ML and Opt Lecture 6

- Cross-entropy loss for fitting multiclass logistic regression
- Train/validation/test splits for model fitting, selection, and generalization reporting
- Linear algebra primer

MLE for Categorical Model

$$y|x \sim \text{Categorical}\left(p_1(x), \cdots, p_K(x)\right)$$

where $p_i(x) = \dfrac{e^{\theta_i^T x}}{Z_\theta(x)}$ where $Z_\theta(x) = \sum_{j=1}^{K} e^{\theta_j^T x}$

recall $p_\theta(x) = \begin{bmatrix} p_1(x) \\ \vdots \\ p_K(x) \end{bmatrix} = \dfrac{e^{\theta x}}{1^T e^{\theta x}} = \text{softmax}(\theta x)$

where $\theta = \begin{bmatrix} \theta_1^T \\ \vdots \\ \theta_K^T \end{bmatrix}$

Given $\{(y_i, x_i)\}_{i=1}^n$, where $y_i \in \mathbb{R}^k$ one-hot encodes the class of the ith example (e.g. class 1 encodes as $\begin{bmatrix}1\\0\\\vdots\\0\end{bmatrix}$ and class k as $\begin{bmatrix}0\\0\\\vdots\\1\end{bmatrix}$), our goal is to learn the model parameters $\theta \in \mathbb{R}^{k \times d}$ using MLE

$$\mathcal{L}(\theta) = \frac{1}{n} \sum_{i=1}^{n} -\log p_\theta(y_i | x_i)$$

Recall

$$p_\theta(y_i = e_j | x_i) = e_j^T p_\theta(x_i) = y_i^T p_\theta(x_i)$$

$$= y_i^T \text{softmax}(\theta x_i)$$

So

$$\mathcal{L}(\theta) = \frac{1}{n}\sum_{i=1}^{n} -\log p_\theta(y_i|x_i)$$

$$= \frac{1}{n}\sum_{i=1}^{n} -\log\left[y_i^T \operatorname{softmax}(\theta x_i)\right]$$

$$= \frac{1}{n}\sum_{i=1}^{n} -y_i^T \log\left[\operatorname{softmax}(\theta x_i)\right]$$

apply log entrywise to its vector argument

$$\overset{?}{=} \frac{1}{n}\sum_{i=1}^{n} \ell\left(\operatorname{softmax}(\theta x_i), y_i\right)$$

what is the correct choice of ℓ to make the NLL have the form of an empirical risk

Guess: ℓ is the cross-entropy loss

What is cross-entropy? Easiest to first define KL-divergence

$$D_{KL}(p \| q) = \sum p_i \ln \frac{p_i}{q_i}$$ if p and q are two probability vectors

$$= \underbrace{\sum p_i \ln p_i}_{\text{Entropy of } p \text{, } H(p)} \underbrace{- \sum p_i \ln q_i}_{\text{cross-entropy of } p \text{ and } q \text{, } H(p,q)}$$

So minimizing $H(p,q)$ w.r.t. q minimizes $D_{KL}(p \| q)$ w.r.t. q

Recall the negative log-likelihood w.r.t. the ith sample

$$-y_i^T \log(\text{softmax}(\theta x_i))$$

$$= \sum_{j=1}^{k} (y_i)_j \ln(p_\theta(x_i)_j)$$

$$= H(y_i, p_\theta(x_i))$$

Note that if the ith example is from class j, so $y_i = e_j$, then

$$-y_i^T \log(\text{softmax}(\theta x_i)) = -e_j^T \ln(p_\theta(x_i))$$

$$= -\ln(p_\theta(x_i)_j)$$

is the log of the probability assigned to the true class

Upshot is that

$$\mathcal{L}(\theta) = \frac{1}{n}\sum_{i=1}^{n} H(y_i, \text{softmax}(\theta x_i))$$

$$= \frac{1}{n}\sum_{i=1}^{n} -\ln\left[p_\theta(x_i)_{y_i}\right]$$

So minimizing the NLL learns model parameters $\hat{\theta}$ that maximize the probabilities assigned by the model to the true classes of the observations, and the loss function for this ERM is

$$\ell(u, v) = H(u, v)$$

The optimization problem is

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}}\ \mathcal{L}(\theta)$$

$$= \underset{\theta}{\operatorname{argmin}}\ \frac{1}{n}\sum_{i=1}^{n} -y_i^T \ln\left(\operatorname{softmax}(\theta x_i)\right)$$

$$= \underset{\theta}{\operatorname{argmin}}\ \frac{1}{n}\sum_{i=1}^{n} -y_i^T \ln\left(\frac{e^{\theta x_i}}{Z_\theta(x_i)}\right)$$

$$= \underset{\theta}{\operatorname{argmin}}\ \frac{1}{n}\sum_{i=1}^{n} -y_i^T\left[\ln\left(e^{\theta x_i}\right) - \left(\ln Z_\theta(x_i)\right)1\right]$$

$$= \underset{\theta}{\operatorname{argmin}}\ \frac{1}{n}\sum_{i=1}^{n}\left[-y_i^T\theta x_i + y_i^T 1 \ln Z_\theta(x_i)\right]$$

$$= \underset{\theta}{\operatorname{argmin}} \; \frac{1}{n} \sum_{i=1}^{n} \left[-y_i^T \theta x_i + \ln Z_\theta(x_i) \right]$$

$$= \underset{\theta}{\operatorname{argmin}} \; \frac{1}{n} \sum_{i=1}^{n} -y_i^T \theta x_i + \frac{1}{n} \sum_{i=1}^{n} \ln Z_\theta(x_i)$$

$$= \underset{\theta}{\operatorname{argmin}} \; \frac{1}{n} \sum_{i=1}^{n} -y_i^T \theta x_i + \frac{1}{n} \sum_{i=1}^{n} \ln \left(\sum_{j=1}^{k} e^{\theta_j^T x_i} \right)$$

This is the optimization problem we solve in practice to learn θ using MLE for multiclass classification.

Practical Considerations

train vs test vs validation splits

use training data to solve our ERM, that is

$$\hat{f} = \underset{f \in \mathcal{F}}{\operatorname{argmin}} \frac{1}{n_{train}} \sum_{i=1}^{n_{train}} \ell(f(x_i), y_i)$$

$$= R_{n_{train}}(f)$$

use fresh samples to estimate the risk

$$R(\hat{f}) = \mathbb{E}_{\mathcal{D}}\, \ell(\hat{f}(x), y)$$

$$\approx \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} \ell(f(x_i), y_i)$$

We use the test set to estimate the "generalization error"

$$R(\hat{f}) - R_{n_{train}}(\hat{f}) \approx R_{n_{test}}(\hat{f}) - R_{n_{train}}(\hat{f})$$

What if we want to choose between different model classes, e.g. $\mathcal{F}_1$ and $\mathcal{F}_2$?

- use training set to find best (empirically) models $\hat{f}_1 \in \mathcal{F}_1$ and $\hat{f}_2 \in \mathcal{F}_2$
- use validation dataset to select best of $\hat{f}_1$ and $\hat{f}_2$ on a fresh independent set of samples
- estimate the generalization gap of this selected model on a fresh independent test set

General flow for supervised ML

- Fix our problem domain, risk (loss) measure, collect appropriate data $\{(x_i, y_i)\}_{i=1}^{n}$
- Split our data into train/validation/test (60/20/20)
- Determine ahead of time a set models and hyperparameters to try
- Fit each of these using the same training data
- Measure their performance using the validation data and select the best, f_{opt}
- Report an estimate of the gen gap / pop risk of f_{opt} using the test data

Lin Alg

vectors in $\mathbb{R}^d$, $\mathbb{R}^n$ — column vectors

linear (in)dependence

$\mathbb{R}^2$

v and u are linearly dependent, because
$v = \alpha u$

v and z are linearly independent, because
$v \neq \alpha z$ for any α

In general, vectors $\{v_1, \dots, v_k\} \subseteq \mathbb{R}^n$ are linearly dependent if there exist α_i not all zero so that

$$\alpha_1 v_1 + \dots + \alpha_k v_k = 0$$

if on the other hand

$$\alpha_1 v_1 + \dots + \alpha_k v_k = 0 \text{ iff } \alpha_i = 0 \text{ for all } i$$

then the vectors are linearly independent

Write $V = \begin{bmatrix} v_1 & \cdots & v_k \end{bmatrix} \in \mathbb{R}^{n \times k}$

the vectors are independent iff

$$V\alpha = 0 \iff \alpha = 0$$

i.e. V has full column rank (rank k)

Spans

$$\text{span}\{v_1, \cdots, v_k\} = \{x : x = \alpha_1 v_1 + \cdots + \alpha_k v_k \text{ for some } \alpha_1, \cdots \alpha_k\}$$

$$= \{x : x = V\alpha \text{ for } \alpha \in \mathbb{R}^k\}$$

this is a linear subspace, with dimension given by column rank of V

bases

Given a subspace $\Pi \subseteq \mathbb{R}^d$, the vectors $\{v_1, \dots, v_k\}$ is a basis for Π if:

1) $\Pi = \text{span}\{v_1, \dots, v_k\}$

2) $\{v_1, \dots, v_k\}$ are linearly independent

dot product / inner product

$u, v \in \mathbb{R}^d \qquad \langle u, v \rangle = u^T v = \sum_{i=1}^{d} u_i v_i$

Euclidean / ℓ_2 / taxo norm $\qquad \|u\|_2^2 = \langle u, u \rangle = \sum_{i=1}^{d} u_i^2$

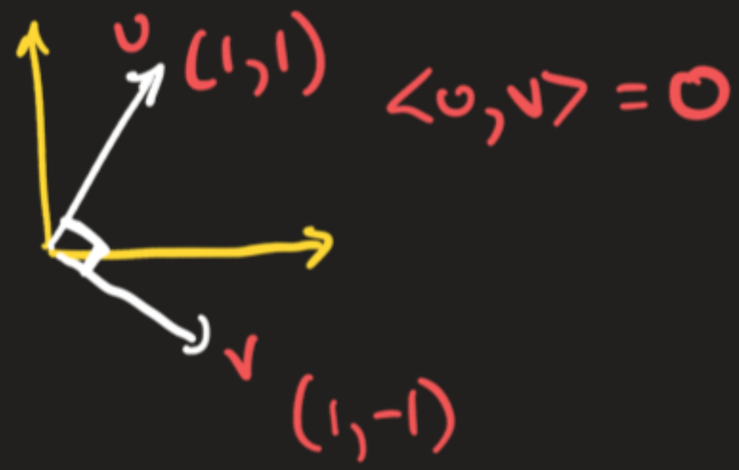
Cauchy-Schwarz inequality

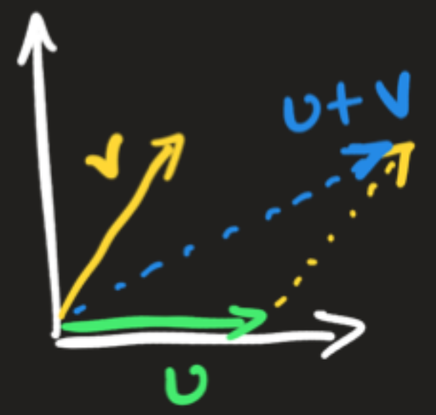
Fact

$|\langle v, w\rangle| \leq \|v\|_2 \|w\|_2$ and equality occurs iff $v = \alpha w$

$\cos\theta = \dfrac{\langle v, w\rangle}{\|v\|_2 \|w\|_2} \in [-1, 1]$ by Cauchy-Schwarz, so we use this to define the angle between vectors in high-dimensional spaces

$v \perp w \iff \langle v, w\rangle = 0 \iff \cos\theta = 0$





$$\|u+v\|_2^2 = \langle u+v,\, u+v\rangle$$

$$= \langle u,\, u+v\rangle + \langle v,\, u+v\rangle$$

$$= \langle u,u\rangle + \langle u,v\rangle + \langle v,u\rangle + \langle v,v\rangle$$

$$= \|u\|_2^2 + \|v\|_2^2 + 2\langle u,v\rangle$$

$$= \|u\|_2^2 + \|v\|_2^2 + 2\cos\theta\, \|u\|_2 \|v\|_2$$

high-dimensional law of cosines

If $\langle u,v\rangle = 0$, then

$$\|u+v\|_2^2 = \|u\|_2^2 + \|v\|_2^2$$

Pythagorean theorem

outer product

$u \in \mathbb{R}^n$, $v \in \mathbb{R}^k$ then outer product

$$uv^T = \begin{bmatrix} | \\ u \\ | \end{bmatrix} \begin{bmatrix} — & v^T & — \end{bmatrix} \in \mathbb{R}^{n \times k} = \begin{pmatrix} u_1 v_1 & \cdots & u_1 v_k \\ \vdots & & \vdots \\ u_n v_1 & \cdots & u_n v_k \end{pmatrix}$$

$$(uv^T)_{ij} = u_i v_j$$

$U \in \mathbb{R}^{n \times k}$, $V \in \mathbb{R}^{d \times k}$ $\Rightarrow$ $UV^T = \sum_{i=1}^{k} u_i v_i^T \in \mathbb{R}^{n \times d}$

$$\begin{bmatrix} | & & | \\ u_1 & \cdots & u_k \\ | & & | \end{bmatrix} \quad \begin{bmatrix} | & & | \\ v_1 & \cdots & v_k \\ | & & | \end{bmatrix}$$

symmetric matrices

A square matrix M is symmetric if $M = M^T$

$\Leftrightarrow \quad M_{ij} = M_{ji}$ for all i, j

eigenpairs

(λ, v) is an eigenpair of $M \in \mathbb{R}^{n \times n}$ ($\lambda \in \mathbb{R}$ is an eigenvalue, $v \in \mathbb{R}^d$ is an eigenvector)

iff

$$Mv = \lambda v$$

Fact:

1) all eigenvectors of symmetric matrices are perpendicular to each other
2) all eigenvalues of symmetric matrices are real
3) we can write $M = \sum_{i=1}^{n} \lambda_i v_i v_i^T$ where (λ_i, v_i) are eigenpairs and $\|v_i\| = 1$

Note: if we write

$$V = \begin{bmatrix} v_1 & \cdots & v_n \end{bmatrix} \in \mathbb{R}^{n \times n}$$

collection of (length 1) eigenvectors of M

$$\Lambda = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} \in \mathbb{R}^{n \times n}$$

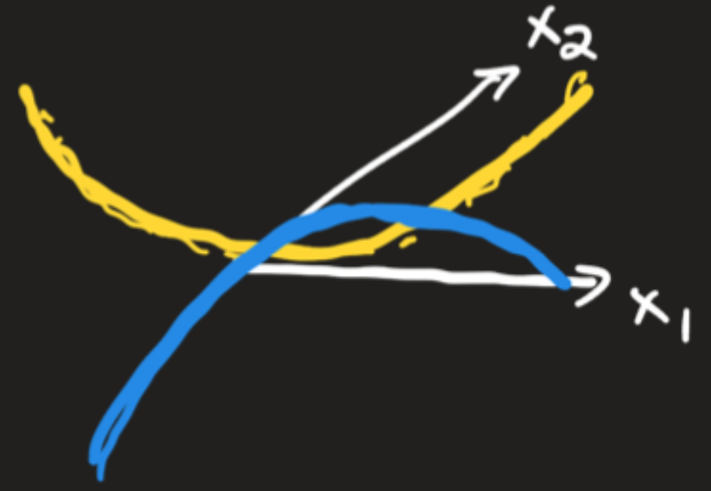
a diagonal matrix w/ corresponding eigenvalues

$$M = V \Sigma V^T$$

Geometric Interpretation

Associate each symmetric matrix M with a quadratic form $m : x \longmapsto x^T M x = \sum_{i=1}^{n} \sum_{j=1} x_i x_j M_{ij}$

$$x^T \begin{bmatrix} 1 & \\ & -1 \end{bmatrix} x = x_1^2 - x_2^2$$



vs $x^T \begin{bmatrix} 1 & \\ & 1 \end{bmatrix} x = x_1^2 + x_2^2$

Geometrically, $M = \sum_{i=1}^{n} \lambda_i v_i v_i^T$, the quadratic form curves upwards in the directions in the span of v_i for which the λ_i are positive, and downward in directions for which λ_i are negative

Positive Semidefiniteness

We say a symmetric matrix $A = \sum \lambda_i v_i v_i^T$ is positive semidefinite (PSD) or positive, written $A \succeq 0$ if $\lambda_i \geq 0$ for all i

positive definite, written $A \succ 0$ if $\lambda_i > 0$ for all i

Facts

1) A is psd iff the corresponding quadratic form $x^T A x \geq 0 \quad \forall x$
2) if A is psd then $CAC^T \succeq 0$
3) For any matrix $AA^T \succeq 0$ and $A^T A \succeq 0$