Decisions and randomness

Christos Dimitrakakis

March 27, 2025

Outline

Preferences

Types of rewards

- For e.g. a student: Tickets to concerts.
- For e.g. an investor: A basket of stocks, bonds and currency.
- For everybody: Money.

Preferences among rewards

For any rewards $x, y \in R$, we either

- (a) Prefer x at least as much as y and write $x \preceq^* y$.
- (b) Prefer x not more than y and write $x \succeq^* y$.
- (c) Prefer x about the same as y and write $x \eqsim^* y$.
- (d) Similarly define $\succ^*$ and $\prec^*$

Utility and Cost

Utility function

To make it easy, assign a utility $U(x)$ to every reward through a utility function $U : R \to \mathbb{R}$.

Utility-derived preferences

We prefer items with higher utility, i.e.

- (a) $U(x) \geq U(y) \Leftrightarrow x \succeq^* y$
- (b) $U(x) \leq U(y) \Leftrightarrow y \succeq^* x$

Cost

It is sometimes more convenient to define a cost function $C : R \to \mathbb{R}$ so that we prefer items with lower cost, i.e.

- $C(x) \geq C(y) \Leftrightarrow y \succeq^* x$

Random outcomes

Choosing among rewards

-[A] Bet 10 CHF on black -[B] Bet 10 CHF on 0 -[C] Bet nothing What is the reward here?

Choosing among trips

-[A] Taking the car to Zurich (50' without delays, 80' with delays) -[B] Taking the train to Zurich (60' without delays) What is the reward here?

Random rewards

- Each gamble gives us different rewards with different probabilities.
- These rewards are then random
- For simplicity, we assign a real-valued utility to outcomes. This is a random variable

Expected utility

Actions, outcomes and utility

In this setting, we obtain random outcomes that depend on our actions.

- Actions $a \in A$
- Outcomes $\omega \in \Omega$.
- Probability of outcomes $P(\omega \mid a)$
- Utility $U : \Omega \to \mathbb{R}$

Expected utility

The expected utility of an action is:

$$\mathbb{E}_P[U \mid a] = \sum_{\omega \in \Omega} U(\omega) P(\omega \mid a).$$

The expected utility hypothesis

We prefer a to a' if and only if

$$\mathbb{E}_P[U \mid a] \geq \mathbb{E}_P[U \mid a']$$

The St-Petersburg Paradox

The game

If you give me x CHF, then I promise to (a) Throw a fair coin until it comes heads. (b) If it does so after T throws, then I will give you 2^T CHF.

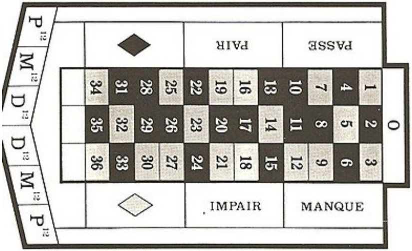
The question

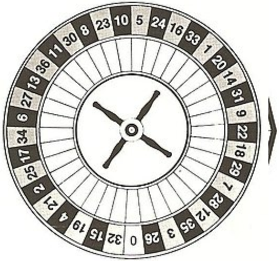
- How much x are you willing to pay to play?
- Given that the expected amount of money is infinite, why are you only willing to pay a small x?

Example: Betting

In this example, probabilities reflect actual randomness

Choice	Win Probability p	Payout w	Expected gain
Don't play	0	0	0
Black	18/37	2	
Red	18/37	2	
0	1/37	36	
1	1/37	36	





What are the expected gains for these bets?

Example: Route selection

- In this example, probabilities reflect subjective beliefs

Choice	Best time	Chance of delay	Delay amount	Expected time
Train	80	5%	5	
Car, route A	60	50%	30	
Car, route B	70	10%	10	

Example: Estimation

- In this example, probabilities are calculated starting from subjective beliefs

Mean-Square Estimation

If we want to guess $\hat{\theta}$, and we knew that $\theta \sim P$, then the guess

$$\hat{\theta} = \mathbb{E}_P(\theta) = \arg\min_{\hat{\theta}} \mathbb{E}_P[(\theta - \hat{\theta})^2]$$

minimises the squared error. This is because

$$\begin{aligned}
\frac{d}{d\hat{\theta}} \mathbb{E}_P[(\theta - \hat{\theta})^2] &= \frac{d}{d\hat{\theta}} \sum_{\omega} [\theta(\omega) - \hat{\theta}]^2 P(\omega) && (1) \\
&= \sum_{\omega} \frac{d}{d\hat{\theta}} [\theta(\omega) - \hat{\theta}]^2 P(\omega) && (2) \\
&= \sum_{\omega} 2[\theta(\omega) - \hat{\theta}](-1) P(\omega) \quad = 2(\hat{\theta} - \mathbb{E}_P[\theta]). && (3)
\end{aligned}$$

Setting this to 0 gives $\hat{\theta} - \mathbb{E}_P[\theta]$

Example: Noisy optimisation

We wish to find the maximum of a function

$$f(x) \triangleq \mathbb{E}[g|x], \qquad \mathbb{E}[g|x] = \int_{-\infty}^{\infty} g(\omega, x)p(\omega)d\omega \tag{4}$$

For this problem we need to use some more complex optimisation method, such as gradient methods

The gradient descent method: one dimension

- Function to minimise $f : \mathbb{R} \to \mathbb{R}$.
- Derivative $\frac{d}{d\theta} f(\beta)$

Gradient descent algorithm

- Input: initial value θ^0, learning rate schedule α_t
- For $t = 1, \dots, T$
  - $\theta^{t+1} = \theta^t - \alpha_t \frac{d}{d\theta} f(\theta^t)$
- Return θ^T

Properties

- If $\sum_t \alpha_t = \infty$ and $\sum_t \alpha_t^2 < \infty$, it finds a local minimum θ^T, i.e. there is $\epsilon > 0$ so that

$$f(\theta^T) < f(\theta), \forall \theta : \|\theta^T - \theta\| < \epsilon.$$

Gradient methods for expected value

Estimate the expected value

$x_t \sim P$ with $\mathbb{E}_P[x_t] = \mu$.

Gradient methods for expected value

Estimate the expected value

$x_t \sim P$ with $\mathbb{E}_P[x_t] = \mu$.

Objective: mean squared error

Here $\ell(x, \theta) = (x - \theta)^2$.

$$\min_\theta \mathbb{E}_P[(x_t - \theta)^2].$$

Gradient methods for expected value

Estimate the expected value

$x_t \sim P$ with $\mathbb{E}_P[x_t] = \mu$.

Objective: mean squared error

Here $\ell(x, \theta) = (x - \theta)^2$.

$$\min_\theta \mathbb{E}_P[(x_t - \theta)^2].$$

Exact gradient update

If we know P, then we can calculate

$$\theta^{t+1} = \theta^t - \alpha_t \frac{d}{d\theta} \mathbb{E}_P[(x - \theta^t)^2] \tag{5}$$

$$\frac{d}{d\theta} \mathbb{E}_P[(x - \theta^t)^2] = 2\,\mathbb{E}_P[x] - \theta^t \tag{6}$$

Gradient for mean estimation

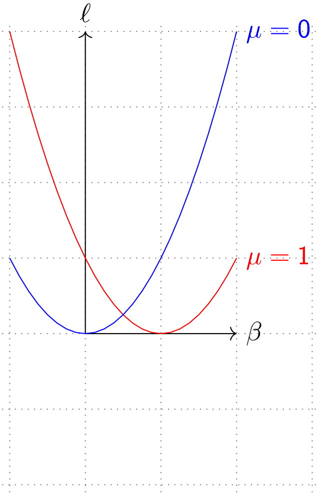
- Let us show this in detail

$$\begin{aligned}\frac{d}{d\theta}\,\mathbb{E}_P[(x-\theta)^2] &= \int_{-\infty}^{\infty} dP(x)\frac{d}{d\theta}(x-\theta)^2\\ &= \int_{-\infty}^{\infty} dP(x)2(x-\theta)\\ &= 2\,\mathbb{E}_P[x]-2\theta.\end{aligned}$$

- If we set the derivative to zero, then we find the optimal solution:
$$\theta^* = \mathbb{E}_P[x]$$

- How can we do this if we only have data $x_t \sim P$?

Mean-squared error cost function



Here we see a plot of $\ell(\mu, \beta) = (\beta - \mu)^2$.

Stochastic gradient for mean estimation

Theorem (Sampling)

For any bounded random variable f,

$$\mathbb{E}_P[f] = \int_X dP(x) f(x) = \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} f(x_t) = \mathbb{E}_P\left[\frac{1}{T}\sum_{t=1}^{T} f(x_t)\right], \qquad x_t \sim P$$

Example (Sampling)

- If we sample x we approximate the gradient:

$$\frac{d}{d\theta}\mathbb{E}_P[(x-\theta)^2] = \int_{-\infty}^{\infty} dP(x) \frac{d}{d\theta}(x-\theta)^2 \approx \frac{1}{T}\sum_{t=1}^{T}\frac{d}{d\theta}(x_t-\theta)^2 = \frac{1}{T}\sum_{t=1}^{T} 2(x_t -$$

Stochastic gradient for mean estimation

Theorem (Sampling)

For any bounded random variable f,

$$\mathbb{E}_P[f] = \int_X dP(x) f(x) = \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} f(x_t) = \mathbb{E}_P\left[\frac{1}{T}\sum_{t=1}^{T} f(x_t)\right], \qquad x_t \sim P$$

Example (Sampling)

- If we sample x we approximate the gradient:

$$\frac{d}{d\theta}\mathbb{E}_P[(x-\theta)^2] = \int_{-\infty}^{\infty} dP(x) \frac{d}{d\theta}(x-\theta)^2 \approx \frac{1}{T}\sum_{t=1}^{T}\frac{d}{d\theta}(x_t-\theta)^2 = \frac{1}{T}\sum_{t=1}^{T} 2(x_t -$$

- If we update θ after each new sample x_t, we obtain:

$$\theta^{t+1} = \theta^t + 2\alpha_t(x_t - \theta^t)$$

The gradient method

- Function to minimise $f : \mathbb{R}^n \to \mathbb{R}$.
- Derivative $\nabla_\theta f(\theta) = \left(\frac{\partial f(\theta)}{\partial \theta_1}, \ldots, \frac{\partial f(\theta)}{\partial \theta_n} \right)$, where $\frac{\partial f}{\partial \beta_n}$ denotes the partial derivative, i.e. varying one argument and keeping the others fixed.

Gradient descent algorithm

- Input: initial value θ^0, learning rate schedule α_t
- For $t = 1, \ldots, T$
  - $\theta^{t+1} = \theta^t - \alpha_t \nabla_\theta f(\theta^t)$
- Return θ^T

Properties

- If $\sum_t \alpha_t = \infty$ and $\sum_t \alpha_t^2 < \infty$, it finds a local minimum θ^T, i.e. there is $\epsilon > 0$ so that

$$f(\theta^T) < f(\theta), \forall \theta : \|\theta^T - \theta\| < \epsilon.$$

Stochastic gradient method

This is the same as the gradient method, but with added noise:

- $\theta^{t+1} = \theta^t - \alpha_t[\nabla_\theta f(\theta^t) + \omega_t]$
- $\mathbb{E}[\omega_t] = 0$ is sufficient for convergence.

Stochastic gradient method

This is the same as the gradient method, but with added noise:

- $\theta^{t+1} = \theta^t - \alpha_t[\nabla_\theta f(\theta^t) + \omega_t]$
- $\mathbb{E}[\omega_t] = 0$ is sufficient for convergence.

Example (When the cost is an expectation)

In machine learning, the cost is frequently an expectation of some function ℓ,

$$f(\theta) = \int_X dP(x)\ell(x,\theta)$$

This can be approximated with a sample

$$f(\theta) \approx \frac{1}{T}\sum_t \ell(x_t,\theta)$$

The same holds for the gradient:

$$\nabla_\theta f(\theta) = \int_X dP(x)\nabla_\theta \ell(x,\theta) \approx \frac{1}{T}\sum_t \nabla_\theta \ell(x_t,\theta)$$

Perceptron algorithm as gradient descent

Target error function

$$\mathbb{E}_{P}^{\theta}[\ell] = \int_{\mathcal{X}} dP(x) \sum_{y} P(y|x)\ell(x, y, \theta)$$

Minimises the error on the true distribution.

Perceptron algorithm as gradient descent

Target error function

$$\mathbb{E}_{P}^{\theta}[\ell] = \int_{\mathcal{X}} dP(x) \sum_{y} P(y|x)\ell(x, y, \theta)$$

Minimises the error on the true distribution.

Empirical error function

$$\mathbb{E}_{D}^{\theta}[\ell] = \frac{1}{T}\sum_{t=1}^{T} \ell(x_t, y_t, \theta), \qquad D = (x_t, y_t)_{t=1}^{T}, \quad x_t, y_t \sim P.$$

Minimises the error on the empirical distribution.

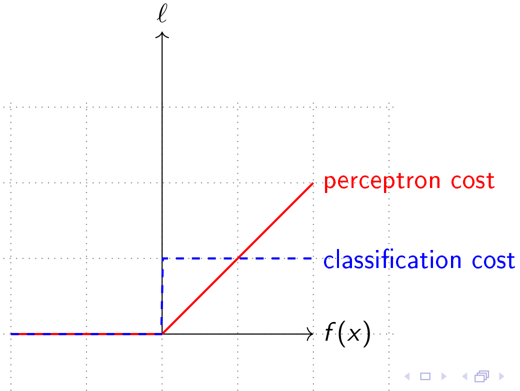
Cost functions and the chain rule

Perceptron cost function

The cost of each example

$$\ell(x, y, \theta) = \overbrace{\mathbb{I}\left\{y(x^\top \theta) < 0\right\}}^{\text{misclassified?}} \overbrace{\left[-y(x^\top \theta)\right]}^{\text{margin of error}} \tag{7}$$

where the indicator function $\mathbb{I}\{A\}$ is 1 when A is true and 0 otherwise.



Derivative of the perceptron cost function

The total cost over the data is defined as

$$L(D,\theta) = \sum_{(x,y)\in D} \ell(x,y,\theta)$$

Taking the derivative, we have

$$\nabla_\theta L(D,\theta) = \nabla_\theta \sum_{(x,y)\in D} \ell(x,y,\beta) = \sum_{(x,y)\in D} \nabla_\theta \ell(x,y,\theta)$$

Reminder: The chain rule

Let $z = g(y)$, $y = f(x)$ so that $z = g(f(x))$. Then $\frac{dz}{dx} = \frac{dz}{dy}\frac{dy}{dx}$

Derivative of the perceptron cost function

The total cost over the data is defined as

$$L(D, \theta) = \sum_{(x,y)\in D} \ell(x, y, \theta)$$

Taking the derivative, we have

$$\nabla_\theta L(D, \theta) = \nabla_\theta \sum_{(x,y)\in D} \ell(x, y, \beta) = \sum_{(x,y)\in D} \nabla_\theta \ell(x, y, \theta)$$

Reminder: The chain rule

Let $z = g(y)$, $y = f(x)$ so that $z = g(f(x))$. Then $\frac{dz}{dx} = \frac{dz}{dy}\frac{dy}{dx}$

Applying the chain rule to calculate the gradient

- $\nabla_\theta \ell(x, y, \theta) = -\mathbb{I}\left\{y(x^\top\theta) < 0\right\} \nabla_\theta[y(x^\top\theta)]$.

Derivative of the perceptron cost function

The total cost over the data is defined as

$$L(D,\theta) = \sum_{(x,y)\in D} \ell(x,y,\theta)$$

Taking the derivative, we have

$$\nabla_\theta L(D,\theta) = \nabla_\theta \sum_{(x,y)\in D} \ell(x,y,\beta) = \sum_{(x,y)\in D} \nabla_\theta \ell(x,y,\theta)$$

Reminder: The chain rule

Let $z = g(y)$, $y = f(x)$ so that $z = g(f(x))$. Then $\frac{dz}{dx} = \frac{dz}{dy}\frac{dy}{dx}$

Applying the chain rule to calculate the gradient

- $\nabla_\theta \ell(x,y,\theta) = -\mathbb{I}\left\{y(x^\top\theta) < 0\right\} \nabla_\theta[y(x^\top\theta)]$.
- $\frac{\partial\theta}{\partial\theta_i}[y(x_t^\top\theta)] = yx_{t,i}$ (gradient of Perceptron's output)

Derivative of the perceptron cost function

The total cost over the data is defined as

$$L(D, \theta) = \sum_{(x,y)\in D} \ell(x, y, \theta)$$

Taking the derivative, we have

$$\nabla_\theta L(D, \theta) = \nabla_\theta \sum_{(x,y)\in D} \ell(x, y, \beta) = \sum_{(x,y)\in D} \nabla_\theta \ell(x, y, \theta)$$

Reminder: The chain rule

Let $z = g(y)$, $y = f(x)$ so that $z = g(f(x))$. Then $\frac{dz}{dx} = \frac{dz}{dy}\frac{dy}{dx}$

Applying the chain rule to calculate the gradient

- $\nabla_\theta \ell(x, y, \theta) = -\mathbb{I}\left\{y(x^\top\theta) < 0\right\} \nabla_\theta[y(x^\top\theta)]$.
- $\frac{\partial\theta}{\partial\theta_i}[y(x_t^\top\theta)] = yx_{t,i}$ (gradient of Perceptron's output)
- Gradient update: $\theta^{t+1} = \theta^t - \nabla_\theta \ell(x, y, \theta) = \theta^t + yx_t$

Derivative of the perceptron cost function

The total cost over the data is defined as

$$L(D,\theta) = \sum_{(x,y)\in D} \ell(x,y,\theta)$$

Taking the derivative, we have

$$\nabla_\theta L(D,\theta) = \nabla_\theta \sum_{(x,y)\in D} \ell(x,y,\beta) = \sum_{(x,y)\in D} \nabla_\theta \ell(x,y,\theta)$$

Reminder: The chain rule

Let $z = g(y)$, $y = f(x)$ so that $z = g(f(x))$. Then $\frac{dz}{dx} = \frac{dz}{dy}\frac{dy}{dx}$

Applying the chain rule to calculate the gradient

- $\nabla_\theta \ell(x,y,\theta) = -\mathbb{I}\left\{y(x^\top\theta) < 0\right\} \nabla_\theta[y(x^\top\theta)]$.
- $\frac{\partial\theta}{\partial\theta_i}[y(x_t^\top\theta)] = yx_{t,i}$ (gradient of Perceptron's output)
- Gradient update: $\theta^{t+1} = \theta^t - \nabla_\theta \ell(x,y,\theta) = \theta^t + yx_t$

Derivative of the perceptron cost function

The total cost over the data is defined as

$$L(D, \theta) = \sum_{(x,y) \in D} \ell(x, y, \theta)$$

Taking the derivative, we have

$$\nabla_\theta L(D, \theta) = \nabla_\theta \sum_{(x,y) \in D} \ell(x, y, \beta) = \sum_{(x,y) \in D} \nabla_\theta \ell(x, y, \theta)$$

Reminder: The chain rule

Let $z = g(y)$, $y = f(x)$ so that $z = g(f(x))$. Then $\frac{dz}{dx} = \frac{dz}{dy}\frac{dy}{dx}$

Applying the chain rule to calculate the gradient

- $\nabla_\theta \ell(x, y, \theta) = -\mathbb{I}\left\{y(x^\top \theta) < 0\right\} \nabla_\theta [y(x^\top \theta)]$.
- $\frac{\partial \theta}{\partial \theta_i}[y(x_t^\top \theta)] = y x_{t,i}$ (gradient of Perceptron's output)
- Gradient update: $\theta^{t+1} = \theta^t - \nabla_\theta \ell(x, y, \theta) = \theta^t + y x_t$

The classification error cost function is not differentiable :(

Margins and confidences

We can think of the output of the network as a measure of confidence

./fig/margin.pdf

Margins and confidences

We can think of the output of the network as a measure of confidence

./fig/margin.pdf

By applying the logit function, we can bound a real number x to $[0, 1]$:

$$f(x) = \frac{e^x}{1 + e^x} = \frac{1}{1 + e^{-x}}$$

Logistic regression

Output as a measure of confidence, given the parameter θ

$$P_\theta(y = 1|x) = \frac{1}{1 + \exp(-x_t^\top \theta)}$$

The original output $x_t^\top \theta$ is now passed through the logit function.

Logistic regression

Output as a measure of confidence, given the parameter θ

$$P_\theta(y=1|x) = \frac{1}{1+\exp(-x_t^\top \theta)}$$

The original output $x_t^\top \theta$ is now passed through the logit function.

Negative Log likelihood

$\ell(x_t, y_t, \theta) = -\ln P_\theta(y_t|x_t) = \ln(1+\exp(-y_t x_t^\top \theta))$

$$\begin{aligned}
\nabla_\theta \ell(x_t, y_t, \theta) &= \frac{1}{1+\exp(-y x_t^\top \theta)} \nabla_\theta [1+\exp(-y x_t^\top \theta)] \\
&= \frac{1}{1+\exp(-y x_t^\top \theta)} \exp(-y x_t^\top \theta)[\nabla_\theta(-y_t x_t^\top \theta)] \\
&= -\frac{1}{1+\exp(x_t^\top \theta)} (x_{t,i})_{i=1}^n e
\end{aligned}$$

- $\mathbb{E}_P(\ell) = \int_X dP(x) \sum_{y \in Y} P(y|x) P_\theta(y_t + x_t)$