Remembrancer

Eugene Francisco

January 2026

remembrancer
noun
A person with the job or responsibility of reminding others of something; a chronicler.

A collection of helpful things to reference quickly. Mostly from Stats315A, CS229, and EE364A.

Probability Bounds

Most copied from John Duchi's probability notes.

Markov's Inequality: Let Z be a non-negative R.V. Then $\mathbb{P}(Z \geq t) \leq \mathbb{E}(Z)/t$.

Chebyshev's Inequality: Let Z be any random variable with finite variance. Then

$$\mathbb{P}(|Z - \mathbb{E}(Z)| \geq t) \leq \mathrm{var}(Z)/t^2 \qquad \text{for } t \geq 0.$$

Hoeffding's Inequality: Let $Z_1, \ldots, Z_n$ be independent bounded r.v. with $Z_i \in [a, b]$ for all i. Then

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n}(Z_i - \mathbb{E}(Z_i)) \geq t\right) \leq \exp\left(-\frac{2nt^2}{(b-a)^2}\right)$$
$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n}(Z_i - \mathbb{E}(Z_i)) \leq -t\right) \leq \exp\left(-\frac{2nt^2}{(b-a)^2}\right)$$

for all $t \geq 0$.

Some forms of Cauchy Schwarz:

$(\mathbb{E}(XY))^2 \leq \mathbb{E}(X^2)\mathbb{E}(Y^2)$.

$\mathrm{cov}(X, Y)^2 \leq \mathrm{var}(X)\,\mathrm{var}(Y)$.

$(\int f(x)g(x)dx)^2 \leq \int f(x)^2dx \int g(x)^2dx$.

$(\sum a_i b_i)^2 \leq (\sum a_i^2)(\sum b_i^2)$

Probability facts

Useful Gaussian Facts Let $X \sim \mathcal{N}(0, \sigma^2)$. Then $\mathbb{E}(X^4) = 3\sigma^4$ and $\mathbb{E}(|X|) = \sigma\sqrt{\frac{2}{\pi}}$.

Interchanging Expectation and Limits

Monotone Convergence Theorem: If X_t is a sequence of random variables so that for all ω, we have

(i) $X_j(\omega) \leq X_t(\omega)$ when $j \leq t$ and

(ii) $X_t(\omega) \to X(\omega)$ as $t \to \infty$,

then

$$\lim_{t\to\infty} \mathbb{E}(X_t) = \mathbb{E}(X)$$

Dominated Convergence Theorem: Let X_t a sequence of random variables. If there exists a *dominating function* g with finite expectation such that for all ω, we have

(i) *pointwise convergence* $X_t(\omega) \to X(\omega)$ as $t \to \infty$ and

(ii) *dominance* $|X_t(\omega)| \leq g(\omega)$ for all t,

then

$$\lim_{t\to\infty} \mathbb{E}(X_t) = \mathbb{E}(X)$$

Info Theory

Fisher Information: The Fisher information $I(\theta^*)$ of a random variable X is

$$\begin{aligned} I(\theta^*) &= -\ell''(\theta^*) \\ &= \int_{\mathbb{R}} \frac{[f'(x|\theta)]^2}{f(x|\theta)} dx \\ &= \text{var}_{X\sim\theta^*}\left[\frac{f'(X|\theta)}{f(X|\theta)}\right] \\ &= \text{var}_{X\sim\theta^*}\left[\frac{d}{d\theta}\log f(X|\theta)\Big|_{\theta=\theta^*}\right] \\ &= \mathbb{E}_{X\sim\theta^*}\left[\left(\frac{d}{d\theta}\log f(X|\theta)\Big|_{\theta=\theta^*}\right)^2\right] \end{aligned}$$

where the last form is the most recognizable: the variance (under θ^*) of the score function evaluated at θ^*.

Entropy: The entropy of $p(x)$ is

$$H(p) = \mathbb{E}_{x\sim p}\left[\log \frac{1}{p(x)}\right].$$

Can be thought of as the expected number of bits to optimally encode an event. Roughly the expected number bits needed for Huffman encoding.

KL Divergence: The KL divergence between $q(x)$ and $p(x)$ is

$$\begin{aligned}\mathrm{KL}(q(x) \parallel p(x)) &= \mathbb{E}_{x\sim q(x)}\left[\log \frac{q(x)}{p(x)}\right] \\ &= \mathbb{E}_{x'\sim q(x)}\left[\log \frac{1}{p(x')}\right] - H(q) \\ &= \underbrace{-\mathbb{E}_{x'\sim q(x)}[\log p(x')]}_{\text{cross entropy of } q \text{ relative to } p} - H(q).\end{aligned}$$

The first line is the standard definition. The second line is revealing: purple is the expected bit length to encode events coming from q using an optimal encoding for p; orange is the expected bit length of messages coming from q using an optimal encoding for q. So the second line is "*how much worse is the encoding for p than the encoding for q at sending messages from q.*" The third line is helpful for the remarks below.

Minimizing $\mathrm{KL}(q \parallel p_\theta)$ w.r.t θ is easy as long as

1. $q(x)$ is easy to sample from (doesn't have to be known)
2. $p_\theta(x)$ is easy to compute.

This is because you can take a sample $x \sim q(x)$ and calculate log-likelihood under p to estimate the cross entropy. And red doesn't matter because it doesn't depend on p.

Helpful Objectives

Cross Entropy Loss

Let $p_\theta(x)$ be a predicted probability distribution and $q(x)$ the true distribution. Then the cross entropy loss is

$$\mathrm{CE}(q(x), p_\theta(x)) = -\mathbb{E}_{x'\sim q(x)}[\log p_\theta(x')].$$

By the remarks above about KL divergence, minimizing cross entropy is the same as minimizing $\mathrm{KL}(q \parallel p_\theta)$. The unbiased estimate of this for training is typically

1. Sample $x \sim q(\cdot)$.
2. Calculate $\log p_\theta(x)$.

Big Idea: Minimizing the cross entropy of q relative to p_θ, $\mathrm{CE}(q, p_\theta)$, w.r.t θ is just minimizing the KL divergence between q and p_θ.

Fitting a quantile (pinball):

Let y a random variable and

$$L_\tau^{\text{pinball}}(q_\tau, y) = \begin{cases} \tau(y - q_\tau) & : \text{if } y \geq q_\tau \\ (1-\tau)(q_\tau - y) & : \text{if } y < q_\tau. \end{cases}$$

The q_τ which minimizes the expected loss is $q_\tau^* =$ the τth quantile of y.

Probability scoring

Scores are useful specifically in the case where we are trying to estimate $q(y|x)$ directly, typically via a parametrized $p_\theta(y|x)$ which we hope is close to $q(y|x)$. **This means that, when we are using scores, θ ends up parametrizing a probability distribution of $y|x$.**

Scores give us a way to quantify how close $p(y|x)$ is to $q(y|x)$. In particular, we first define $s(p, y)$ to be some quantification of "how surprising is y under p" (higher number is worse). Then we define

$$s(p, q) = \mathbb{E}_{y\sim q}[S(p, y)].$$

In other words, "on average, how surprising is $x \sim q$ under p." If p and q are very similar, we would expect that the average surprise of y is quite low. This gives intuition for the following definition.

Proper Scores: A score s is *proper* if

$$q = \underset{p}{\operatorname{argmin}}\, S(p, q).$$

Log score for probability forecasts:

Great for density representation. $S(p, y) = -\log(p(y))$. Strictly proper because

$$S(p, q) - S(q, q) = \int \log \frac{q(y)}{p(y)} q(y) dy$$

is the KL divergence (non-negative).

Quantile Score:

If the forecast is given as a collection of quantiles $q_\tau, \tau \in \mathcal{T}$[1], quantile score is

$$S_{\mathcal{T}}(\{q_\tau\}_{\tau\in\mathcal{T}}, y) = \sum_{\tau\in\mathcal{T}} L_\tau^{\text{pinball}}(q_\tau, y)$$

Intuition: $\tau \in [0, 1]$ and we want q_τ to be a threshold such that $y \sim Q$ is below u around τ of the time and above around $1 - \tau$ of the time.

[1] The τ here are the numbers between $0, 1$ and the q_τ are the forecasted quantiles.

T-test for Cross Val

Due to Nadeau and Bengio 2003. Suppose K-fold CV has gives us observed out of sample risk estimates $\{\widehat{R}_{f_i}\}_{i=1}^k$ and $\{\widehat{R}_{g_i}\}_{i=1}^k$. The r.v. $\widehat{R}_{f_i}$ are unbiased, identically distributed, but likely not independent since they will share training data.[2] Suppose we want to know the distribution of $K^{-1}\sum_{i=1}^K(\widehat{R}_{f_i} - \widehat{R}_{g_i})$ The corrected t-test variance of this sample mean is

$$\begin{aligned}\sigma^2_{\text{corrected}} &= \left(\frac{1}{K} + \frac{n_{\text{test}}}{n_{\text{train}}}\right)\hat{\sigma}^2 \\ &\approx \text{var}\left(\frac{1}{K}\sum_{i=1}^K(\widehat{R}_{f_i} - \widehat{R}_{g_i})\right).\end{aligned}$$

Convex Optimization:

Dual as lower bound.

Suppose we have the task

$$\begin{aligned}\text{Minimize} \quad & f(x) \\ \\ \text{subject to} \quad & f_i(x) \leq 0 \qquad i \in \{1,\dots,n\} \\ & h_i(x) = 0 \qquad i \in \{1,\dots,m\}.\end{aligned}$$

Let p^* be the optimal value of the problem. Then $p^* \geq q^*$ where q^* is the optimal value of the dual problem. What is the dual problem? First,

$$\begin{aligned}\mathcal{L}(x,\lambda,\nu) = f(x) + \sum_{i=1}^n \lambda_i f_i(x) + \sum_{i=1}^m \nu_i h_i(x) \\ g(\lambda,\nu) = \inf_x \mathcal{L}(x,\lambda,\nu).\end{aligned}$$

Then, the dual problem is

$$\begin{aligned}\text{Maximize} \quad & g(\lambda,\nu) \\ \\ \text{subject to} \quad & \lambda \succeq 0.\end{aligned}$$

The optimal value of the dual, q^*, is a lower bound for p^*. And g is always concave.

KKT conditions.

For a standard setup (not necessarily convex) problem, the KKT conditions on primal and dual variables x, λ, ν

(i) *Primal constraints:* $f_i(x) \leq 0$ for $i \in \{1,\dots,m\}$ and $h_i(x) = 0$ for $i \in \{1,\dots,p\}$.

(ii) *Dual constraints:* $\lambda \succeq 0$.

(iii) *Complementary slackness:* $\lambda_i f_i(x) = 0$ for $i \in \{1,\dots,m\}$.

(iv) *Gradient of Lagrangian w.r.t x is 0:*

$$\nabla_x \mathcal{L}(x,\lambda,\nu) = 0.$$

If $x^\star, \lambda^\star, \nu^\star$ are optimal, the KKT conditions will hold. Moreover, when the problem is convex, these KKT conditions hold if and only if $x^\star, \lambda^\star, \nu^\star$ are optimal. Similarly, if Slater's condition[3] holds for the problem, then KKT holds if and only if $x^\star, \lambda^\star, \nu^\star$ are optimal.

[2] The randomness in the R.V. comes from the training data $\mathcal{T}$ and the choice of evaluation fold, and possibly stochasticity in the learning method.

[3] This means there is a strictly feasible point. I.e., there exists x s.t. $f_i(x) < 0$ for $i \in \{1,\dots,m\}$.

cvxypy reminders:

```
import numpy as np
import cvxpy as cp

A = np.random.randn(10, 5)
b = np.random.randn(10, 1)
x = cp.Variable((5, 1))
constraints = [A @ x <= b]
obj = cp.Minimize(cp.norm2(x)) # Make sure this is DCP
problem = cp.Problem(constraints, obj)
problem.solve()
print(problem.value)
print(x.value)
```

Cool Learning Theory Stuff

Mostly taken from Bach.[4]

Risk and Error Definitions:

Definition: True risk and empirical risk respectively:

$$\mathcal{R}(f) = \mathbb{E}_{x,y\sim\mathbb{P}}[\ell(f(x), y)] \qquad \widehat{\mathcal{R}}(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i), y_i)$$

where $x_i, y_i \overset{\text{iid}}{\sim} \mathbb{P}$.

Tail bound on true vs empirical Risk:

Let $\mathcal{F}$ be a family of functions from X to Y. Suppose that $\ell(y, f(x))$ is almost surely in $[0, \ell_\infty]$. Let $\mathcal{R}$ be the model risk and $\widehat{\mathcal{R}}$ be the empirical risk on any independent data (over same $\mathbb{P}$ as $\mathcal{R}$). Then

$$\sup_{f\in\mathcal{F}} |\mathcal{R}(f) - \widehat{\mathcal{R}}(f)| \leq 2R_n(\mathcal{H}) + \frac{\ell_\infty}{\sqrt{2n}}\sqrt{\log\frac{2}{\delta}}$$

with probability greater than $1-\delta$. Here $\mathcal{H} = \{(x, y) \to \ell(y, f(x)) : f \in \mathcal{F}\}$ and R_n is the Rademacher complexity. In particular, for $\hat{f}$ (the minimizer in $\mathcal{F}$), we have

$$\mathcal{R}(\hat{f}) \leq \widehat{\mathcal{R}}(\hat{f}) + 2R_n(\mathcal{H}) + \frac{\ell_\infty}{\sqrt{2n}}\sqrt{\log\frac{2}{\delta}}. \tag{4.23}$$

Risk Tail Bound applied to structural minimization:

Consider m families $\mathcal{F}_1, \ldots, \mathcal{F}_m$ with associated empirical risk minimizers $\hat{f}_1, \ldots, \hat{f}_m$. Weight each $\mathcal{F}_i$ with a prior π_i ($\sum \pi_i = 1$). Choose $\hat{\imath}$ to be the i which minimizes the upper bound in (4.23). Then

$$R(\hat{f}_{\hat{\imath}}) \leq \min_{i\in\{1,\ldots,m\}} \left\{ \inf_{f_i\in\mathcal{F}_i} \mathcal{R}(f_i) + 4R_n(\mathcal{H}_i) + 2\frac{\ell_\infty}{\sqrt{2n}}\sqrt{\log\frac{1}{\pi_i}} \right\} + 2\frac{\ell_\infty}{\sqrt{2n}}\sqrt{\log\frac{2}{\delta}}.$$

In other words, the risk of the best thing we choose is upper bounded by the other model risks with a penalty for model complexity and the prior π_i.

[4] *Learning Theory from First Principles* Francis Bach.

Risk Tail Bound for val set minimization:

Same setup as above with m models and priors. Fix a proportion ρ of the training data for validation. Let $\hat{f}_i$ be the minimizer of the risk *on the training data* ($(1-\rho)n$ data points). Choose $\hat{\imath}$ which minimizes

$$\widehat{\mathcal{R}}_{\rho n}^{\text{validation}}(\hat{f}_i) + \frac{\ell_\infty}{\sqrt{2\rho n}}\sqrt{\log\frac{1}{\pi_i}}$$

where $\widehat{\mathcal{R}}_{\rho n}^{\text{validation}}(\hat{f}_i)$ is the empirical risk evaluated over only the validation set. Then (using Hoeffding's inequality w.r.t. randomness of val set, for which each $\hat{f}_i$ is deterministic) and the union bound, we can get with prob greater than $1-\delta$

$$\mathcal{R}(\hat{f}_{\hat{\imath}}) \le \min_{i\in\{1,\dots,m\}} \mathcal{R}(\hat{f}_i) + \frac{\ell_\infty}{\sqrt{2\rho n}}\sqrt{\log\frac{2}{\pi_i}} + \frac{\ell_\infty}{\sqrt{2\rho n}}\sqrt{\log\frac{2}{\delta}}.$$

Risk tail bound when using Holdout set:

For a classifier f on binary classification task with $0,1$ loss, as long as

$$n \ge \frac{\log(2/\delta)}{2\epsilon^2},$$

then with probability $1-\delta$,

$$|R_S[f] - R[f]| \le \epsilon.$$

Here $R(f)$ is the true risk and $R_S(f)$ is empirical risk from a holdout set that is not used for training. Can be generalized with Hoeffding's inequality.

Cross Validation Asymptotic Distribution:

Suppose A satisfies the following assumption. There are constants $0 < C_- \le C_+ < \infty$ and $0.25 < \gamma < 0.5$ such that, when trained on n samples $\{X_i, Y_i\}_{i=1}^n$, the excess risk $\hat{\mu}(\cdot)$ scales as

$$\lim_{n\to\infty} \mathbb{P}\left(n^\gamma \mathbb{E}\left[(\hat{\mu}(X)-\mu(X))^2 | \{X_i,Y_i\}\right]^{1/2} \le C_-\right) = 0$$
$$\lim_{n\to\infty} \mathbb{P}\left(n^\gamma \mathbb{E}\left[(\hat{\mu}(X)-\mu(X))^2 | \{X_i,Y_i\}\right]^{1/2} \le C_+\right) = 1.$$

Then

$$\sqrt{n}\left(\widehat{CV}_{n,K}(A) - \text{Err}^\star\right) \xrightarrow{d} \mathcal{N}\left(0, \text{var}\left[(Y-\mu(X))^2\right]\right).$$

Also

$$\frac{\widehat{CV}_{n,K}(A') - \widehat{CV}_{n,K}(A)}{\Delta^2_{n,K}(A') - \Delta^2_{n,K}(A)} \xrightarrow{p} 1$$

and $\lim_{n\to\infty}\mathbb{P}\left[\widehat{CV}_{n,K}(A') > \widehat{CV}_{n,K}(A)\right] = 1$. Here we have split the cross validation can be written as

$$\widehat{\text{CV}}_{n,K}(A) = \widehat{\text{CV}}^*_{n,K} + 2Z_{n,K}(A) + \Delta^2_{n,K}(A)$$

where

$$\underbrace{\widehat{\mathrm{CV}}^*_{n,K}}_{I} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \mu(X_i))^2$$

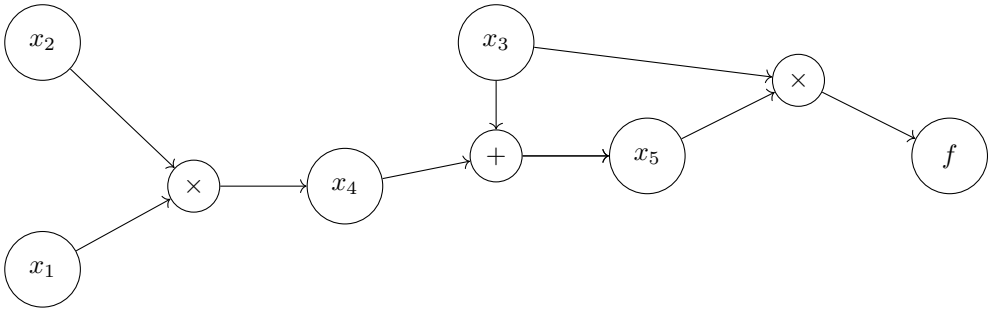
$$\underbrace{\Delta^2_{n,K}(A)}_{II} = \frac{1}{n}\sum_{k=1}^{K}\sum_{i\in S_k}(\mu(X_i) - \hat{\mu}^{(-k)}(X_i))^2$$

$$\underbrace{Z_{n,K}(A)}_{III} = \frac{1}{n}\sum_{k=1}^{K}\sum_{i\in S_k}(Y_i - \mu(X_i))$$
$$\cdot\,(\mu(X_i) - \hat{\mu}^{(-k)}(X_i)).$$

Autograd:

Computational Tree:

Prob easiest to see with example. Let $f(x_1, x_2, x_3) = (x_1 \cdot x_2 + x_3) \cdot x_3$. Then this is represented by the computation graph



In the picture above, think of x_1 and x_2 as parents of x_4. And so x_4 is a child of x_2 and x_1. Note that x_4 and x_5 are intermediate variables that have gradients used for computation but only x_1, x_2, x_3 are updated in backprop.

Chain Rule for Backprop:

The big important formula for back-propogation is just chain rule

$$\frac{d}{dx}L = \sum_{c\in\mathrm{children}(x)} \underbrace{\left(\frac{\partial}{\partial x}c\right)}_{\triangle}\underbrace{\left(\frac{d}{dc}L\right)}_{\star}$$

where $\triangle$ is the local derivative (this is A's responsibility to store for each of its children) and $\star$ is the accumulated derivative (presumably we have already calculated this for each c before getting to A and it is stored in c's grad field). For example, using the example in the computational tree above,

$$\frac{d}{dx_1}f = \underbrace{\frac{\partial x_4}{\partial x_1}}_{\triangle}\cdot\underbrace{\frac{\partial f}{\partial x_4}}_{\star}.$$

$\triangle$ we calculate in the moment using what we know about x_4 (in particular that $x_4 = x_1 \cdot x_2$). And $\star$ we presumably have already calculated and is stored as a field in x_4.