Regression and Optimization

Square Error (featurized):
$L(w) = \frac{1}{n}\sum(y_i - w^\top \phi_j(x_i)^2) = \frac{1}{n}||y-\Phi w||_2^2$
$\nabla_w L(w) = \frac{2}{n}\Phi^\top(\Phi\hat{w} - y)$ $(\Phi^\top\Phi$ psd)
Gradient Descent: $w^{t+1} = w^t - \eta\nabla_w L(w^t)$
Convergence: $||w^{t+1}-\hat w||_2^2 \le \rho^{t+1}||w^0-\hat w||_2^2$ with speed $\rho = ||I-\eta X^\top X||_{op}$ for $\eta \le \frac{2}{\lambda_{\max}}$.
$\eta^*: \frac{2}{\lambda_{\min}+\lambda_{\max}}, \rho^*: 1-\eta^*\lambda_{\min} = \frac{\kappa-1}{\kappa+1}, \kappa: \frac{\lambda_{\max}}{\lambda_{\min}}$
Momentum: $w^{t+1} = w^t + \Delta w^{t-1} - \eta\nabla L(w^t)$
SGD: $w^{t+1} = w^t - \eta\nabla L_{\mathcal S}(w^t), \mathcal S \subset [n]$

Model Selection

Empirical Risk: $L(\hat f; D) = \frac1n\sum l(\hat f(x_i), y_i)$
Exp. Estimation Err.: $\mathbb E_X\, l(\hat f_D(x), f^*(x))$
Gen. Err.: $L(\hat f_D; \mathbb P_{X,Y}) = \mathbb E_{X,Y}\, l(\hat f_D(X), Y)$
Test Err.: $\frac{1}{|D_{\text{test}}|}\sum l(\hat f(x), y) \overset{\text{LLN}}{\to} L(\hat f; \mathbb P_{X,Y})$
For $\mathbb E$[Gen. Err.]: $D = D_{\text{train}} \uplus D_{\text{val}} \uplus D_{\text{test}}$
D_{val} is used for independent model selection.
K-Fold CV: $D_{\text{train}}, D_{\text{val}} \overset{\text{find}}{\to} \lambda, \ldots \overset{\text{use}}{\to} \hat f_{D_{\text{train}}\uplus D_{\text{val}}}$

Bias-Variance Tradeoff

$\mathbb E$[Gen. Err.] = Bias^2 + Variance + Noise
$\mathbb E[L(\hat f_D; \mathbb P_{X,Y})] = \mathbb E_X[(\mathbb E_D[\hat f_D(X)] - f^*(X))^2]$
$+\mathbb E_X[\mathbb E_D[(\hat f_D(x) - \mathbb E_D[\hat f_D])^2]] + \sigma^2$.
Bias: Diff. of average model $\mathbb E_D[\hat f_D]$ to f^*.
Variance: Diff. of some model $\hat f$ to $\mathbb E_D[\hat f_D]$.

Regularization

Lasso: argmin$(||y-Xw||_2^2 + \lambda||w||_1)$ $\lambda\in\mathbb R$
Ridge: argmin$(||y-Xw||_2^2+\lambda||w||_2^2)$ $\lambda\in\mathbb R$
With closed form: $\hat w = (X^\top X + \lambda I^d)^{-1}X^\top y$.
Thus $\lambda \nearrow \implies$ bias $\nearrow$ and variance $\searrow$.

Classification

Zero-One Loss: $l_{0-1}(\hat f(x), y) = \mathbb I_{\{y \ne \text{sign} \hat f(x)\}}$.
a$_{0-1}$: $l(\hat y, y): c_{FP}\mathbb I_{\hat y=1,y=-1} + c_{FN}\mathbb I_{\hat y=-1,y=1}$
Prop. $l(\hat f(x), y) = g(y\hat f(x))$: • $\searrow$ • conv. • diff.
• 0 if $y=\hat y$ • robust to noise • ¬∗-Grad for $y\ne\hat y$
Exponential loss: $g_{\exp}(y\hat f(x)) = e^{-y\hat f(x)}$ (∗)
Logistic Loss: $g_{\log}(y\hat f(x)) = \log(1+e^{-y\hat f(x)})$
Linear loss: $g_{\text{lin}}(y\hat f(x)) = -y\hat f(x)$
Cross Entropy: $-\log(e^{f_y(x)}/\sum_{k\hat=\text{class}} e^{f_k(x)})$
Softmax: $[\text{softmax}(f(x))]_i = e^{f_i(x)}/\sum_k e^{f_k(x)}$
Logistic/Sigmoid: $\sigma(z) = 1/(1+e^{-z})$

Linear Classifiers $w^\top x$ (with log. loss)

GD $\to w||w_{MM} = \text{argmax}_{||w||=1}$ margin(w) w/ margin$(w) = \min_i y_i\langle w, x_i\rangle$ (min distance to x_i)
Hard SVM: $\min_w ||w||_2$ s.t. $\forall i.\ y_i w^\top x_i \ge 1$

Other Methods

kNN: Classify by k nearest neighbors classes.
Decision Trees: Tree w/ rules $r_v(x) = \mathbb I_{\{x_i > t_i\}}$.

Hypothesis Testing

	y_{+1}	y_{-1}
$\hat y_{+1}$	TP	FP/T_I
$\hat y_{-1}$	FN/T_{II}	TN

FNR $= \frac{\#FN}{\#y=1}$ FDR $= \frac{\#FP}{\#y=1}$
Precision $= \frac{\#TP}{\#\hat y=+1}$
FPR $= \frac{\#FP}{\#y=-1}$ Recall/TPR $= \frac{\#TP}{\#y=+1}$
τ decision instead of 0: τ small: TPR/FPR ↑; τ medium: FNR/FPR↓; τ big: FPR/TPR ↓
AUROC: Plot TPR(1-FNR)/FPR, with diff. τ
F1-Score: $\frac{2}{\frac{1}{\text{recall}}+\frac{1}{\text{precision}}}$, want both large.

Generalizations (Gen. Err. = GE)

Worst-group GE: $\sup_{g\in G}\mathbb E^g_{(x,y)}\mathbb I_{\{y\ne\hat y\}}$
Domain-shift GE: Accurate on data $\sim D_{\text{test}}$.
Adversarially robust: $\mathbb E_{(x,y)}\sup_{x'\in T(x)}\mathbb I_{\{y\ne\hat y\}}$

Kernel Trick

As $w \in \text{Im}(\Phi^\top) \Rightarrow w = \Phi^\top\alpha$; $K_{i,j} = k(x_i,x_j)$.
Conditions for a valid kernel function k:
• $k(x,z) = k(z,x)$ • K psd s.t. $\forall x.\ x^\top Kx \ge 0$
Want to find map ϕ s.t. $k(x,y) = \langle\phi(x),\phi(y)\rangle$.
Inner Product kernel: $k(x,z) = h(\langle x,z\rangle)$
Poly ker.: $k(x,z) = (c_{\ge0}+\langle x,z\rangle)^m, d_\phi = \binom{d+m}{d}$
RFB kernel: $k(x,z) = \exp\left(\frac{||x-z||_2^\alpha}{\tau}\right)$ which is
Gaussian: $\alpha=2$, **Laplacian**: $\alpha=1$. $d_\phi=\infty$
Kernel Composition: • $k_1 \dotplus k_2$ • $c\cdot k$ $(c>0)$
• $k((x\ y),(x'\ y')) = k_1(x\ x') \dotplus k_2(y\ y')$
Kernelized Ridge: $\frac1n||y-K\alpha||^2 + \lambda\alpha^\top K\alpha$
The final model is $\hat f(x) = \hat\alpha^\top[k(x_i,x)]_i$.

Neural Networks

Activation Function: $\phi(x;w) = \varphi(w^\top x)$
• **tanh**: $\frac{\exp(z)-\exp(-z)}{\exp(z)+\exp(-z)}$ • **relu**: $\max\{0,z\}$ • $\sigma(z)$
Universal Approx. Thm.: $\forall\varepsilon_{>0}, \exists$neural network that approximates any function within ε.

Forward Propagation $W \in \mathbb R^{out\times in}$

Input l.: $v^{(0)} = [x;1]$ *Output l.*: $f = W^{(L)}v^{(L-1)}$
Hidden l.: $z^{(l)} = W^{(l)}v^{(l-1)}$ & $v^{(l)} = [\varphi(z^{(l)}); 1]$

Backward Propagation

Given from L+1, to compute, given from FP.
$(\nabla_{W^{(L)}} l)^\top = \frac{\partial l}{\partial f}\frac{\partial f}{\partial W^{(L)}} = \frac{\partial l}{\partial f}v^{(L-1)}$
$(\nabla_{W^{(L-1)}} l)^\top = \frac{\partial l}{\partial f}\frac{\partial f}{\partial z^{(L-1)}}\frac{\partial z^{(L-1)}}{\partial W^{(L-1)}} = \ldots\ldots v^{(L-2)}$
$(\nabla_{W^{(L-2)}} l)^\top = \frac{\partial l}{\partial f}\frac{\partial f}{\partial z^{(L-1)}}\frac{\partial z^{(L-1)}}{\partial z^{(L-2)}}\frac{\partial z^{(L-2)}}{\partial W^{(L-2)}}$
Where error $\delta^{(l)} = \varphi(z^{(l)}) \odot (W^{(l+1)\top}\delta^{(l_1)})$
and $\nabla_{W^{(l)}} l = \delta^{(l)} v^{(l-1)\top}$ to calc the gradient.

Overfitting and Robustness

To avoid 0, ∗ grad. keep $\mathbb V$ of activation const.
Init W: tanh: $\mathcal N(\frac{1}{n_{in}}$ or $\frac{2}{n_{in}+n_{out}})$; relu: $\mathcal N(\frac{2}{n_{in}})$.
GD: η piecewise const. ↓ or w/ momentum.
Prevent Overfitting: • Dropout(Eval $\hat w = wp$)
• Regularization • Normalization • Early Stop

CNN and other architectures

CNN-Formulas: ***C***han., ***K***er. size, m = #Ker.
- Dim: $f(W)\times f(H)\times m, f(i) = \frac{i+2P-K_i}{S}+1$
- Params: $p = (K_W\cdot K_H\cdot C+1)\cdot m$, $+1 \hat=$ Bias

Pooling Layers: Pool units to decrease width.
ResNet: $v^{(l+1)} = v^{(l)} + r(v^{(l)})$ w/ skip conn.

Clustering / K-Means Problem

Problem.: Minimize $\sum\min_{j\in[k]}||x_i-\mu_j||_2^2$
Lloyd's heuristic: 1. Init μ_j 2. Assign x_i to closest μ_j 3. Set μ_j as mean of assigned points.
Conv. to local opt (exp.). $\mathcal O(nkd)$ per iter.
K-Means++: $\mu_1 = x_i$ with $i\sim\mathcal U\{1,\ldots,n\}$, then given $\mu_{1:j}$, pick $\mu_j+1 = x_i$ with prob. $p(i) \propto \min_{l\in[j]}||x_i-\mu_l^{(0)}||_2^2$. $\mathcal O(\log k)$ opt. sol.
Pick k by heuristics, regularization, etc.

Dimensionality Reduction

$w^* = \text{argmin}_{w,z,||w||_2=1}\sum_i||x_i - wz_i||_2^2$
$z_i^* = w^\top x_i \implies w^* = \text{argmin}_{||w||_2=1} w^\top\Sigma w$
With $\Sigma = \frac1n\sum_i x_ix_i^\top$ as the empirical covariance matrix (assuming $\mu=0$). Solution given by principal CV of Σ. (= max. empirical var.)
PCA problem ($k>1$): $w\to W$ s.t. $W^\top W = I$, $W = [v_1|\ldots|v_k]$ the k-first eigenvectors of Σ.

Repr. $z_i = W^\top x_i$. Recon. $\tilde x_i = WW^\top x_i$
PCA via SVD: $X = U\Sigma V^\top \to W = V_{:,1:k}$
Kernelized PCA: With $w = \sum\alpha_j\phi(x_j)$ and $\text{argmax}_{||w||=1} w^\top\Sigma w = \text{argmax}\, w^\top X^\top Xw \implies$
$\alpha^* = \text{argmax}_\alpha \frac{\alpha^\top K^\top K\alpha}{\alpha^\top K\alpha}$
With closed form solution (for any k):
$\alpha^{(i)} = \frac{1}{\sqrt{\lambda_i}}v_i$ from $K = \sum_i\lambda_iv_iv_i^\top, \lambda_1\ge\ldots\ge\lambda_n$.
$\implies z_i = \sum_j\alpha_j^{(i)}k(x_j,x)$ as projection.
Autoencoder: $W^* = \text{argmin}\sum||x_i - f_W(x_i)||_2^2$
Thus $f(x;\theta) = f_{\text{dec}}(f_{\text{enc}}(x;\theta_{\text{enc}});\theta_{\text{dec}})$ and if activation is identity and square loss ≡ PCA.

Probabilistic Modeling

Suppose we have access to $\mathbb P_{XY}$ then opt. sol.:
Reg, (SE): $\hat f(x) = \mathbb E[Y \mid X=x], Y = f^*(X)+\varepsilon$
C.$_{0-1}$: $\hat f(x) = \mathbb P_{Y|X}(Y\ne\text{sgn} f(X)), y = \varepsilon y^*(x)$
Get $\mathbb P(Y \mid X)$ from $\mathbb P(X,Y)$, but not vice versa.
Naive $\mathbb P_{XY}$ Est.: Kernel density est./histogram

Parametric Models for $\mathbb P_{XY}$

Best of distribution family $\mathscr P = \{\mathbb P_{XY};\theta\in\Theta\}$
MLE: Likelihood: $p(D;\theta) = \prod p(x_i;\theta)$ with its estimator $\theta_{\text{MLE}} = \text{argmax}\log p(D;\theta)$.

Discriminative $p(x,y) = p(y \mid x;\gamma)p(x;\pi)$

Ex. Reg. $X\sim\mathcal N(\mu,1), \mathbb P_{Y|x;w} = \mathcal N(w^\top x, 1)$:
1. $\hat\mu_{\text{MLE}} = \frac1n\sum x_i$ as sample mean for $\mathbb P_X$.
2. $\hat w_{\text{MLE}} = \text{argmin}\sum(y_i - w^\top x_i)^2$ for $\mathbb P_{Y|x;w}$.
3. $\hat p(x,y) = p(x;\hat\mu_{\text{MLE}})\cdot p(y \mid x;\hat w_{\text{MLE}})$

Ex. Cl. $X\sim\mathcal N(\mu,1), p(y \mid x;w) = \sigma(yw^\top x)$.
1. $\mu = \hat\mu_{\text{MLE}}$ 2. $\hat w_{\text{MLE}} = \text{argmin}\sum g_{\log}(y_iw^\top x_i)$

Generative $p(x,y) = p(x \mid y;\gamma)p(y;\pi)$

Setup Ex. $Y\sim\text{Cat}(\pi), \mathbb P_{X|y;\mu_y,\Sigma_y}\sim\mathcal N(\mu_y,\Sigma_y)$ with $\pi\in\Pi, \Sigma_y\in S$ and $y\in\{1,2\}$.
Gaus. Naïve Bayes: $\Sigma_y = \text{diag}[\sigma^2_{y,1},\ldots,\sigma^2_{y,d}]$.
1. $[\hat\pi]_j = \hat p_j = \frac{\#\{Y=j\}}{n}$ 2. $\hat\mu_y = \frac{1}{\#\{Y=y\}}\sum_{i:y_i=y}x_i$
3. $\hat\sigma_{y,k} = \frac{1}{\#\{Y=y\}}\sum_{i:y_i=y}(x_{i,k}-\mu_{y,k})^2$ w/ MLE.

GNB performs better for small sample sizes.
Has correct uncertainty for big samples. If iid: $\hat y = \text{argmax}\, p(y \mid x) = \text{argmax}\, p(y)\prod p(x_i \mid y)$.
GBC/QDA: Same as GNB, less restrictive:
$\hat\Sigma_y = \frac{1}{\#\{Y=y\}}\sum_{i:y_i=y}(x_i-\hat\mu_y)(x_i-\hat\mu_y)^\top$.
Linear Discriminant Analysis: $\forall y: \sigma_y = \sigma$

Bayesian Modeling

Ass. data iid. from $\mathbb{P}_{\cdot|\theta}$ with prior distribution $\theta \sim \mathbb{P}_\theta$. Then $p(D) = \int p(D \mid \theta) p(\theta)\, d\theta$.

MAP: Posterior: $p(\theta \mid D) = \frac{p(D|\theta)p(\theta)}{\int p(D|\theta)p(\theta)\,d\theta}$ & $\hat{\theta} = \operatorname{argmax} \log p(\theta \mid D) = \operatorname{argmax} \log p(D, \theta)$

Ex. Reg.: $y_i = w^\top x_i + \varepsilon_i$, $w \sim \mathcal{N}(0, \sigma_w^2 I_d)$, $\varepsilon \sim \mathcal{N}(0,1)$, $\mathscr{P} = \{\mathbb{P}_{Y|X;w} = \mathcal{N}(\langle w, x\rangle, 1)\}$.

$$\hat{w}_{\text{MAP}} = \operatorname{argmin} \tfrac{1}{2}||y - Xw||_2^2 + \tfrac{1}{2\sigma_w^2}||w||^2$$

$=$ ridge sol. If $p(w) = \frac{1}{z} e^{-\frac{||w||_1}{\sigma_w}}$ laplacian then

$$\hat{w}_{\text{MAP}} = \operatorname{argmin} \tfrac{1}{2}||y - Xw||_2^2 + \tfrac{1}{\sigma_w}||w||_1$$

which is the lasso sol. $\rightarrow \hat{\mathbb{P}}_{Y|X} = \mathbb{P}_{Y|X;\hat{w}_{\text{MAP}}}$.

Bayes. Model Avg: Gives distribution of f^*:

$$\hat{p}(y \mid x; D) = \hat{E}_{\theta|D} p(y \mid x; \theta)$$
$$= \int_\Theta p(y \mid x; \theta) \hat{p}(\theta \mid D)\, d\theta$$

Decision Theory

Decision rules $a : X \to A$, with A as action set. Find $a^*(x) = \operatorname{argmin} \hat{\mathbb{E}}[l(a(x), y) \mid X = x]$

Applications of decision theory w/ $\mathbb{P}(Y \mid X)$**:**
• Reg. SE: $\hat{f}(x) = \operatorname{argmin}_a \hat{\mathbb{E}}[(Y - a)^2 \mid X = x]$ $= \hat{E}[Y \mid X = x]$ • 0-1: $\hat{y}(x) = \operatorname{argmax}_y \hat{p}(y \mid x)$ $= \operatorname{argmin}_a \hat{E}[\mathbb{I}_{a \neq Y} \mid X = x]$ • a0-1: Boundary $\pi(x)$ to $\pi(x) = \frac{c_{FN}}{c_{FP} + c_{FN}}$ • Abstention 0-1: with $A = \{-1, +1, r\}$ and $l(\hat{y}, y) = \mathbb{I}_{\hat{y} \neq y} \mathbb{I}_{\hat{y} \neq r} + c\,\mathbb{I}_{\hat{y} = r}$ obtain $\hat{y} = r$ if $c < \hat{p}(y = -1 \mid x) < 1 - c$.

Summary (Gen. Classification)

1. Est. $p(y)$ 2. Est. $p(x \mid y)$ 3. Obtain $p(y \mid x)$

$$\hat{y} = \operatorname{argmax}_y p(y \mid x)$$
$$= \operatorname{argmax}_y \log p(y) + \log p(x \mid y)$$

Gaussian Mixture Models

We assume $p(x \mid \theta) = \sum_j w_j \mathcal{N}(x \mid \mu_j, \Sigma_j)$ and thus the optimization problem is defined as

$$\operatorname{argmin} -\sum_i \log \sum_j w_j \mathcal{N}(x_i \mid \mu_j, \Sigma_j)$$

Fitting a GMM $\equiv$ GBC without labels.

Hard-EM

E-Step: Predict most likely class for each x_i.

$$z_i^{(t)} = \operatorname{argmax}_z p(z \mid x_i, \theta^{t-1})$$
$$= \operatorname{argmax}_z p(z \mid \theta^{(t-1)}) p(x_i \mid z, \theta^{(t-1)})$$

M-Step: Compute MLE as for GBC.

Uniform w_j, identical spherical $\Sigma_j \Rightarrow$ k-means

Soft-EM

E-Step: Calc cluster membership weights:

$$\gamma_j^{(t)} = p(Z = j \mid x, \Sigma, \mu, w) = \frac{w_j p(x_i|\Sigma_j, \mu_j)}{\sum_l w_l p(x|\Sigma_l, \mu_l)}$$

M-Step: Fit cluster to weighted x_i (MLE):

$$w_j^{(t)} = \tfrac{1}{n}\sum_{i=1}^n \gamma_j^{(t)}(x_i) \qquad \mu_j^{(t)} = \frac{\sum_{i=1}^n x_i \cdot \gamma_j^{(t)}(x_i)}{\sum_{i=1}^n \gamma_j^{(t)}(x_i)}$$

$$\Sigma_j^{(t)} = \frac{\sum_{i=1}^n \gamma_j^{(t)}(x_i)(x_i - \mu_j^{(t)})(x_i - \mu_j^{(t)})^\top}{\sum_{i=1}^n \gamma_j^{(t)}(x_i)}$$

Hard-EM props. + variance $\to 0 \Rightarrow$ k-means. CV for j, maximize log-likelihood on val set.

EM for SSL

E-Step: For x_i with label y_i: $\gamma_j^{(t)}(x_i) = \mathbb{I}_{\{j = y_i\}}$.

GM Bayes Cl.: 1. Est. $\mathbb{P}_Y$ 2. Est.$p(x \mid y)$ via GMM 3. $p(y \mid x) = \frac{1}{z} p(y) p(x \mid y)$.

Density Est.: Anomaly detection/data imputation. Compare est. density of x_i against threshold τ (CV) $\to$ control estimated FPR.

General EM

E: expected sufficient statistic, *M*: MLE

E-Step: Calculate the expected complete data log-likelihood (function of θ):

$$Q(\theta; \theta^{(t-1)}) = \mathbb{E}_Z[\log p(X, Z \mid \theta) \mid X, \theta^{(t-1)}]$$
$$= \sum_i \sum_{z_i} \gamma_{z_i}(x_i) \log p(x_i, z_i \mid \theta)$$

W/ $\gamma_z(x) = p(z \mid x, \theta^{(t-1)})$, depends on $\theta^{(t-1)}$.

M-Step: Max. $\theta^{(t)} = \operatorname{argmax}_\theta Q(\theta; \theta^{(t-1)})$.

Equivalent to train a GBC with weighted data. Each EM-iteration increases data likelihood.

EM-Init: w unif, μ k-m++, Σ spherical (S^2)

Degeneracy: Loss $\to -\infty$ as $\mu \to x, \sigma \to 0$. Thus add $\nu^2 I$ to covariances (ν by CV). Same as adding a Wishart prior on Σ and calc. MAP.

Generative Modeling with NN

Model word $X_i \in [N]$ as categorical variable. $p(\text{Sentence}) = p(X_1, \dots, X_m) \to N^m - 1$ param.

Key idea: Estimate conditional distribution:

$$\mathbb{P}(X_t = x \mid X_{1:t-1} = x_{1:t-1})$$
$$\approx \mathbb{P}(X_t = x \mid X_{t-k:t-1} = x_{t-k:t-1}, \theta)$$
$$:= \text{Cat}(x \mid \text{softmax}(f(x_{t-k:t_1}, \theta)))$$

With f as NN with params θ. Use CE-Loss: $L(\theta) = \sum_t \log \mathbb{P}(X_t = x \mid X_{t-k:t-1} = x_{t-k:t-1}, \theta)$

Self-supervision: Use next word as label.

Simple transformer (decoder only)

Computational Model: $Z_0 = XW_e + W_p$ with $X = (x_{t-k}, \dots, x_{t-1}) \in \mathbb{R}^{k \times N}$ and W_e is (learnable *word embedding* matrix), W_p is a (fixed) *position embedding* matrix, Z_l = transformer block and $P = \text{softmax}(Z_n W_e^\top)$.

(Self-)Attention: Learn to predict a weighted, directed graph. $z_i^{l+1} = \sum_{j=1:k} \text{score}_{i,j} v_j^l$. Score measures directed similarity of word i to j. Self-attention needs both sides to be the same phrase. Each word has a "key" vector k_i, a "query" vector q_i and a "value" vector v_i all **predicted**. Then we can add masking, such that only attend to preceding words (adding $m_{i,j} = -\infty$ if $j > i$, else 0).

$$\text{score}_{i,j} = q_i^\top k_j \propto \frac{\exp(q_i k_j^\top / \sqrt{d_k} + m_{i,j})}{\sum_{j'} \exp(q_i k_{j'}^\top / \sqrt{d_k} + m_{i,j'})}$$

$$Z' := \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}} + M\right) V \quad \text{(SM rowwise)}$$

Right of $\propto$ is the normalized scaled dot product attention, to remove 0-gradients.

Multi-Head Attention: Use multiple queries, keys, values for each word (Q_h, K_h, V_h) each in $\mathbb{R}^{k \times d_v}$. Then concatenate to get single output $Z \in \mathbb{R}^{k \times (h \cdot d_v)}$.

In reality "tokens" are used instead of words (e.g. BPE: byte-pair encoding). Text generated from LLMs often is not directly useful, need "RL from Human Feedback".

Math Additions

Convexity:

0. $f(\lambda x + (1-\lambda) y) \leq \lambda f(x) + (1-\lambda) f(y)$
1. $f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle$
2. $D^2 f(x) \succeq 0$ (psd)

- $\alpha f + \beta g, \alpha, \beta > 0$ convex if f, g convex.
- $f \circ g$ convex if f convex, g affine or f non-decreasing, g convex.
- $\max(f, g)$ convex if f, g convex.

Derivatives (Denom. lay.): • $\frac{\partial}{\partial x} Ax = A^\top$
• $\frac{\partial}{\partial x} x^\top A = A$ • $\frac{\partial}{\partial x} \alpha = \vec{0}$ • $\frac{\partial}{\partial x} x^\top a = \frac{\partial}{\partial x} a^\top x = a$
• $\frac{\partial}{\partial x} b^\top X x = A^\top b$ • $\frac{\partial}{\partial x} x^\top A x = (A + A^\top) x$
• $\frac{\partial}{\partial x} x^\top x = 2x$ • $\frac{\partial}{\partial x} ||y - Xx||_2^2 = 2X^\top (Xx - y)$

Density of $\mathcal{N}(\mu, \Sigma)$**:**

$$p(x \mid y; \mu_y, \Sigma_y) = \frac{1}{(2\pi)^{\frac{d}{2}} (\det \Sigma_y)^{\frac{1}{2}}} e^{-\frac{(x - \mu_y)^\top \Sigma_y^{-1} (x - \mu_y)}{2}}$$

Shortcuts, Tips and Tricks

Covariances and PCA: $\frac{1}{n} \sum_{i=1}^n x_i x_i^\top = \frac{1}{n} X^\top X$. Let $\lambda_1 \geq \ldots \geq \lambda_d \geq 0$ denote eigenvalues of $\frac{1}{n} X^\top X$ (spd/sym) and σ_i denote i-th singular value of X, then $\lambda_i = \sigma_i^2 / n$. $L(k) = \sum_{j=k+1}^d \lambda_j$. If $\text{Cov}(X, Y) > 0$, then data: $\nearrow$, < 0: $\swarrow$. $\text{Cov}(X, Y) = \mathbb{E}((X - \mathbb{E}(X))(Y - \mathbb{E}(Y))^\top)$ $\mathbb{V}(WX) = W\, \mathbb{V}(X) W^\top$

Trace Tr: • Linear • $\text{Tr}(ABCD) = \text{Tr}(DABC)$ • $\text{Tr}(A) = \sum_i \lambda_i$ • $\text{Tr}(XX^\top) = \sum_{i,j} X_{i,j}^2 = ||X||_2^2$

Kernels: Valid: • $\frac{1}{1 - xy}$ • 2^{xy} • $e^{k(x,y)}$ • $\cos(x - y)$ • $\min(x, y)$ • $\frac{\min(x,y)}{\max(x,y)}$ • $g(x) k(x, y) g(y)$ **Invalid:** • $\max(x, y)$ • $f(k(x, y))$, f any poly. • $\cos(x + y)$

MLE: • $\hat{p}_{poi} = \hat{\mu}_{\mathcal{N}} = \frac{\sum x_i}{n}$ • $\hat{\lambda}_{exp} = \hat{p}_{geo} = \frac{n}{\sum x_i}$ • $\hat{p}_{bin} = \frac{1}{N} \frac{\sum_{i=1}^N x_i}{n}$ • $\hat{\sigma}_{\mathcal{N}} = \frac{1}{n} \sum (x_i - \hat{\mu}_{\mathcal{N}})^2$

KL-Divergence: Divergence between reference distribution P and another distribution Q.

$$D_{KL}(P \parallel Q) := \mathbb{E}_{X \sim P}[\log \tfrac{p(X)}{q(X)}]$$
$$= \int_{\mathbb{R}} p(x) \log \tfrac{p(x)}{q(x)} dx$$