\mathbf{X},\mathbf{y}|\theta)$,

(3) Bayes $y=\mathrm{argmax}_y\,p(y|X)\propto p(y)\prod_{i=1}^np(x_i|y)$

Prob. Discr. Classifier (2D: log. regr)

(1) Assume the posterior $P(y=1|X)=\sigma(w^\top x+w_0)=\sigma(\tilde w^\top x)$.

(2) MLE over likelihood $p(\mathbf{y}|\mathbf{X},w)$

$=p(\mathbf{y}=1|\mathbf{X},w)^y\cdot(1-p(\mathbf{y}=1|\mathbf{X},w))^{1-y}$

$\Rightarrow L(w)=\log p(\mathbf{y}|\mathbf{X},\mathbf{w})$

$=c+\sum_i[y_i\log\sigma(\mathbf{w}^\top x_i)$

$+(1-y_i)\log(1-\sigma(\mathbf{w}^\top x_i)]$

(3) GD/ Newton's over $-L(w)$.

(4) w^* to predict.

Discriminative Classifier

Choose loss func $\mathcal{L}:\mathcal{Y}\times\mathcal{Y}\to\mathbb{R}^+$,

Approximate exp. risk with the emp. loss $\hat R$. Optimal classif.

$c^*=\mathrm{argmin}_c\hat R$

4.3 Least Squares (LDA, QDA)
Make the model predictions as close as possible to a set of target values.
LDA: Assume $\Sigma_0 = \Sigma_1$
$p(y \mid x) = \sigma(\mathbf{w}^T x + w_0)$
QDA: General
$p(y \mid x) = \sigma(x^T \mathbf{W} x + x^T \mathbf{w} + w_0)$

4.4 Fisher's Linear Discriminant
Max. distance of means of projected classes to find projective sep. plane.
proj mean: $\mathbf{m}_k = \frac{1}{n_k}\sum_{n\in\mathcal{C}_k} w^T x_n = w^T m_k$
Within-class var ($y_k = w^T x_k$):
$s_k^2 = \sum_{n\in\mathcal{C}_k}(y_k - \mathbf{m}_k)^2$
Dist of proj means: $|w^T(m_1 - m_2)|$
Class proj. cov: $\mathbf{s}_1^2+\mathbf{s}_2^2 = w^T(s_1^2+s_2^2)w$

Fishers Criterion:
$J(w) = \frac{(\mathbf{m}_1+\mathbf{m}_2)^2}{\mathbf{s}_1^2+\mathbf{s}_2^2} = \frac{\text{between class var}}{\text{within class var}}$
$= \frac{w^T(m_1-m_2)(m_1-m_2)^T w}{w^T(s_1^2+s_2^2)w} = \frac{w^T S_B w}{w^T S_w w}$

Classification with fisher:
$\mathbf{w}^T x = \sum_i w[i]x[i]$
1. Fisher's projection $w^* \propto S_w^{-1}(\overline{x}_0 - \overline{x}_1)$
2. Fit mix of gaussians
3. Bayes decision theory

4.5 Perceptron Algorithm
Goal: Compute $w \in \mathbb{R}^d = sgn(w^T x_i)$
Cost Function: $L(\mathbf{w}) = \sum_{i\le n}\mathcal{L}(y_i, c(x_i)) = \sum_{i\in\mathcal{M}} -y_i\mathbf{w}^\top x_i$
$\nabla L(\mathbf{w}) = \sum_{i: y_i\mathbf{w}^\top x_i<0} -y_i x_i$
GD with update: $\eta(k)(-y_i x_i)$

Variable increment perceptron **converges** if Train set is lin.sep., $\eta(k)\ge 0$, $\sum_{k=0}^t \eta(k)\to\infty$ for $t\to\infty$, $\frac{\sum_{k\le t}\eta^2(k)}{(\sum_{k\le t}\eta(k))^2}\to 0$ for $t\to\infty$

4.6 Lagrange Dual Formulation
$\min_w f(w)$ *s.t.* $g_i(w)=0, h_j(w)\le 0$
1. Generalized Lagrangian: $\mathcal{L}(\mathbf{w},\lambda,\alpha) = f(\mathbf{w}) + \sum_i \lambda_i g_i(\mathbf{w}) + \sum_j \alpha_j h_j(\mathbf{w}), \alpha_j\ge 0$
2. $\max_{\alpha,\lambda}\min_w \mathcal{L} \le \min_w \max_{\alpha,\lambda}\mathcal{L}$
3. constraints $\nabla_\mathbf{w}\mathcal{L} = 0, \nabla_{w_0}\mathcal{L}=0$
4. $\max_{\alpha,\lambda}\mathcal{L}$ with plugged in $\Rightarrow \alpha_i$

4.7 Slaters ⇒ Str. dual. ⇒ compl. Sl.
- weak d.: $d^*\le p^*$, strong d: $d^*=p^*$
- slaters: $\exists x: h_j(x)<0$ (strict)

Complementary Slackness:
$\lambda_i f_i(x^*) = 0 \quad \forall i:$
$\lambda_i > 0 \Rightarrow f_i(x^*)=0, f_i(x^*)<0 \Rightarrow \lambda_i = 0$

4.8 Support Vector Machine (SVM)
$\min_{w,w_0}\frac12\|w\|$ s.t. $1-y_i(\mathbf{w}^T x_i + w_0)\le 0$
$\mathbf{y}_i$ are support vectors

Functional Margin Problem:
minimizes $||\mathbf{w}||$ for m=1:
$L(\mathbf{w}, w_0, \alpha) = \frac12\mathbf{w}^T\mathbf{w} - \sum_{i=1}^n \alpha_i[z_i(\mathbf{w}^T\mathbf{y}_i + w_0)-1]$
where αs are Lagrange multipliers.

Dual Representation:
Conditions: $\frac{\partial\mathcal{L}}{\partial\mathbf{w}}=0, \frac{\partial\mathcal{L}}{\partial w_0}=0$
$\Rightarrow w^* = \sum_i \alpha_i y_i x_i,\quad \sum_i\alpha_i y_i = 0$
$\max_\alpha \frac12\sum_{i,j}\alpha_i\alpha_j y_i y_j x_i^T x_j + \sum_i \alpha_i - \sum_{i,j}\alpha_i\alpha_j y_i y_j x_i^T x_j, \alpha_i\ge 0, \sum_i\alpha_i y_i = 0$
Simplifies to:
$\max_\alpha \sum_i \alpha_i - \frac12\sum_{i,j}\alpha_i\alpha_j y_i y_j x_i^T x_j$, s.t. $\alpha_i\ge 0, \sum_i\alpha_i y_i = 0$
Optimal Margin: $\mathbf{w}^T\mathbf{w} = \sum_{i\in SV}\alpha_i^*$

$w^T x + w_0 = \sum_{i=1}^n(\alpha_i y_i x_i)^T x + w_0 = \sum_{i=1}^n \alpha_i y_i\langle x_i, x\rangle + w_0$, efficient.

NonLinear SVM
Use kernel in discriminant funct:
$g(\mathbf{x}) = \sum_{i,j=1}^n \alpha_i\alpha_j z_i z_j K(\mathbf{x_i},\mathbf{x})$

Soft Margin SVM (relax constraints)
$\min_{\mathbf{w},w_0,\xi}\frac12\|\mathbf{w}\|^2 + C\sum_{i\le n}\xi_i$
s.t. $y_i(\mathbf{w}^T x_i + w_0)\ge 1-\xi_i, \quad \xi_i\ge 0$

Multiclass SVM
score per class, set margin as min diff of largest + second largest score.
$\min_w \frac12\|w\| = \min_{\{w_z\}_{n=1}^M}\frac12\sum_z^M w_z^T w_z$
$w^T = w_1^T,\dots,w_M^T$ s.t. $\forall y_i\in Y$
$(\mathbf{w}_{z_i}^T\mathbf{y}_i + w_{z_i,0}) - \max_{z\ne z_i}(\mathbf{w}_z^T\mathbf{y}_i + w_{z,0})\ge 1$
$\hat{z} = \text{argmax}_z(w_z^T y + w_{z,0})$

Structured SVM
$\min_w \frac12\|w\|^2$ s.t.
$w^T\Psi(x_i,y_i)\ge \Delta(y_i,y') + w^T\Psi(x_i,y')$
$\forall y'\ne y_i, i\le n$

Output Space Representation as joint feature map: $\psi(z,y)$
Scoring function: $f_\mathbf{w}(z,\mathbf{y}) = \mathbf{w}^T\psi(\mathbf{z},\mathbf{y})$
Classify: $\hat{z} = h(\mathbf{y})\,\text{argmax}_{z\in\mathcal{K}} f_{\mathbf{w}(z,\mathbf{y})}$

5 Ensemble Methods
5.1 Bagging
Bootstrap sets: Draw M bootstrap sets, Train M base models $b^{(1)},\dots,b^{(M)}$, aggregate

Random forests
Bagging with trees. Each tree considers **subset of variables**.
Reduce corr. between base trees.

5.2 Boosting
Fit models iteratively (model depends on prev. fitted). Each m. gives higher weight to the observations that were wrong in prev. step).

Ada Boost (Adaptive Boosting):
Loss function: 0-1 Loss, place high weights on samples that are very hard to classify. Detect Outliers by high w.

Gradient Boosting
Learn dir from the residual error instd of updating the weights.
$f_M(x) = \sum_{i=1}^M\beta_i h_i(x)$

Forward Stagewise Additive Modeling
Method to approximately compute a classifier of the form
$c(x) = sgn(\sum_t \alpha_t b^{(t)}$ that approximately minimizes the empirical loss
$\sum_{i\le n} L(y_i, c(x_i))$. ⇒ **AdaBoost equ.**

6 Deep Learning
6.1 Activation Functions
Make NN function non-linear.
ReLu: $f(x) = \begin{cases}0 & for\ x<0\\ x & for\ x\ge 0\end{cases}$
Sigmoid: $\sigma(x) = \frac{1}{1+\exp(-x)}$
Tanh: $\tanh(x) = \frac{2}{1+e^{-2x}} - 1$

6.2 Training Neural Networks
$\min_\theta\sum_{i\le n}\mathcal{L}(y_i, NN_\theta(x_i))$

6.3 Regularization
Early stop., Dropout, bay. priors, L2

6.4 Variational Autoencoders
Learn meaningful representations without supervision.

Objective
enc_θ mapping measurements in $\mathcal{X}$ to prob. dists. over space $\mathcal{Z}$
$enc_\theta: x\in\mathcal{X}\mapsto p_\theta(\cdot|x)$ over $\mathcal{Z}$
Variational Inference: find posterior

(1) Define prior and calculate likelihood (decoder), (2) approximate posterior (encoder)
Informative, disentangled and robust by the choice of $p_\theta(\cdot|Z)$ and $q_\phi(\cdot|x)$.

Denoising Autoencoder
Blank out parts of the input image during training; more robust.

7 Clustering
k-means or EM. Neither can detect outliers! EM more sensitive to outl. (no constraints on the covariance matrix).

7.1 k-means
Assign each x to closest center. Compute new centers. Repeat.

7.2 Gaussian Mixtures
Direct optimization of log-likelihood is (sum within the log) → no closed form solution.
EM Mixture models solve this: introduce latent indicator vars for mode assignments, max. joint likelihood of observable and latent vars.

7.3 EM algorithm
$M_{\mathbf{x}c} = \begin{cases}1 & c\ generated\ \mathbf{x}\\ 0 & otw\end{cases}$
This gives
$P(\mathcal{X}, M|\theta) = \prod_{x\in\mathcal{X}}\prod_{c=1}^k(\pi_c P(\mathbf{x}|\theta_c))^{M_{\mathbf{x}c}}$

E-Step
$\gamma_{\mathbf{x}c} = \mathbb{E}_M[M_{\mathbf{x}c}|\mathcal{X},\theta^{(j)}] = \frac{P(\mathbf{x}|c,\theta^{(j)})P(c|\theta^{(j)})}{P(\mathbf{x}|\theta^{(j)})}$

M-Step
$\mu_c^{(j+1)} = \frac{\sum_{\mathbf{x}\in\mathcal{X}}\gamma_{\mathbf{x}c}\mathbf{x}}{\sum_{\mathbf{x}\in\mathcal{X}}\gamma_{\mathbf{x}c}}$
$(\sigma_c^2)^{(j+1)} = \frac{\sum_{\mathbf{x}\in\mathcal{X}}\gamma_{\mathbf{x}c}(\mathbf{x}-\mu_c)^2}{\sum_{\mathbf{x}\in\mathcal{X}}\gamma_{\mathbf{x}c}}$
$\pi_c^{(j+1)} = \frac{1}{|\mathcal{X}|}\sum_{\mathbf{x}\in\mathcal{X}}\gamma_{\mathbf{x}c}$

8 Non-Param Bayesian Methods
8.1 Dirichlet (Multivariate Beta)
$Dir(\mathbf{x}|\alpha) = \frac{1}{B(\alpha)}\cdot\prod_{k=1}^n x_k^{\alpha_k-1}$
$B(\alpha) = \frac{\prod_{k=1}^n\Gamma(\alpha_k)}{\Gamma(\sum_{k=1}^n\alpha_k)}$

Dirichlet Proces $DP(\alpha, H)$:
$(G(T_1)\dots G(T_K))\sim Dir(\alpha H(T_1)..\alpha H(T_k))$

Stick-Breaking Process
$\beta_k\sim Beta(1,\alpha), \rho_k = \beta_k(1-\sum_{i=1}^{k-1}\rho_i)$

Prior:
$p(z_i=k|\mathbf{z}_{-i},\alpha) = \begin{cases}\frac{N_{k,-i}}{\alpha+N-1} & existing\ k\\ \frac{\alpha}{\alpha+N-1} & otherwise\end{cases}$

Chinese Restaurant Problem
Clustering property to draw samples. Sit at table ∝ # people on it.

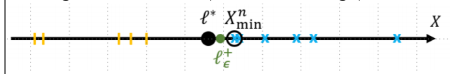
8.2 Gibbs Sampling
Init: assign data to cluster with prior $\pi_i, \sum\pi_i<1$ (using e.g. stick-br.)
Remove x from k, compute θ_k,
Compute Gibbs sampler prob. (CRP) and sample new cluster assignment $z_i\sim p(z_i|x_{-i},\theta_k)$

Final Gibbs sampler (Stick-Breaking):
$p(z_i=k|\mathbf{z}_{-i},\mathbf{x},\alpha,\mu) = Prior\times likelihood$
$= \begin{cases}\frac{N_{k,-i}}{\alpha+N-1}p(x_i|\mathbf{x}_{-i,k},\mu) & existing\ k\\ \frac{\alpha}{\alpha+N-1}p(x_i|\mu) & otherwise\end{cases}$

9 PAC Learning
9.1 The PAC Learning Model
ε error parameter, δ confidence val
- PAC learn.: $\mathbf{P}(\mathcal{R}(\hat{c}_n^*)\le\varepsilon)\ge 1-\delta$
- General Setting: $\mathbf{P}(\mathcal{R}(\hat{c}_n^*) - \inf_{c\in\mathcal{C}}\mathcal{R}(c)\le\varepsilon)\ge 1-\delta$
- Efficiently PAC learnable: Algorithm runs in poly time in $1/\varepsilon$ and $1/\delta$ (computing X_{min}^n and compl. of n)

9.2 Rectangle Learning
Pick tight rectangle. Diff between picked rectangle $\hat{R}$ and true rectangle R with few examples. Rectangles are efficiently PAC learnable.

9.3 Example: Half-line learning
$\mathcal{C} = \mathcal{H} = \{\mathbb{I}_{[l,\infty)}: l\in\mathbb{R}\}$, where
$\mathbb{I}_{[l,\infty)} = \begin{cases}0 & x<l\\ 1 & x\ge l\end{cases}$ *fix* $f^*\in\mathbb{R}$, *consider* $c^* = \mathbb{I}_{[l^*,\infty)}$
let
$X_{min}^n := \min_{i\le n, Y_i=1} X_i, \hat{c}_n := \mathbb{I}_{X_{min}^n,\infty)}$
Let $l_\varepsilon^+\in\mathbb{R}$ s.t. $\mathbf{P}(l^*\le X_i\le l_\varepsilon^+) = \varepsilon$.



If $l_\varepsilon^+\le X_{min}^n$: $l^*\le X_i\le X_{min}^n$.
Then $\mathcal{R}(\hat{c}_n) = \mathbf{P}(l^*\le X_i\le X_{min}^n)$
$\ge \mathbf{P}(l^*\le X_i\le l_\varepsilon^+) = \varepsilon$
$\mathbf{P}(l_\varepsilon^+\le X_{min}^n) = \prod_i^n \mathbf{P}[X_i\notin[l^*,l_\varepsilon^+)] = \prod(1-\mathbf{P}(l^*\le X_i\le l_\varepsilon^+)) = (1-\varepsilon)^n$

$\mathbf{P}(\mathcal{R}(\hat{c}_n) \geq \varepsilon) = \mathbf{P}(l_\varepsilon^+ \leq X_{min}^n)$
$= (1-\varepsilon)^n \leq \delta. \rightarrow n \geq \frac{1}{\varepsilon} \log \frac{1}{\delta}.$