Posterior vulnerability and leakage

Mário S. Alvim
(msalvim@dcc.ufmg.br)

Quantitative Information Flow

DCC-UFMG
(2021/1)

Introduction

- Recall that:
  - Given a secret X with prior distribution π, the "threat" to X can be measured as its g-vulnerability $V_g(\pi)$ for a suitably chosen gain function g.
  - The choice of g reflects the importance of the secret, and might vary depending on the interests of the defender and adversary.
- Consider now there's a channel matrix C: $\mathcal{X} \rightarrowtriangle \mathcal{Y}$, which might leak some information about X.

  Here we'll consider how to quantify the g-vulnerability of X after C is run.

Posterior g-vulnerability and its basic properties

Posterior g-vulnerability and its basic properties

- There are two perspectives we can take on "after C is run".
- The **dynamic perspective** considers the effect of the adversary's observing a particular channel output y.
  - The adversary updates her knowledge about X, changing it from the prior π to the posterior distribution $p_{X|y}$.
  - This run of C changes the Bayes vulnerability from $V_1(\pi)$ to $V_1(p_{X|y})$.
  - Then we might measure the amount of leakage as the difference

$$V_1(p_{X|y}) - V_1(\pi)\,.$$

- While the dynamic view of leakage is natural, it does suffer from some significant drawbacks...

Posterior g-vulnerability and its basic properties

- First drawback of the dynamic view: Bayes vulnerability can decrease.

- Example 1 Consider channel matrix C

C	y_1	y_2
x_1	1	0
x_2	0	1
x_3	0	1
x_4	0	1
x_5	0	1

under prior $\pi = (9/10, 1/40, 1/40, 1/40, 1/40)$,

whose Bayes vulnerability is $V_1(\pi) = 9/10$.

If output y_2 is observed, the posterior distribution is

$$p_{X|y_2} = (0, 1/4, 1/4, 1/4, 1/4) \ ,$$

and Bayes vulnerability has decreased from $9/10$ to $1/4$, leading to a negative leakage of $1/4 - 9/10 = -13/20$.

Perhaps counter-intuitively, running the channel seems to have decreased the threat posed by the adversary. ◁

Posterior g-vulnerability and its basic properties

- Example 2 A real-world scenario corresponding to the previous example is the case of a doctor's trying to diagnose an unknown disease.

  Based on the symptoms, there might be only one likely diagnosis, making the prior "vulnerability" large.

  But if a medical test refutes that diagnosis, then the doctor is left with no idea of the disease, making the posterior "vulnerability" small.

  (Here again we have a benevolent adversary.) ◁

Posterior g-vulnerability and its basic properties

- Second drawback of the dynamic view: policy enforcement.

- Example 3 Imagine using an execution monitor to track the Bayes vulnerability of the secret, verifying that it stays below some threshold.

  If a run produces an output that leaks too much, what could the monitor do?

  It might try to respond by aborting the execution, but the very act of aborting might in itself leak a lot of information to the adversary.

  For example, in the case of a password checker, aborting the execution when the adversary enters the correct password would reveal the entire password to the adversary (and also makes the password checker useless). ◁

Posterior g-vulnerability and its basic properties

- For the reasons above, we consider a different perspective.
- The **static perspective** considers all the possible outputs of a channel C, independent of any particular execution.

  Here a channel C maps
  - the prior π
  - to a hyper-distribution $[\pi \triangleright C]$, which represents:
    - all the possible "worlds" of adversary knowledge about X (the inners),
    - together with their probabilities (the outer).
- It's natural to calculate the g-vulnerability of each of the inners, to see how vulnerable the secret is in that "world".

  But how should that set of vulnerabilities be combined?

Posterior g-vulnerability and its basic properties

- One possibility is to consider the **maximum** of those vulnerabilities, as this represents the maximum threat to the secret.

  That approach is quite pessimistic, however, because it pays no attention to the outer probability of the worst inner, which may indeed be highly unlikely.

- Example 4 Consider a password checker when the user has tried to log in with some guess g.

  There are two possible outputs: *accept* and *reject*.

  In the case of output *accept*, the posterior distribution is a point distribution on g, which means that the max-case posterior Bayes vulnerability is

  $$V_1(p_{X|accept}) = 1.$$

  Hence, using maximum posterior vulnerability, a password checker is judged to be as bad as a channel that leaks the password completely! ◁

Posterior g-vulnerability and its basic properties

- We find it more useful to define posterior g-vulnerability as the **expected value** of the g-vulnerability over the hyper-distribution.

Definition (5.2)

Given prior π, gain function $g\colon \mathbb{G}\mathcal{X}$ and channel C, the **posterior g-vulnerability** $V_g[\pi \triangleright C]$ is defined as the expected value of V_g over $[\pi \triangleright C]$, that is

$$V_g[\pi \triangleright C] \;:=\; \sum_i a_i V_g(\delta^i)\;, \qquad \text{where} \qquad [\pi \triangleright C] \;=\; \sum_i a_i [\delta^i] \quad .$$

(We are thus overloading V_g, allowing it to take as argument either a distribution in $\mathbb{D}\mathcal{X}$ or a hyper-distribution in $\mathbb{D}^2\mathcal{X}$, in the latter case omitting function-application parentheses if the argument has brackets already.)

- Note that posterior g-vulnerability:
  - Can be written as $V_g[\pi \triangleright C] = \mathcal{E}_{[\pi \triangleright C]}\, V_g$.
  - Is well defined on abstract channels.

Posterior g-vulnerability and its basic properties

- **Example 5** Consider Bayes vulnerability, which is g_{id}-vulnerability.

  In a previous example we have seen that:

The channel

C	y_1	y_2	y_3	y_4
x_1	$1/2$	$1/2$	0	0
x_2	0	$1/4$	$1/2$	$1/4$
x_3	$1/2$	$1/3$	$1/6$	0

maps the prior distribution

$$\pi = (1/4, 1/2, 1/4)$$

to the hyper-distribution

$[\pi \triangleright C]$	$1/4$	$1/3$	$7/24$	$1/8$
x_1	$1/2$	$3/8$	0	0
x_2	0	$3/8$	$6/7$	1
x_3	$1/2$	$1/4$	$1/7$	0

.

- The prior Bayes vulnerability is

$$V_g(\pi) = 1/2 \ ,$$

  as the adversary's best action *a priori* is to guess x_2.

- The posterior Bayes vulnerability is

$$\begin{aligned} V_1[\pi \triangleright C] &= 1/4 \cdot V_1(1/2, 0, 1/2) + 1/3 \cdot V_1(3/8, 3/8, 1/4) + \\ &\quad 7/24 \cdot V_1(0, 6/7, 1/7) + 1/8 \cdot V_1(0, 1, 0) \\ &= 1/4 \cdot 1/2 + 1/3 \cdot 3/8 + 7/24 \cdot 6/7 + 1/8 \cdot 1 \quad = \quad 5/8 \quad . \end{aligned}$$

Posterior g-vulnerability and its basic properties

- Example 5 (Continued)

Notice that posterior Bayes vulnerability is larger prior Bayes vulnerability.

That reflects the fact that the channel outputs help the adversary to choose actions that are better for her.

- The first posterior distribution in $[\pi \triangleright C]$ directs her not to guess x_2 but instead x_1 or x_3, which are equally good.
- On the other three posterior distributions, guessing x_2 remains optimal.

Thus her probability of guessing X correctly increases from $1/2$ to $5/8$. $\triangleleft$

Posterior g-vulnerability and its basic properties

- Example 6 Consider again a scenario having a secret set $\mathcal{X} = \{x_1, x_2\}$ with prior $\pi = (0.3, 0.7)$.

  Consider action set $\mathcal{W} = \{w_1, w_2, w_3, w_4, w_5\}$ Suppose that channel C is leading to a gain function having values as in

G	x_1	x_2
w_1	-1.0	1.0
w_2	0.0	0.5
w_3	0.4	0.1
w_4	0.8	-0.9
w_5	0.1	0.2

.

C	y_1	y_2
x_1	0.75	0.25
x_2	0.25	0.75

.

Notice that C reveals rather a lot of information about the secret: if the input is x_i (where i is 1 or 2), then three-fourths of the time the output is y_i.

Now let us compute $V_g[\pi \triangleright \mathsf{C}]$ directly from the concrete channel.

Posterior g-vulnerability and its basic properties

- Example 6 (Continued)

We first find hyper-distribution $[\pi \triangleright C]$.

Scaling the rows of C with π, we get joint matrix J:

J	y_1	y_2
x_1	0.225	0.075
x_2	0.175	0.525

By summing and normalizing the columns of J, we find that $[\pi \triangleright C]$ has inners

- $\delta^1 = (0.5625, 0.4375)$ and
- $\delta^2 = (0.125, 0.875)$,

with outer probabilities 0.4 and 0.6, respectively.

Posterior g-vulnerability and its basic properties

- Example 6 (Continued)

Now we must compute $V_g(\delta^1)$ and $V_g(\delta^2)$.

For each inner, this requires computing the expected gain for each possible w in $\mathcal{W}$, to see which action is best.

For δ^1 we get the following results:

$$
\begin{array}{rcrcr}
\delta^1_{x_1} g(w_1, x_1) + \delta^1_{x_2} g(w_1, x_2) = & 0.5625 \cdot (-1.0) + 0.4375 \cdot 1.0 & = & -0.12500 \\
\delta^1_{x_1} g(w_2, x_1) + \delta^1_{x_2} g(w_2, x_2) = & 0.5625 \cdot 0.0 + 0.4375 \cdot 0.5 & = & 0.21875 \\
\delta^1_{x_1} g(w_3, x_1) + \delta^1_{x_2} g(w_3, x_2) = & 0.5625 \cdot 0.4 + 0.4375 \cdot 0.1 & = & 0.26875 \\
\delta^1_{x_1} g(w_4, x_1) + \delta^1_{x_2} g(w_4, x_2) = & 0.5625 \cdot 0.8 + 0.4375 \cdot (-0.9) & = & 0.05625 \\
\delta^1_{x_1} g(w_5, x_1) + \delta^1_{x_2} g(w_5, x_2) = & 0.5625 \cdot 0.1 + 0.4375 \cdot 0.2 & = & 0.14375
\end{array}
$$

Here we find that w_3 is the best action and $V_g(\delta^1) = 0.26875$.

Posterior g-vulnerability and its basic properties

- Example 6 (Continued)

For δ^2 we get

$$\begin{aligned}
\delta^2_{x_1} g(w_1, x_1) + \delta^2_{x_2} g(w_1, x_2) &= 0.125 \cdot (-1.0) + 0.875 \cdot 1.0 &&= 0.7500 \\
\delta^2_{x_1} g(w_2, x_1) + \delta^2_{x_2} g(w_2, x_2) &= 0.125 \cdot 0.0 + 0.875 \cdot 0.5 &&= 0.4375 \\
\delta^2_{x_1} g(w_3, x_1) + \delta^2_{x_2} g(w_3, x_2) &= 0.125 \cdot 0.4 + 0.875 \cdot 0.1 &&= 0.1375 \\
\delta^2_{x_1} g(w_4, x_1) + \delta^2_{x_2} g(w_4, x_2) &= 0.125 \cdot 0.8 + 0.875 \cdot (-0.9) &&= -0.6875 \\
\delta^2_{x_1} g(w_5, x_1) + \delta^2_{x_2} g(w_5, x_2) &= 0.125 \cdot 0.1 + 0.875 \cdot 0.2 &&= 0.1875
\end{aligned}$$

Here we find instead that w_1 is the best action and $V_g(\delta^2) = 0.75$.

Posterior g-vulnerability and its basic properties

- Example 6 (Continued)

Finally, we compute $V_g[\pi \triangleright C]$ overall by weighting the g-vulnerabilities of the inners with their respective outer probabilities:

$$V_g[\pi \triangleright C] \;=\; 0.4 \cdot V_g(\delta^1) + 0.6 \cdot V_g(\delta^2) \;=\; 0.4 \cdot 0.26875 + 0.6 \cdot 0.75 \;=\; 0.5575\,.$$

Let's consider the operational significance of the fact that $V_g[\pi \triangleright C] = 0.5575$.

In a previous example we found that $V_g(\pi) = 0.4$, which is less than 0.5575.

This reflects that C allows the adversary to use a better strategy for actions:

- on output y_1 (corresponding to δ^1) her best action is w_3, and
- and on output y_2 (corresponding to δ^2) her best action is w_1.

In contrast, if she knew only π, her only reasonable strategy would be to choose action w_1 every time. ◁

Posterior g-vulnerability and its basic properties

- Example 7 To get a fuller understanding of posterior g-vulnerability for the gain function g and channel matrix C considered in the previous example let us now graph $V_g[\pi \triangleright C]$ as a function of a general prior

$$\pi = (x, 1-x),$$

where $0 \leq x \leq 1$.

Here we get the hyper-distribution

$[\pi \triangleright C]$	$\frac{1+2x}{4}$	$\frac{3-2x}{4}$
x_1	$\frac{3x}{1+2x}$	$\frac{x}{3-2x}$
x_2	$\frac{1-x}{1+2x}$	$\frac{3-3x}{3-2x}$

.

Posterior g-vulnerability and its basic properties

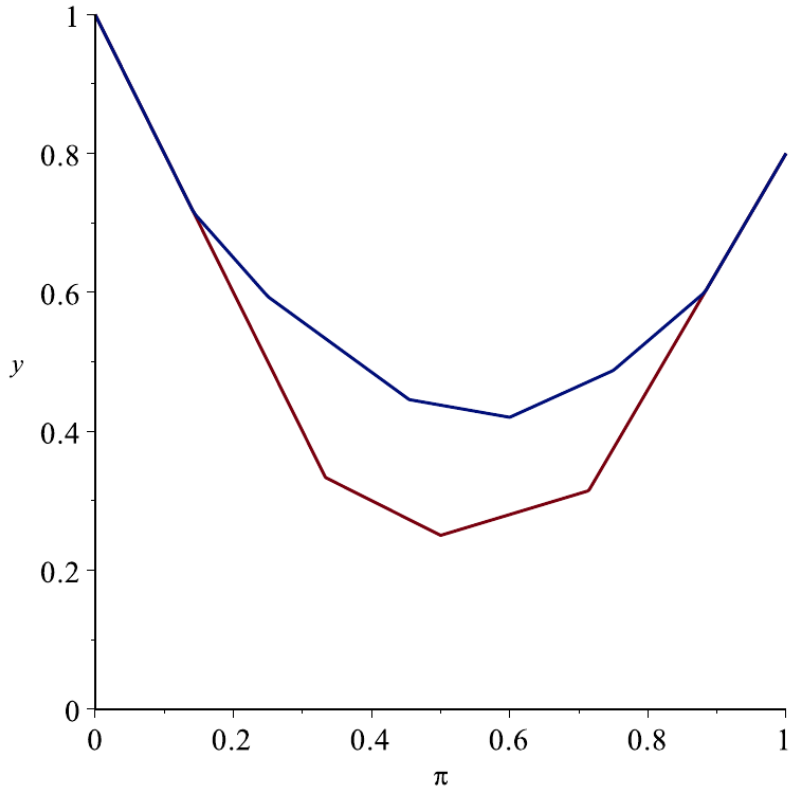
- Example 7 (Continued)

The figure compares the graphs of:

- $V_g(\pi)$ (in red), and
- $V_g[\pi \triangleright C]$ (in blue).

Notice that $V_g[\pi \triangleright C]$ is often greater than $V_g(\pi)$.

This is to be expected intuitively, since channel C increases the adversary's knowledge about X, enabling a better choice of actions.

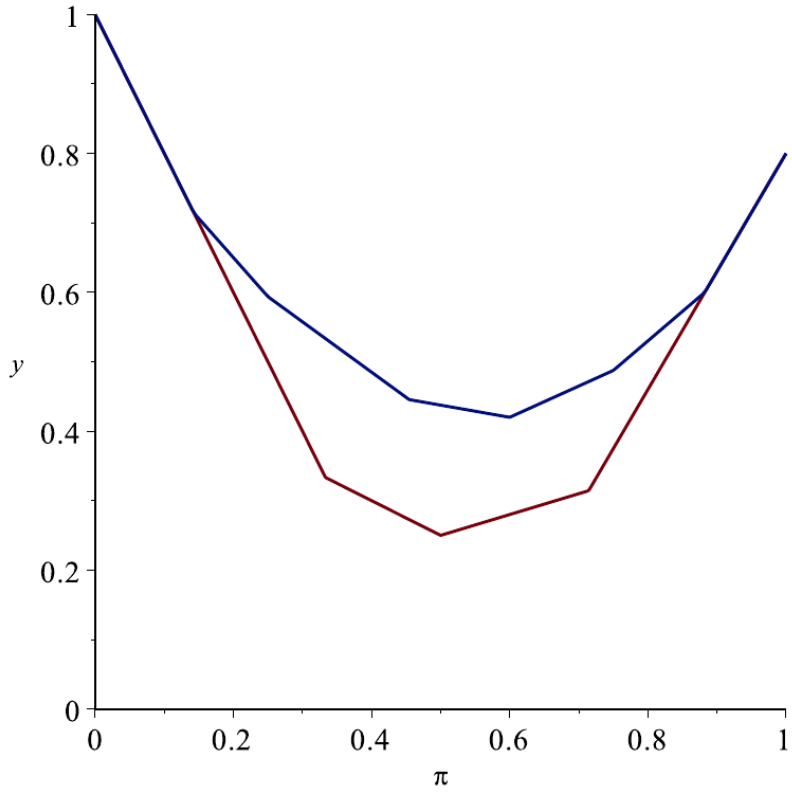


Posterior g-vulnerability and its basic properties

- Example 7 (Continued)

In the prior situation, the adversary's choice of which action to take is guided only by π itself.

In the posterior situation, her choice can be guided by both π and the output of channel C, so she can choose one action if the output is y_1 and another if the output is y_2.



Posterior g-vulnerability and its basic properties

- Example 7 (Continued)

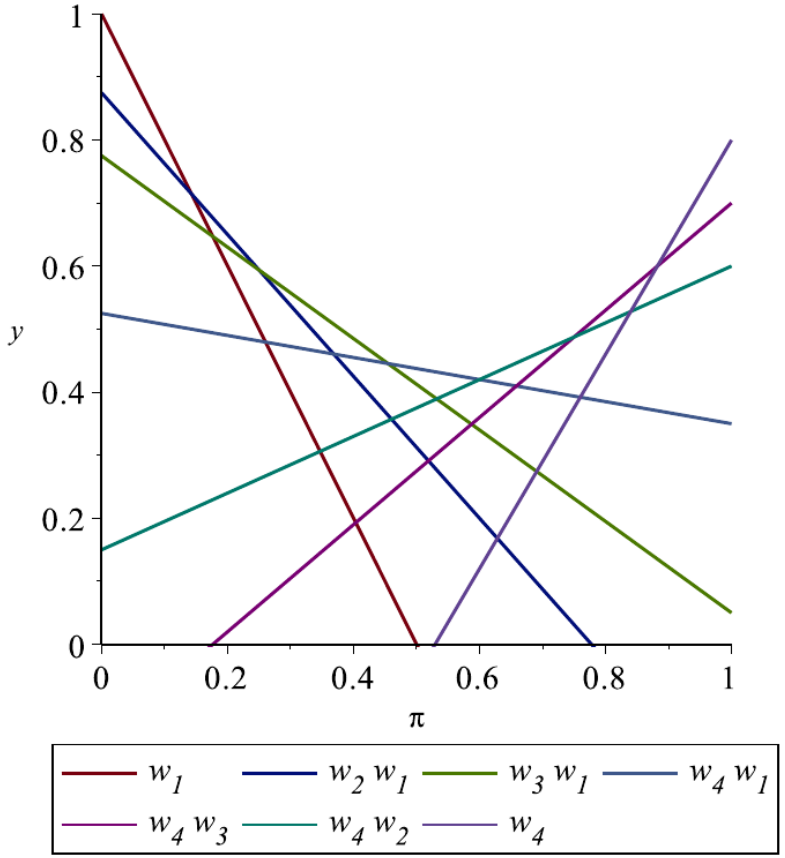
Let $w_i w_j$ denote the strategy of choosing:

- action w_i on output y_1, and
- action w_j on output y_2.

The adversary has $|\mathcal{W}|^{|\mathcal{Y}|} = 5^2 = 25$ possible strategies.

We denote by w_i the "degenerate" strategy $w_i w_i$ in which she chooses w_i regardless of the output.

Let's plot the expected gain for seven of these strategies. (Each of the remaining 18 strategies is dominated by the other seven.)



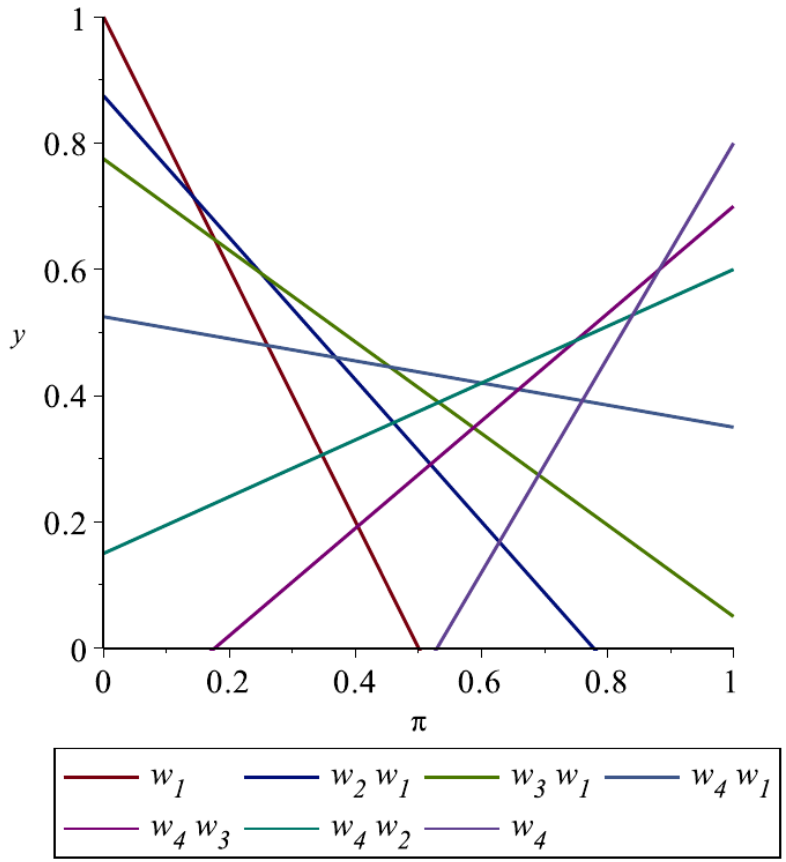
Posterior g-vulnerability and its basic properties

- Example 7 (Continued)

$V_g[\pi \triangleright C]$ lies along the maximum of those seven strategies, each of which turns out to be optimal for certain π.

It can be considered to be the envelope (from above) of all the individual strategy lines and, in general, the envelope of hyperplanes.

Notice that strategies w_1 and w_4 are optimal in the portions of the graph where $V_g(\pi) = V_g[\pi \triangleright C]$, which makes sense because if the optimal strategy for π ignores the output of C, then C is of no help to the adversary on that π.



Posterior g-vulnerability and its basic properties

- Example 7 (Continued)

Finally, let's compare the posterior g-vulnerability of C with that of the channel that leaks everything.

This channel $\mathbb{O}$ has reduced matrix

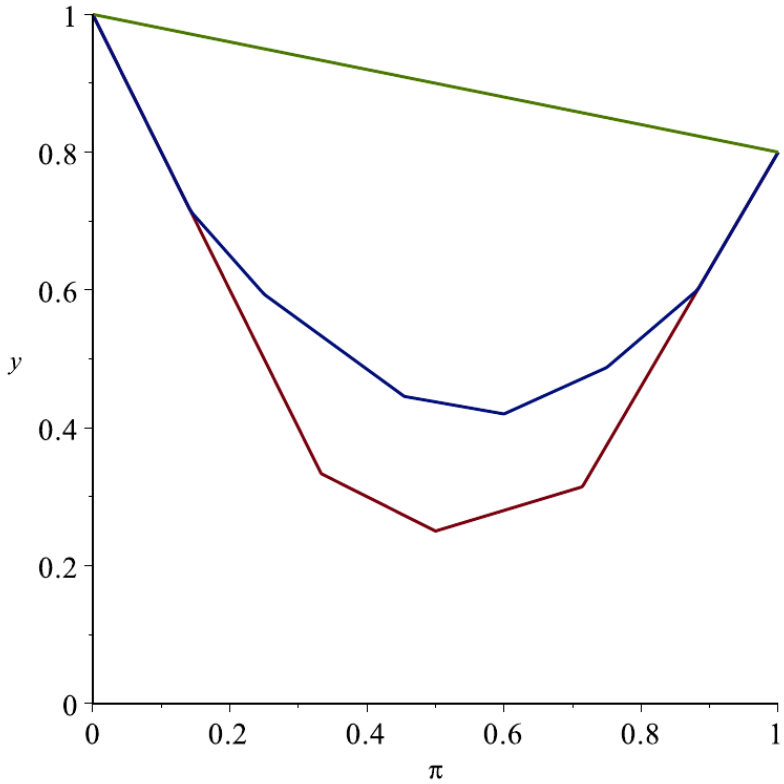
$\mathbb{O}$	x_1	x_2
x_1	1	0
x_2	0	1

,

which copies its input to its output.

The figure compares:

- $V_g(\pi)$ (in red),
- $V_g[\pi \triangleright \mathsf{C}]$ (in blue), and
- $V_g[\pi \triangleright \mathbb{O}]$ (in green).



Posterior g-vulnerability and its basic properties

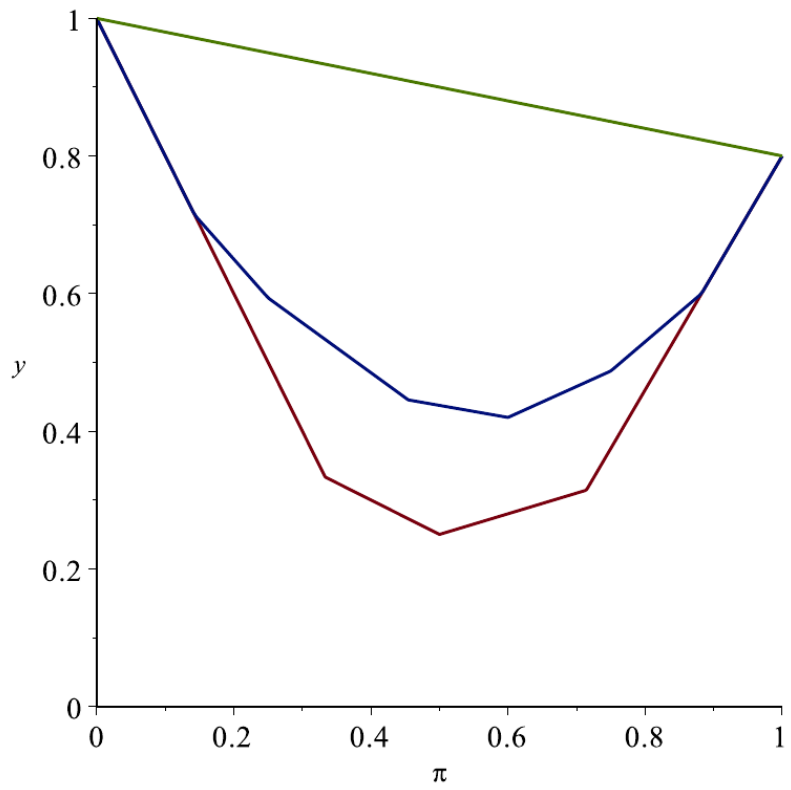
- Example 7 (Continued)

$V_g[\pi \triangleright \mathbb{O}]$ reflects that the output of $\mathbb{O}$ reveals whether the secret is:

  - x_1 (in which case the adversary should take action w_4, giving gain 0.8) or
  - x_2 (in which case she should take action w_1, giving gain 1.0).

Hence on prior $(x, 1-x)$ her expected gain is

$$x \cdot 0.8 + (1-x) \cdot 1.0 \quad = \quad 1 - 0.2 \cdot x \quad .$$



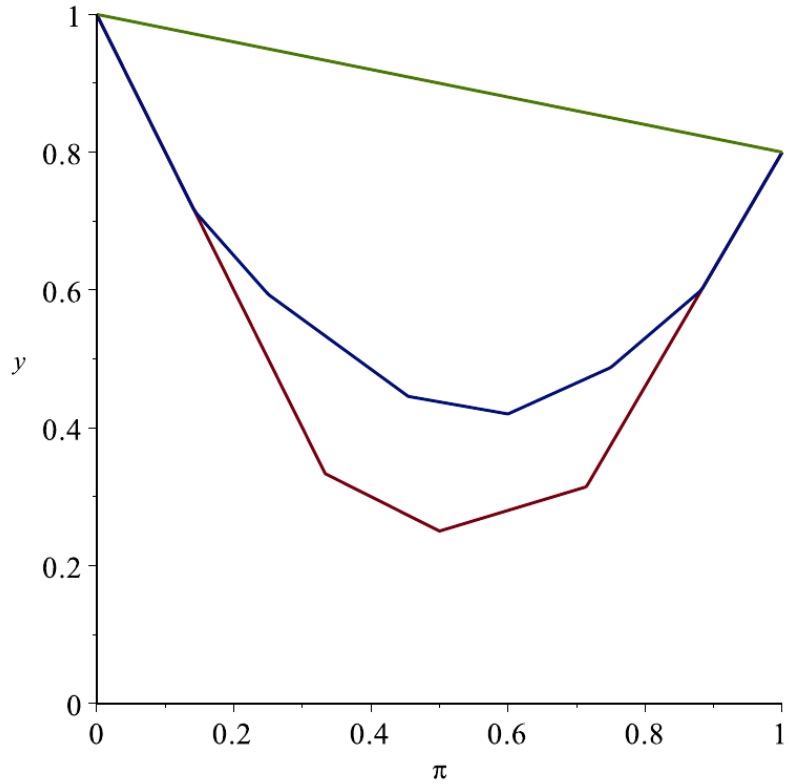
Posterior g-vulnerability and its basic properties

- Example 7 (Continued)

Note that we have for every π that

$$V_g(\pi) \leq V_g[\pi \triangleright \mathsf{C}] \leq V_g[\pi \triangleright \mathbb{O}] \ .$$

Those inequalities turn out to hold in general, as we will prove soon.



Posterior g-vulnerability and its basic properties

- We now establish some important properties of posterior g-vulnerability.
- Recall that $V_g[\pi \triangleright C]$ is defined in terms of the hyper-distribution $[\pi \triangleright C]$.

  But it's useful to establish formulae that allow calculation of posterior g-vulnerability directly from a channel matrix $\mathsf{C}: \mathcal{X} \to \mathcal{Y}$ that realizes C.

  We first show that $V_g[\pi \triangleright \mathsf{C}]$ can be characterized in terms of the posterior distributions $p_{X|y}$ for $y \in \mathcal{Y}$.

Theorem (5.6)

Given prior π, gain function $g: \mathbb{G}\mathcal{X}$, and channel matrix C from $\mathcal{X}$ to $\mathcal{Y}$, we have

$$V_g[\pi \triangleright \mathsf{C}] \quad = \quad \sum_{\substack{y \in \mathcal{Y} \\ p(y) \neq 0}} p(y) V_g(p_{X|y}) \quad .$$

Posterior g-vulnerability and its basic properties

- **Proof.** Recall that the support of hyper-distribution $[\pi \triangleright C]$ consists of the posterior distributions $p_{X|y}$ such that $p(y) \neq 0$.

  But recall that if outputs $y_1, \ldots, y_k$ should give rise to the same posterior distribution,

$$p_{X|y_1} \quad = \quad \cdots \quad = \quad p_{X|y_k} \quad ,$$

  then the hyper-distribution coalesces them into a single inner distribution $p_{X|y_1}$ with outer probability $p(y_1) + \cdots + p(y_k)$.

  Since then we have

$$(p(y_1) + \cdots + p(y_k)) V_g(p_{X|y_1}) \quad = \quad p(y_1) V_g(p_{X|y_1}) + \cdots + p(y_k) V_g(p_{X|y_k}) \quad ,$$

  the desired equality follows. □

Posterior g-vulnerability and its basic properties

- The next theorem shows that $V_g[\pi \triangleright \mathsf{C}]$ can be calculated directly from π and C and g, with no need to calculate the posterior distributions.

Theorem (5.7)

Given prior π, gain function $g\colon \mathbb{G}^{\text{fin}}\mathcal{X}$, and channel matrix C from $\mathcal{X}$ to $\mathcal{Y}$, we have

$$V_g[\pi \triangleright \mathsf{C}] \quad = \quad \sum_{y \in \mathcal{Y}} \max_{w \in \mathcal{W}} \sum_{x \in \mathcal{X}} \pi_x \mathsf{C}_{x,y} g(w,x) \quad .$$

For generic $g\colon \mathbb{G}\mathcal{X}$, the max *should be changed to* sup.

Proof. Completing this proof is part of your homework assignment for this lecture! ◁

Posterior g-vulnerability and its basic properties

- Now we consider properties of the behavior of posterior g-vulnerability.
- First we show that g-vulnerability satisfies **monotonicity**.

Theorem (5.8 - Monotonicity)

Posterior g-vulnerability is always greater than or equal to prior g-vulnerability: for any prior π, channel C and gain function $g\colon \mathbb{G}\mathcal{X}$, we have

$$V_g[\pi \triangleright C] \geq V_g(\pi) \ .$$

Proof. The property follows from two key facts: that V_g is convex (Theorem 3.13) and that the expectation of a hyper-distribution $[\pi \triangleright C]$ is the prior π (Corollary 4.8). So, letting $[\pi \triangleright C] = \sum_i a_i[\delta^i]$, we can reason as follows:

$$V_g[\pi \triangleright C] \quad = \quad \sum_i a_i V_g(\delta^i) \quad \geq \quad V_g(\sum_i a_i \delta^i) \quad = \quad V_g(\pi) \quad .$$

The convexity is what justifies the inequality in the middle. □

Posterior g-vulnerability and its basic properties

- We noted just above that $V_g(\pi)$ is a convex function of π.
- We now establish the behavior of posterior vulnerability.

Theorem (5.9 - Convexity of posterior V_g)

For any $g\colon \mathbb{G}\mathcal{X}$, posterior g-vulnerability is a convex function of priors:

$$V_g[\sum_i a_i \delta^i \triangleright C] \;\leq\; \sum_i a_i\, V_g[\delta^i \triangleright C]\,.$$

Posterior g-vulnerability is as well a convex function of concrete channels:

$$V_g[\pi \triangleright \sum_i a_i C^i] \;\leq\; \sum_i a_i\, V_g[\pi \triangleright C^i]\,.$$

But on abstract channels, posterior g-vulnerability is actually linear, not merely convex:

$$V_g[\pi \triangleright \sum_i a_i C^i] \;=\; \sum_i a_i\, V_g[\pi \triangleright C^i]\,.$$

Posterior g-vulnerability and its basic properties

- **Proof.** For convexity on priors, if C is any channel matrix realizing C then

$$
\begin{array}{lll}
 & V_g[\sum_i a_i \delta^i \triangleright C] & \\
= & V_g[\sum_i a_i \delta^i \triangleright \mathsf{C}] & \text{“change from } C \text{ to } \mathsf{C}\text{”} \\
= & \sum_y \max_w \sum_x (\sum_i a_i \delta^i)_x \mathsf{C}_{x,y} g(w,x) & \text{“Theorem 5.7 ”} \\
= & \sum_y \max_w \sum_x \sum_i a_i \delta^i_x \mathsf{C}_{x,y} g(w,x) & \text{“cvx. comb. of distributions”} \\
= & \sum_y \max_w \sum_i a_i \sum_x \delta^i_x \mathsf{C}_{x,y} g(w,x) & \text{“reorganizing sum”} \\
\leq & \sum_y \sum_i a_i \max_w \sum_x \delta^i_x \mathsf{C}_{x,y} g(w,x) & \text{“max of sum} \leq \text{sum of max”} \\
= & \sum_i a_i \sum_y \max_w \sum_x \delta^i_x \mathsf{C}_{x,y} g(w,x) & \text{“reorganizing sum”} \\
= & \sum_i a_i V_g[\delta^i \triangleright \mathsf{C}] & \text{“Theorem 5.7”} \\
= & \sum_i a_i V_g[\delta^i \triangleright C] \quad . &
\end{array}
$$

For generic $g\colon \mathbb{G}\mathcal{X}$, the max's should be changed to sup's.

Similar arguments prove convexity on concrete channels (for which we appeal to $(\sum_i a_i \mathsf{C}^i)_{x,y} = \sum_i a_i \mathsf{C}^i_{x,y}$) and linearity on abstract channels (appealing instead to $[\pi \triangleright \sum_i a_i C^i] = \sum_i a_i [\pi \triangleright C^i]$). □

Posterior g-vulnerability and its basic properties

- Finally, we show that the gain-function$-$algebra properties that were proved also carry over to posterior g-vulnerability.

Theorem (5.10)

For any $g\colon \mathbb{G}\mathcal{X}$, prior π, channel C, $k \geq 0$ and $\kappa \in \mathbb{R}^{|\mathcal{X}|}$, we have

$$\begin{array}{rcl} V_{g_{\times k}}[\pi \triangleright C] & = & k V_g[\pi \triangleright C]\ , \\ V_{g_{+\kappa}}[\pi \triangleright C] & = & V_g[\pi \triangleright C] + \kappa \cdot \pi \quad . \end{array}$$

Proof. These follow from Theorem 3.14. Letting $[\pi \triangleright C] = \sum_i a_i [\delta^i]$:

$$\begin{array}{rcll} & & V_{g_{\times k}}[\pi \triangleright C] & \\ = & & \sum_i a_i V_{g_{\times k}}(\delta^i) & \\ = & & \sum_i a_i k V_g(\delta^i) & \text{“Theorem 3.14”} \\ = & & k \sum_i a_i V_g(\delta^i) & \\ = & & k V_g[\pi \triangleright C] \quad , & \end{array}$$

and similarly for $V_{g_{+\kappa}}[\pi \triangleright C]$. □

Multiplicative and additive g-leakage

Multiplicative and additive g-leakage

- Now we are ready to define g-leakage.

  We do so by comparing the prior and posterior g-vulnerabilities, either:

  - "multiplicatively" (focusing on the relative difference), or
  - "additively" (focusing on the absolute difference).

Definition (5.11 - g-leakage)

Given prior distribution π, gain function $g\colon \mathbb{G}\mathcal{X}$, and channel C, the **multiplicative g-leakage** is

$$\mathcal{L}_g^{\times}(\pi, C) \ := \ \frac{V_g[\pi \triangleright C]}{V_g(\pi)} \quad ,$$

and the **additive g-leakage** is

$$\mathcal{L}_g^{+}(\pi, C) \ := \ V_g[\pi \triangleright C] - V_g(\pi) \quad .$$

Multiplicative and additive g-leakage

- Notice that multiplicative leakage does not really make sense if vulnerability can be both positive and negative.

  (That's why we care about non-negative vulnerabilities).

- By Theorem 5.8 (Monotonicity), the smallest possible leakage occurs when the prior- and posterior vulnerabilities are equal. Then:

  - the additive g-leakage is 0, and
  - the multiplicative g-leakage is 1.

  In both those cases, we say that there is no leakage.

Multiplicative and additive g-leakage

- If $V_g(\pi) = 0$, the quotient will be undefined in multiplicative g-leakage.

  In this case we define:

$$\mathcal{L}_g^{\times}(\pi, C) = \begin{cases} 1, & \text{if } V_g[\pi \triangleright C] \text{ is also 0,} \\ +\infty, & \text{otherwise.} \end{cases}$$

  But $+\infty$ is never needed for non-negative gain functions ($\mathbb{G}^{+}\mathcal{X}$).

Theorem (5.12)

For any non-negative gain function $g\colon \mathbb{G}^{+}\mathcal{X}$, prior π, and abstract channel C, if $V_g(\pi) = 0$ then also $V_g[\pi \triangleright C] = 0$.

Proof. If g is a non-negative gain function and $\sup_w \sum_x \pi_x g(w, x) = 0$, then for each x in the support of π we must have $g(w, x) = 0$ for every w. Since every inner of $[\pi \triangleright C]$ has support that is a subset of the support of π, it follows that the posterior g-vulnerability must also be 0. □

Multiplicative and additive g-leakage

- We next observe that the gain-function algebra (Theorems 3.14 and 5.10) together imply that each variant of g-leakage is invariant with respect to one of the gain-function operators, while possibly affected by the other.

Theorem (5.13)

For all $g\colon \mathbb{G}\mathcal{X}$, prior π, channel C, scalar constant $k \geq 0$, and tuple $\kappa\colon \mathbb{R}^{|\mathcal{X}|}$, the following hold (provided of course that $g_{+\kappa}$ is in $\mathbb{G}\mathcal{X}$):

1. *Multiplicative leakage is invariant under scaling but affected by shifting:*

$$\begin{aligned} \mathcal{L}^{\times}_{g_{\times k}}(\pi, C) &= \mathcal{L}^{\times}_{g}(\pi, C)\ , \\ \mathcal{L}^{\times}_{g_{+\kappa}}(\pi, C) &= (1-\lambda)\mathcal{L}^{\times}_{g}(\pi, C) + \lambda \qquad \textit{where} \qquad \lambda := \frac{\kappa \cdot \pi}{V_{g_{+\kappa}}(\pi)}\ . \end{aligned}$$

2. *Additive leakage is invariant under shifting but affected by scaling:*

$$\begin{aligned} \mathcal{L}^{+}_{g_{+\kappa}}(\pi, C) &= \mathcal{L}^{+}_{g}(\pi, C)\ , \\ \mathcal{L}^{+}_{g_{\times k}}(\pi, C) &= k\ \mathcal{L}^{+}_{g}(\pi, C)\ . \end{aligned}$$

Multiplicative and additive g-leakage

- **Proof.** All cases are direct consequences of Theorems 3.14 and 5.10.

  We showcase here the case of shifting for multiplicative leakage, which is the most challenging.

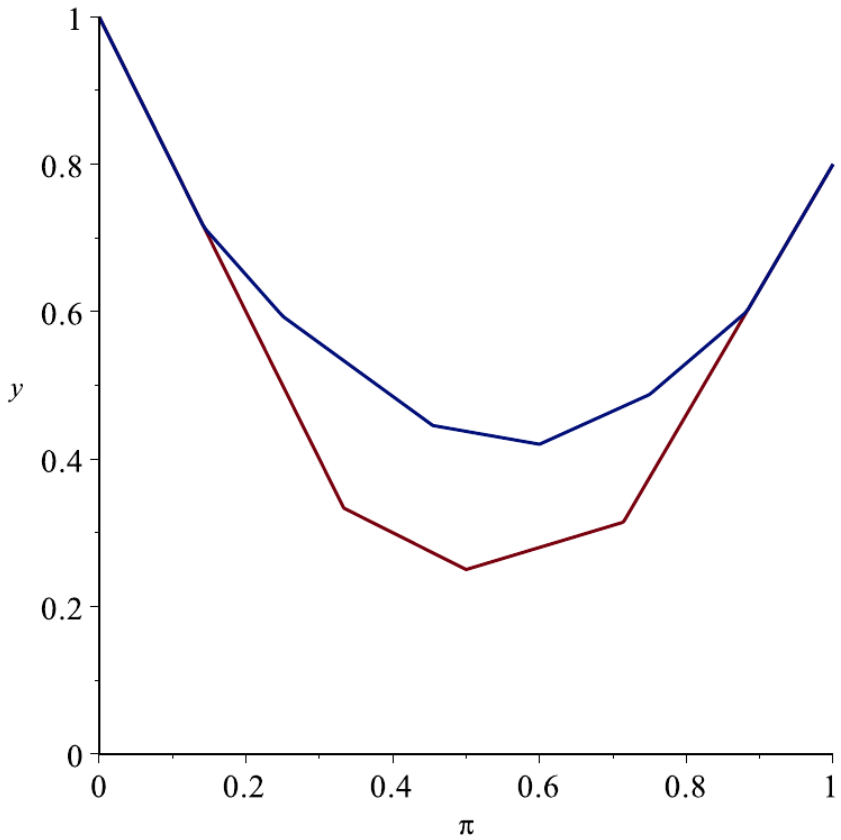
  Assume that both $V_g(\pi)$ and $V_{g_{+\kappa}}(\pi)$ are nonzero, and let $a = V_g[\pi \triangleright C]$, $b = V_g(\pi)$, and $k = \kappa \cdot \pi$.

  We have that

$$
\begin{aligned}
& (1-\lambda)\mathcal{L}^{\times}_{g}(\pi, C) + \lambda \\
= \quad & (1 - {}^{k}/_{b+k})\,\tfrac{a}{b} + {}^{k}/_{b+k} && \text{“def. of } \mathcal{L}^{\times}_{g}(\pi, C), b \neq 0, b+k \neq 0\text{”} \\
= \quad & {}^{a(b+k)}/_{b(b+k)} - {}^{ak}/_{b(b+k)} + {}^{bk}/_{b(b+k)} && \text{“algebra”} \\
= \quad & b(a+k) \,/\, b(b+k) && \text{“algebra”} \\
= \quad & V_{g_{+\kappa}}[\pi \triangleright C] \,/\, V_{g_{+\kappa}}(\pi) && \text{“ Thm. 3.14, Thm. 5.10”} \\
= \quad & \mathcal{L}^{\times}_{g_{+\kappa}}(\pi, C) \quad . && \text{“def. of } \mathcal{L}^{\times}_{g_{+\kappa}}(\pi, C)\text{”}
\end{aligned}
$$

Multiplicative and additive g-leakage

- Example 8 To gain more insight into additive and multiplicative leakage, recall that in a previous example we found the following graph for prior $V_g(\pi)$ (red) and posterior $V_g[\pi \triangleright C]$ (blue) vulnerabilities:



Multiplicative and additive g-leakage

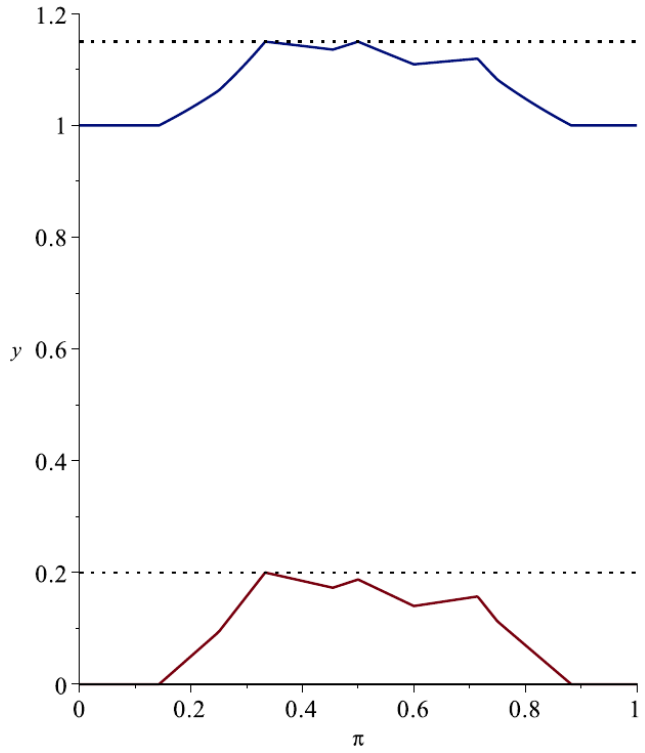
- Example 8 (Continued)

To demonstrate the effect of shifting, we use the shifted gain function g_{+1}, which adds 1 to all the gain values of g.

Notice that that shifting:

- makes g non-negative;
- increases all vulnerabilities by 1;
- has no effect on the additive leakage (even if affects multiplicative leakage).

The figure shows the additive (red) and multiplicative (blue) g_{+1}-leakage of C.

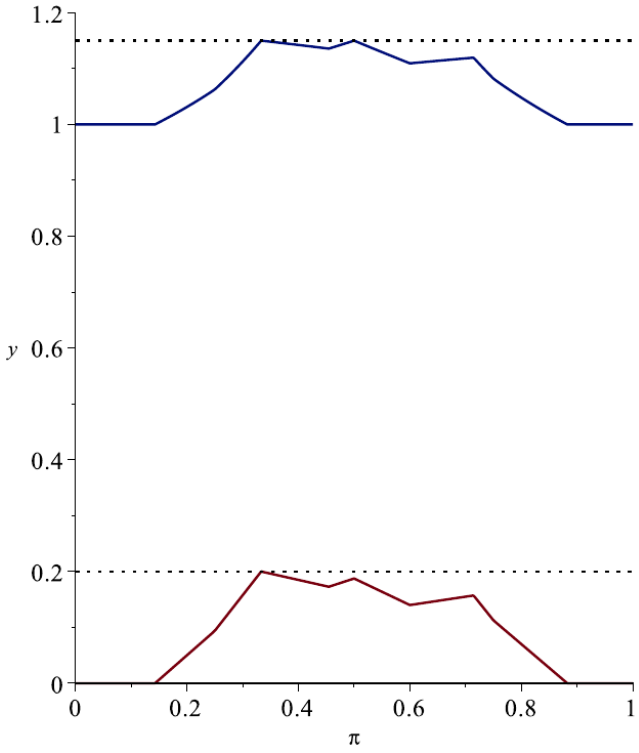


Multiplicative and additive g-leakage

- Example 8 (Continued)

Note that:

- Additive leakage ranges between 0 and about 0.2.
- Multiplicative leakage ranges between 1 and about 1.15.
- For heavily skewed priors there is no leakage: additive leakage is 0, and multiplicative leakage is 1.
- Both leakages are neither convex nor concave functions of π.
- Additive and multiplicative leakages aren't maximized always at the same prior.



A closer look at posterior Bayes vulnerability and Bayes leakage

A closer look at posterior Bayes vulnerability and Bayes leakage

- Let's look more closely at Bayes vulnerability and Bayes leakage.
- First, we notice that posterior Bayes vulnerability has clear operational significance wrt. confidentiality:

  $V_1[\pi \triangleright \mathsf{C}]$ corresponds to a smart adversary $\mathcal{A}$'s probability of winning the following game.

-- *Defender*
$X :\in \pi$ -- *Choose X according to prior.*
$Y :\in \mathsf{C}_{X,\_}$ -- *Output Y according to C.*

-- *and then Adversary*
$W :\in \mathcal{A}(\pi, \mathsf{C}, Y)$ -- *Choose guess, and...*
`if` $W{=}X$ `then` "win" `else` "lose" -- *...win if correct.*

A closer look at posterior Bayes vulnerability and Bayes leakage

- Recall that joint matrix J is defined by $\mathsf{J}{=}\pi{\triangleright}\mathsf{C}$ so that $\mathsf{J}_{x,y} = \pi_x \mathsf{C}_{x,y}$.

  There's a easy way to calculate posterior Bayes vulnerability from J.

Theorem (5.15)

Posterior Bayes vulnerability is the sum of the column maximums of the joint matrix J*:*

$$V_1[\pi{\triangleright}\mathsf{C}] \quad = \quad \sum_{y \in \mathcal{Y}} \max_{x \in \mathcal{X}} \mathsf{J}_{x,y} \quad .$$

Proof. Given channel matrix C and prior π, we reason

$$\begin{array}{lll}
 & V_1[\pi{\triangleright}\mathsf{C}] & \\
= & V_{g_{\mathsf{id}}}[\pi{\triangleright}\mathsf{C}] & \text{“Thm. 3.6”} \\
= & \sum_y \max_w \sum_x \pi_x \, \mathsf{C}_{x,y} \, g_{\mathsf{id}}(w,x) & \text{“Thm. 5.7”} \\
= & \sum_y \max_w \pi_w \mathsf{C}_{w,y} & \text{“}g_{\mathsf{id}}(w,x) = (1 \text{ if } w{=}x \text{ else } 0)\text{; one-point rule”} \\
= & \sum_y \max_x \mathsf{J}_{x,y} \quad . & \text{“rename } w \text{ to } x\text{; definition of J”}
\end{array}$$

A closer look at posterior Bayes vulnerability and Bayes leakage

- Notice that prior Bayes vulnerability can also be expressed in terms of J:

$$V_1(\pi) \quad = \quad \max_{x \in \mathcal{X}} \pi_x \quad = \quad \max_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} \mathsf{J}_{x,y} \quad .$$

- Thus:
  - posterior Bayes vulnerability is the sum of the column maximums of J; while
  - prior Bayes vulnerability is the maximum of the row sums of J.
- Hence, $V_1[\pi \triangleright \mathsf{C}] > V_1(\pi)$ only if the column maximums of J do not all occur in the same row.

  Equivalently, there is no Bayes leakage if the adversary's best guess about X is unaffected by the output Y.

  That can sometimes be surprising, as in the following example, which illustrates the so-called **base-rate fallacy**.

A closer look at posterior Bayes vulnerability and Bayes leakage

- Example 9 **Imperfect disease test.** Suppose that C is the channel matrix of a good, but imperfect, test for a certain disease:

C	*positive*	*negative*
disease	$9/10$	$1/10$
no disease	$1/10$	$9/10$

Suppose that for the population under consideration (say, age 40–50, no symptoms, no family history) the prior π is heavily biased towards "no disease":

$$\pi = (1/100, 99/100) \quad .$$

A closer look at posterior Bayes vulnerability and Bayes leakage

- Example 9 (Continued)

Then, although the channel might appear to be quite reliable, we find that there is no Bayes leakage.

For we find that the hyper-distribution $[\pi \triangleright C]$ is

$[\pi \triangleright C]$	$27/250$	$223/250$
disease	$1/12$	$1/892$
no disease	$11/12$	$891/892$

,

reflecting the fact that:

- a positive test result (corresponding to the first inner) increases the probability of disease from $1/100$ to $1/12$, while
- a negative test result (corresponding to the second inner) decreases it to $1/892$.

A closer look at posterior Bayes vulnerability and Bayes leakage

- Example 9 (Continued)

Nevertheless, C's output is useless in winning the Posterior Bayes-vulnerability Game, since the doctor's best action is always to guess "no disease".

And indeed we see that

$$V_1[\pi \triangleright \mathsf{C}] \quad = \quad {}^{27}/_{250} \cdot {}^{11}/_{12} + {}^{223}/_{250} \cdot {}^{891}/_{892} \quad = \quad {}^{99}/_{100} \quad .$$

We can calculate $V_1[\pi \triangleright \mathsf{C}]$ more directly, using Thm. 5.15. Since J is

J	*positive*	*negative*
disease	${}^{9}/_{1000}$	${}^{1}/_{1000}$
no disease	${}^{99}/_{1000}$	${}^{891}/_{1000}$

,

we find that

$$V_1[\pi \triangleright \mathsf{C}] \quad = \quad {}^{99}/_{1000} + {}^{891}/_{1000} \quad = \quad {}^{99}/_{100} \quad .$$

A closer look at posterior Bayes vulnerability and Bayes leakage

- Example 9 (Continued)

  Since

$$V_1[\pi \triangleright \mathsf{C}] \quad = \quad {}^{99}/_{100} \quad = \quad V_1(\pi) \ ,$$

  we see that the prior- and posterior vulnerabilities are equal.

  Hence there is no Bayes leakage for this π and C. ◁

A closer look at posterior Bayes vulnerability and Bayes leakage

- Multiplicative Bayes leakage on a uniform prior can be calculated very easily.

Theorem

For any channel matrix C*, if* $\boldsymbol{\vartheta}$ *is uniform then the multiplicative Bayes leakage is equal to the sum of the column maximums of* C*:*

$$\mathcal{L}_1^{\times}(\boldsymbol{\vartheta}, \mathsf{C}) \quad = \quad \sum_{y \in \mathcal{Y}} \max_{x \in \mathcal{X}} \mathsf{C}_{x,y} \quad .$$

Proof. It's a corollary of Theorem 5.15. If $\mathcal{X}$ has size n and $\boldsymbol{\vartheta}$ is uniform:

$$\begin{aligned} \mathcal{L}_1^{\times}(\boldsymbol{\vartheta}, \mathsf{C}) \quad &= \quad \frac{V_1[\boldsymbol{\vartheta} \triangleright \mathsf{C}]}{V_1(\boldsymbol{\vartheta})} \quad = \quad \frac{\sum_y \max_x \mathsf{J}_{x,y}}{1/n} \\ &= \quad \frac{\sum_y \max_x \frac{\mathsf{C}_{x,y}}{n}}{1/n} \quad = \quad \sum_y \max_x \mathsf{C}_{x,y} \, . \end{aligned}$$

□

Measuring leakage with Shannon entropy

Measuring leakage with Shannon entropy

- Recall that **Shannon entropy**, which is at the heart of **information theory**, is defined by

$$H(\pi) \;:=\; -\sum_{x \in \mathcal{X}} \pi_x \log_2 \pi_x \;.$$

It gives an obvious way to measure information leakage (from the uncertainty rather than the vulnerability perspective):

1. prior uncertainty is $H(\pi)$, traditionally denoted $H(X)$;
2. posterior uncertainty is $H[\pi \triangleright \mathsf{C}]$, traditionally denoted $H(X|Y)$; and
3. information leakage is the difference $H(\pi) - H[\pi \triangleright \mathsf{C}]$, which is called **mutual information** and traditionally denoted $I(X;Y)$.

- Early researchers in QIF adopted this as the definition of information leakage.

But by now we are accustomed to ask:

"What is the operational significance of mutual information with respect to confidentiality?"

Measuring leakage with Shannon entropy

- As we have seen, **Shannon's source-coding theorem** gives such a significance in terms of the expected number of arbitrary *yes/no* questions about X needed to determine its value.

  However:

  - this scenario is perhaps not particularly common in practice; and
  - high Shannon entropy does not guarantee low Bayes vulnerability.

Example 10 Let's revisit a scenario we have already seen.

Suppose that X is a 64-bit unsigned integer, so that

$$\mathcal{X} = \{0, 1, 2, \ldots, 2^{64} - 1\} \ ,$$

and suppose that $\vartheta = (2^{-64}, 2^{-64}, \ldots, 2^{-64})$ is uniform.

First, we have prior uncertainty

$$H(\vartheta) = \log_2 2^{64} = 64 \ .$$

Measuring leakage with Shannon entropy

- Example 10 (Continued)

Now consider the following channel C:

$$Y := \texttt{ if } (X \texttt{ mod } 8) = 0 \texttt{ then } X \texttt{ else } 1 \quad .$$

Now, for posterior uncertainty $H[\vartheta \triangleright \mathrm{C}]$, we calculate $H(p_{X|y})$ for each y.

- In the `then` branch, Y can be any of the 2^{61} multiples of 8 in the given range.

  For each such y, $p_{X|y}$ is the point distribution $[y]$, so $H(p_{X|y}) = 0$.

  Moreover, for each such y we have $p_Y(y) = 2^{-64}$.
- In the `else` branch, Y is 1, and X can be any value that isn't a multiple of 8.

  Hence $p_{X|1}$ is uniform on the $7/8 \cdot 2^{64} = 7 \cdot 2^{61}$ values of X, and

  $$H(p_{X|1}) \quad = \quad \log_2(7 \cdot 2^{61}) \quad = \quad \log_2 7 + 61 \quad \approx \quad 2.807 + 61 \quad \approx \quad 63.807 \quad .$$

  Moreover $p_Y(1) = 7/8$.

Measuring leakage with Shannon entropy

- Example 10 (Continued)

Finally, by putting those together we obtain

$$H[\vartheta \triangleright \mathrm{C}] \quad \approx \quad 2^{61} \cdot 2^{-64} \cdot 0 \; + \; {}^{7}/_{8} \cdot 63.807 \quad \approx \quad 55.831$$

and

$$I(X;Y) \quad = \quad H(\vartheta) - H[\vartheta \triangleright \mathrm{C}] \quad \approx \quad 64 - 55.831 \quad \approx \quad 8.169 \quad .$$

If we measure leakage using mutual information, channel C leaks just 8.169 bits out of 64, leaving a posterior uncertainty of 55.831 bits.

This suggests that C doesn't harm the confidentiality of X very much.

But notice that whenever Y is not equal to 1, which happens with probability ${}^{1}/_{8}$, the exact value of X is revealed, showing that C is very bad indeed!

So mutual-information leakage can be quite misleading wrt. confidentiality.

Measuring leakage with Shannon entropy

- Example 10 (Continued)

In contrast, consider the multiplicative Bayes leakage $\mathcal{L}_1^{\times}(\vartheta, \mathrm{C})$.

Using Thm. 5.17, that can be calculated as the sum of the column maximums of C, which (since C is deterministic) is just the number of possible output values of C. Note that:

- the `then` branch gives 2^{61} possible values, and
- the `else` branch gives 1 more.

Hence we have

$$\mathcal{L}_1^{\times}(\vartheta, \mathrm{C}) = 2^{61} + 1 \ ,$$

which implies that the posterior Bayes vulnerability is

$$V_1[\vartheta \triangleright \mathrm{C}] \quad = \quad V_1(\vartheta) \cdot \mathcal{L}_1^{\times}(\vartheta, \mathrm{C}) \quad = \quad 2^{-64} \cdot (2^{61} + 1) \quad = \quad 1/8 + 2^{-64} \quad ,$$

which is very high indeed if compared to the prior vulnerability.

Measuring leakage with Shannon entropy

- Example 10 (Continued)

We remark that we can get the same result directly by observing that the adversary's chance of guessing X correctly given the output of C is

- 1 in the case when $Y \neq 1$, which happens with probability $1/8$, and
- $8/7 \cdot 2^{-64}$ when $Y = 1$, which happens with probability $7/8$.

Hence we get

$$V_1[\vartheta \triangleright \mathsf{C}] \quad = \quad 1/8 \cdot 1 + 7/8 \cdot 8/7 \cdot 2^{-64} \quad = \quad 1/8 + 2^{-64} \quad ,$$

which agrees with the previous calculation. ◁

More properties of posterior g-vulnerability and g-leakage

More properties of posterior g-vulnerability and g-leakage: A matrix-based formulation of posterior g-vulnerability

- Here we show that posterior g-vulnerability, like posterior Bayes vulnerability, can be formulated in terms of the joint matrix J.

Theorem (5.18)

For $g\colon \mathbb{G}^{\text{fin}}\mathcal{X}$, posterior g-vulnerability is the sum of the column maximums of the matrix GJ :

$$V_g[\pi \triangleright C] \quad = \quad \sum_{y \in \mathcal{Y}} \max_{w \in \mathcal{W}} (GJ)_{w,y} \quad .$$

Proof. This follows easily from Theorem 5.7:

$$\begin{array}{lll} & V_g[\pi \triangleright C] & \\ = & \sum_y \max_w \sum_x \pi_x C_{x,y} g(w,x) & \text{“Thm. 5.7”} \\ = & \sum_y \max_w \sum_x J_{x,y} G_{w,x} & \text{“definitions of G and J”} \\ = & \sum_y \max_w (GJ)_{w,y} \quad . & \text{“definition of matrix multiplication”} \end{array}$$

□

More properties of posterior g-vulnerability and g-leakage: A matrix-based formulation of posterior g-vulnerability

- There is an important operational meaning to the column maximums of GJ.

  If the maximum of column y occurs at row w, it means that action w is optimal for the adversary, given output y.

- Moreover, since the matrix representation of g_{id} is the $\mathcal{X}\times\mathcal{X}$-indexed identity matrix I and $\mathsf{I}\cdot\mathsf{J}=\mathsf{J}$, we can see that Theorem 5.15, which says

$$V_1[\pi\triangleright\mathsf{C}] \quad = \quad \sum_{y\in\mathcal{Y}} \max_{x\in\mathcal{W}} \mathsf{J}_{x,y}$$

  is hence a simple corollary to Theorem 5.18, which says

$$V_g[\pi\triangleright\mathsf{C}] \quad = \quad \sum_{y\in\mathcal{Y}} \max_{w\in\mathcal{W}} (\mathsf{GJ})_{w,y} \ .$$

More properties of posterior g-vulnerability and g-leakage: A matrix-based formulation of posterior g-vulnerability

- **Example 11** Consider again a scenario having a secret set $\mathcal{X} = \{x_1, x_2\}$ with prior $\pi = (0.3, 0.7)$.

  Consider action set $\mathcal{W} = \{w_1, w_2, w_3, w_4, w_5\}$ Suppose that channel C is leading to a gain function having values as in

G	x_1	x_2
w_1	-1.0	1.0
w_2	0.0	0.5
w_3	0.4	0.1
w_4	0.8	-0.9
w_5	0.1	0.2

.

C	y_1	y_2
x_1	0.75	0.25
x_2	0.25	0.75

.

In a previous example we went through a lengthy calculation to show that

$$V_g[\pi \triangleright \mathsf{C}] = 0.5575 \ .$$

Theorem 5.18 lets us calculate that result much more easily.

More properties of posterior g-vulnerability and g-leakage: A matrix-based formulation of posterior g-vulnerability

- Example 11 (Continued)

First we compute the matrix product GJ as follows:

G	x_1	x_2
w_1	-1.0	1.0
w_2	0.0	0.5
w_3	0.4	0.1
w_4	0.8	-0.9
w_5	0.1	0.2

J	y_1	y_2
x_1	0.225	0.075
x_2	0.175	0.525

$=$

GJ	y_1	y_2
w_1	-0.0500	0.4500
w_2	0.0875	0.2625
w_3	0.1075	0.0825
w_4	0.0225	-0.4125
w_5	0.0575	0.1125

.

Then, looking at the column maximums of GJ, we obtain

$$V_g[\pi \triangleright \mathrm{C}] \quad = \quad \sum_{y \in \mathcal{Y}} \max_{w \in \mathcal{W}} (\mathrm{GJ})_{w,y} \quad = \quad 0.1075 + 0.4500 \quad = \quad 0.5575 \quad .$$

More properties of posterior g-vulnerability and g-leakage: A matrix-based formulation of posterior g-vulnerability

- We can also formulate prior g-vulnerability in terms of GJ.

Theorem (5.20)

For $g\colon \mathbb{G}^{\text{fin}}\mathcal{X}$, prior g-vulnerability is the maximum of the row sums of the matrix GJ:

$$V_g(\pi) \quad = \quad \max_{w \in \mathcal{W}} \sum_{y \in \mathcal{Y}} (GJ)_{w,y} \quad .$$

Proof. We can reason as follows:

$$\begin{array}{lll}
& V_g(\pi) & \\
= & \max_w \sum_x \pi_x g(w,x) & \\
= & \max_w \sum_x \pi_x g(w,x) \sum_y \mathsf{C}_{x,y} & \text{“}\textstyle\sum_y \mathsf{C}_{x,y} = 1\text{”} \\
= & \max_w \sum_y \sum_x \pi_x \mathsf{C}_{x,y} g(w,x) & \text{“reorganizing sum”} \\
= & \max_w \sum_y \sum_x \mathsf{J}_{x,y} \mathsf{G}_{w,x} & \text{“definitions of J and G”} \\
= & \max_w \sum_y (\mathsf{GJ})_{w,y} \quad . & \text{“definition of matrix multiplication”}
\end{array}$$

□

More properties of posterior g-vulnerability and g-leakage: A matrix-based formulation of posterior g-vulnerability

- Note that

  Theorem 5.18:

  $$V_g[\pi \triangleright \mathsf{C}] = \sum_{y \in \mathcal{Y}} \max_{w \in \mathcal{W}} (\mathsf{GJ})_{w,y} \ ,$$

  and Theorem 5.20:

  $$V_g(\pi) = \max_{w \in \mathcal{W}} \sum_{y \in \mathcal{Y}} (\mathsf{GJ})_{w,y}$$

  imply that

  $$V_g[\pi \triangleright \mathsf{C}] = V_g(\pi)$$

  just when all the column maximums of GJ occur in the same row w, say.

  In this case:

  - The adversary's best action is always that w, i.e. is the same, regardless of the output of C.
  - For this g and π, channel C is useless to her (no leakage).

More properties of posterior g-vulnerability and g-leakage: A trace-based formulation of posterior g-vulnerability

- We now build on Theorem 5 .18 to develop an important formulation of posterior g-vulnerability as a matrix trace.

Definition (5.21 - Trace of square matrix)

If M is a square matrix, then its **trace** is the sum of its diagonal entries: thus $\mathbf{tr}(\mathsf{M}) = \sum_i \mathsf{M}_{i,i}$.

More properties of posterior g-vulnerability and g-leakage: A trace-based formulation of posterior g-vulnerability

- Note that the formulation of posterior g-vulnerability can be:
  - Theorem 5.7:

$$V_g[\pi \triangleright \mathsf{C}] = \sum_{y \in \mathcal{Y}} \max_{w \in \mathcal{W}} \sum_{x \in \mathcal{X}} \pi_x \mathsf{C}_{x,y} g(w,x)$$

  - Theorem 5.18:

$$V_g[\pi \triangleright \mathsf{C}] = \sum_{y \in \mathcal{Y}} \max_{w \in \mathcal{W}} (\mathsf{GJ})_{w,y}$$

  Both formulate $V_g[\pi \triangleright \mathsf{C}]$ as a summation over $\mathcal{Y}$ of a maximization over $\mathcal{W}$.

  This means means that for each y in $\mathcal{Y}$, a best w in $\mathcal{W}$ must be selected.

- This selection can be reified as a **strategy** S, and expressed as a channel matrix from $\mathcal{Y}$ to $\mathcal{W}$ that tells which action w is best, given output y.
- S can be probabilistic, since there could be more than one w that is optimal for a given y.

  That is, row y of S can give some distribution to the actions w that are optimal given output y.

More properties of posterior g-vulnerability and g-leakage: A trace-based formulation of posterior g-vulnerability

- Now we show that taking the sum of the column maximums of a matrix can be expressed neatly as a maximization over strategies of a matrix trace.

Lemma (5.22)

If M *is any* $\mathcal{W}\times\mathcal{Y}$*-indexed matrix, then*

$$\max_{\mathsf{S}} \mathbf{tr}(\mathsf{SM}) \quad = \quad \sum_{y\in\mathcal{Y}} \max_{w\in\mathcal{W}} \mathsf{M}_{w,y} \quad ,$$

where S *ranges over strategies from* $\mathcal{Y}$ *to* $\mathcal{W}$*. Moreover, the maximum is always realized on some deterministic strategy* S*.*

More properties of posterior g-vulnerability and g-leakage: A trace-based formulation of posterior g-vulnerability

- **Proof.** By the definition of matrix multiplication,

$$(\mathsf{SM})_{y,y} = \sum_{w} \mathsf{S}_{y,w} \mathsf{M}_{w,y} \ ,$$

which means that $(\mathsf{SM})_{y,y}$ is a convex combination of column y of M.

But any convex combination must be less than or equal to the maximum, i.e.

$$(\mathsf{SM})_{y,y} \leq \max_{w} \mathsf{M}_{w,y} \ ,$$

with equality holding if S satisfies the condition that $\mathsf{S}_{y,w} > 0$ only if $\mathsf{M}_{w,y}$ is a maximum in column y. (I.e. non-maximum elements of a column aren't used in the convex combination.)

In particular, this condition is satisfied by a deterministic S that selects one best w for each y.

So if S is any matrix satisfying the condition, then SM contains the column maximums of M on its diagonal, and the lemma follows. □

More properties of posterior g-vulnerability and g-leakage: A trace-based formulation of posterior g-vulnerability

- A benefit of trace-based formulations is that trace satisfies a remarkable cyclic property.

Theorem (5.23)

If matrix product AB *is square, than* $\mathbf{tr}(\mathsf{AB}) = \mathbf{tr}(\mathsf{BA})$.

Proof. If A is indexed by $\mathcal{X}\times\mathcal{Y}$ and B is indexed by $\mathcal{Y}\times\mathcal{X}$, then we have

$$
\begin{array}{lll}
 & \mathbf{tr}(\mathsf{AB}) & \\
= & \sum_x (\mathsf{AB})_{x,x} & \\
= & \sum_x \sum_y \mathsf{A}_{x,y}\mathsf{B}_{y,x} & \\
= & \sum_y \sum_x \mathsf{B}_{y,x}\mathsf{A}_{x,y} & \text{“reorganizing sum”} \\
= & \sum_y (\mathsf{BA})_{y,y} & \\
= & \mathbf{tr}(\mathsf{BA}) \quad . &
\end{array}
$$

□

More properties of posterior g-vulnerability and g-leakage: A trace-based formulation of posterior g-vulnerability

- Moreover, by the associativity of matrix multiplication, the cyclic property generalizes to a product of any number of matrices:

$$\mathbf{tr}(\mathsf{ABCD}) \quad = \quad \mathbf{tr}(\mathsf{BCDA}) \quad = \quad \mathbf{tr}(\mathsf{CDAB}) \quad = \quad \mathbf{tr}(\mathsf{DABC}).$$

- We now show a trace-based formulation of posterior g-vulnerability.

Theorem (5.24)

For any $g\colon \mathbb{G}^{\mathsf{fin}}\mathcal{X}$, prior π, and channel matrix $\mathsf{C}\colon \mathcal{X} \to\!\triangleright \mathcal{Y}$, we have

$$V_g[\pi \triangleright \mathsf{C}] \quad = \quad \max_{\mathsf{S}} \ \mathbf{tr}(G\, \lceil \pi \rfloor\, \mathsf{CS}) \quad ,$$

where S ranges over strategies from $\mathcal{Y}$ to $\mathcal{W}$ and $\lceil \pi \rfloor$ denotes a diagonal matrix with π on its diagonal.

Moreover, the maximum can always be realized by a deterministic strategy.

More properties of posterior g-vulnerability and g-leakage: A trace-based formulation of posterior g-vulnerability

- **Proof.** We reason as follows:

$$
\begin{array}{lll}
 & V_g[\pi \triangleright C] & \\
= & \sum_y \max_w (GJ)_{w,y} & \text{“Thm. 5.18”} \\
= & \max_S \mathbf{tr}(SGJ) & \text{“Lem. 5.22”} \\
= & \max_S \mathbf{tr}(SG\,\lceil \pi \rfloor\, C) & \text{“}J = \pi \triangleright C = \lceil \pi \rfloor C\text{”} \\
= & \max_S \mathbf{tr}(G\,\lceil \pi \rfloor\, CS) \quad . & \text{“cyclic property of trace”}
\end{array}
$$

□

- This trace-based formulation of posterior g-vulnerability lets us derive relationships among leakage measures by exploiting two key properties:
  1. the associativity of matrix multiplication; and
  2. the cyclic property of trace.

  The following two examples illustrate that we can regroup the matrices in $\mathbf{tr}(G\,\lceil \pi \rfloor\, CS)$ to obtain new interpretations of $V_g[\pi \triangleright C]$.

More properties of posterior g-vulnerability and g-leakage: A trace-based formulation of posterior g-vulnerability

- Example 12 **Moving the prior into the gain function.** For any prior π, observe that the matrix $\lceil \pi \rfloor$ can be factored into $\lceil \rho \rfloor \lceil \vartheta \rfloor$, where:
  - ϑ is the uniform distribution on $\mathcal{X}$, and
  - $\rho_x = \pi_x / \vartheta_x$. (Note that ρ is typically not a probability distribution.)

  Hence we have

$$\mathbf{tr}(\mathsf{G}\lceil \pi \rfloor \mathsf{C}\mathsf{S}) \quad = \quad \mathbf{tr}(\mathsf{G}(\lceil \rho \rfloor \lceil \vartheta \rfloor)\mathsf{C}\mathsf{S}) \quad = \quad \mathbf{tr}((\mathsf{G}\lceil \rho \rfloor)\lceil \vartheta \rfloor \mathsf{C}\mathsf{S}) \quad ,$$

  which implies that we can rewrite

$$V_g[\pi \triangleright \mathsf{C}] = V_{g'}[\vartheta \triangleright \mathsf{C}] \ ,$$

  where g' is the gain function represented by $\mathsf{G}\lceil \rho \rfloor$.

  (Notice in contrast that moving the gain function into the prior is not possible except in the special case where G is diagonal.) ◁

More properties of posterior g-vulnerability and g-leakage: A trace-based formulation of posterior g-vulnerability

- Example 13 **Moving the channel matrix into the strategy.** Since any channel matrix C trivially factors into IC, where I is the $\mathcal{X}\times\mathcal{X}$-indexed identity matrix, we have

$$\mathbf{tr}(G^{\lceil}\pi^{\rfloor}CS) \quad = \quad \mathbf{tr}(G^{\lceil}\pi^{\rfloor}(IC)S) \quad = \quad \mathbf{tr}(G^{\lceil}\pi^{\rfloor}I(CS)) \quad ,$$

where I is now the channel and CS can be seen as a strategy from $\mathcal{X}$ to $\mathcal{W}$.

There's no guarantee that CS is optimal but we can still derive:

$$\begin{array}{lll} & V_g[\pi\triangleright C] & \\ = & \max_S \mathbf{tr}(G^{\lceil}\pi^{\rfloor}CS) & \text{“Thm. 5.24”} \\ = & \max_S \mathbf{tr}(G^{\lceil}\pi^{\rfloor}I(CS)) & \text{“as above: I is identity matrix”} \\ \leq & \max_{S'} \mathbf{tr}(G^{\lceil}\pi^{\rfloor}IS') & \text{“CS might not be optimal”} \\ = & V_g[\pi\triangleright I] \quad . & \end{array}$$

Thus the posterior g-vulnerability on any channel C never exceeds that on the channel $\mathbb{O}$ that leaks everything, which as a reduced channel matrix is I. $\triangleleft$

Examples of channels and their leakage

Examples of channels and their leakage

- Let's now consider some example channels and their leakage.

- Example 14 Recall the gain functions for a password database.

  Here the secret is an array X containing 10-bit passwords for a set U of 1000 users, so that for u in U the entry $X[u]$ is user u's password.

  One possibility is to use g_{id}, corresponding to Bayes vulnerability, which represents an adversary trying to guess the entire array X.

  A second possibility is to use g_{pw}, which represents an adversary interested in guessing some user's password, with no preference as to whose it is.

  Here we have

$$\mathcal{W} \quad = \quad \{(u,x) \mid u \in U \text{ and } 0 \leq x \leq 1023\}$$

  and

$$g_{\mathsf{pw}}((u,x),X) \quad = \quad \begin{cases} 1 & \text{if } X[u] = x \\ 0 & \text{otherwise} \end{cases} .$$

Examples of channels and their leakage

- Example 14 (Continued)

A third possibility is to use g_{Gates}, which (we recall) specifies that one of the users is Bill Gates, whose password is much more valuable than the other passwords.

Here we have the same $\mathcal{W}$ as for g_{pw} and

$$g_{\text{Gates}}((u,x),X) \quad = \quad \begin{cases} 1 & \text{if } u = \text{Bill Gates and } x = X[u] \\ 0.01 & \text{if } u \neq \text{Bill Gates and } x = X[u] \\ 0 & \text{otherwise} \quad . \end{cases}$$

Examples of channels and their leakage

- Example 14 (Continued)

Suppose now that the prior π is uniform, meaning that the 1000 passwords in X are independent and uniformly distributed on [0..1023].

Consider channel C, which reveals some randomly chosen user's password:

$$\begin{array}{lll} u :\in & \text{uniform}(U) & \text{--}\textit{Choose } u \textit{ uniformly from } U. \\ Y := & (u,\ X[u]) & \text{--}\textit{Leak password via the observable output } Y. \end{array}$$

Using Bayes vulnerability (i.e. g_{id}), we find that

- $V_1(\pi) = 2^{-10,000}$, since there are 10,000 completely unknown bits in X.
- $V_1[\pi \triangleright \mathsf{C}] = 2^{-9990}$, since now 10 of the 10,000 bits of X are known.

Hence the multiplicative Bayes leakage is

$$\mathcal{L}_1^{\times}(\pi, \mathsf{C}) \quad = \quad 2^{-9990}/2^{-10,000} \quad = \quad 2^{10}\ .$$

Examples of channels and their leakage

- Example 14 (Continued)

In contrast, if we measure leakage using g_{pw}, then we find:

- The prior g_{pw}-vulnerability is 2^{-10}.

  This follows from the fact that the expected gain from every action (u, x) in $\mathcal{W}$ is 2^{-10} since, for every u, the entry $X[u]$ is uniformly distributed on [0..1023].

- The posterior g_{pw}-vulnerability is $V_{g_{\text{pw}}}[\pi \triangleright \mathsf{C}] = 1$.

  This follows from the fact that since given that $Y = (u, X[u])$ the adversary can perform action $(u, X[u])$ and be sure of getting gain 1.

Hence the multiplicative g_{pw}-leakage is

$$\mathcal{L}^{\times}_{g_{\text{pw}}}(\pi, \mathsf{C}) = 1/2^{-10} = 2^{10} \ .$$

Examples of channels and their leakage

- Example 14 (Continued)

It's interesting that the multiplicative leakage is the same, that is 2^{10}, for both Bayes vulnerability and g_{pw}-vulnerability.

On the other hand, if we look at additive leakage, then:

- With Bayes vulnerability we get a negligible additive leakage

$$\mathcal{L}_1^+(\pi, \mathsf{C}) = 2^{-9990} - 2^{-10,000} \ .$$

- With g_{pw}-vulnerability we get large additive leakage

$$\mathcal{L}_{g_{\text{pw}}}^+(\pi, \mathsf{C}) = 1 - 2^{-10} \approx 0.999 \ .$$

In summary, we find with Bayes vulnerability that:

- C's multiplicative leakage is large while its additive leakage is negligible, while
- with g_{pw}-vulnerability both C's multiplicative and additive leakage are large. ◁

Examples of channels and their leakage

- However, the situation in the previous example could be reversed with other channels and gain functions.

- Example 15 Consider that

  (a) under gain function g_1 the vulnerability increases from 0.4 to 0.7; and

  (b) under gain function g_2 the vulnerability increases from 0.000001 to 0.300001.

  Then the additive leakage is 0.3 in both cases, but:

  (a) the multiplicative leakage under g_1 is modest (1.75); while

  (b) under g_2 it is huge $(300,001)$. ◁

  Those examples suggest that perhaps leakage should be judged to be "significant" only if both the multiplicative- and additive leakages are "large".

Examples of channels and their leakage

- Example 16 Let's go back to our example with 1000 passwords, each consisting of 10 bits.

  It's interesting to consider a variant of channel C that selects 10 random users and leaks just the last bit of each of their passwords.

  In this case:

  - The multiplicative Bayes leakage is still the same 2^{10}, since the variant still reveals 10 bits of X to the adversary.
  - But the multiplicative g_{pw}-leakage is now only 2, because the posterior g_{pw}-vulnerability is just 2^{-9}, since now at least 9 bits of each user's password remain unknown.

  Thus gain function g_{pw} captures the fact that some sets of 10 bits are worth more than others.

Examples of channels and their leakage

- Example 16 (Continued)

Let's now consider channel D, which leaks Bill Gates' password with probability $1/10$ but otherwise leaks nothing:

$$\begin{array}{ll} n :\in & \text{uniform}(0..9) \\ Y := & \texttt{if } n{=}0 \texttt{ then } X[\text{“Bill Gates”}] \texttt{ else } 0 \end{array}$$

If we consider channel D under multiplicative Bayes leakage, we find that

$$V_1[\pi \triangleright \mathsf{D}] \quad = \quad 1/10 \cdot 2^{-9990} + 9/10 \cdot 2^{-10,000} \ ,$$

which means that its multiplicative Bayes leakage is

$$\mathcal{L}_1^{\times}(\pi, \mathsf{D}) \quad = \quad \frac{1/10 \cdot 2^{-9990} + 9/10 \cdot 2^{-10,000}}{2^{-10,000}} \quad = \quad 1/10 \cdot 2^{10} + 9/10 \quad = \quad 103.3 \quad ,$$

which is much less that C's multiplicative Bayes leakage of 1024.

Examples of channels and their leakage

- Example 16 (Continued)

But if we analyze the multiplicative g_{Gates}-leakage of channels C and D, we find that the situation is reversed.

First note that

$$V_{g_{\text{Gates}}}(\pi) = 2^{-10} \ ,$$

since action (Bill Gates, x) has expected gain 2^{-10} for every x.

Next note that channel C's posterior g_{Gates}-vulnerability is

$$V_{g_{\text{Gates}}}[\pi \triangleright \mathsf{C}] \quad = \quad {}^{1}\!/_{1000} \cdot 1 + {}^{999}\!/_{1000} \cdot 0.01 \quad = \quad 0.01099$$

since the best action is always $(u, X[u])$, which gives gain 1 if $u = \text{Bill Gates}$ and gain 0.01 otherwise.

Hence

$$\mathcal{L}^{\times}_{g_{\text{Gates}}}(\pi, \mathsf{C}) \quad = \quad {}^{0.01099}\!/_{2^{-10}} \quad = \quad 11.25376 \quad .$$

Examples of channels and their leakage

- **Example 16** (Continued)

  In contrast, channel D's posterior g_{Gates}-vulnerability is

$$V_{g_{\text{Gates}}}[\pi \triangleright \text{D}] \quad = \quad {}^{1}\!/_{10} \cdot 1 + {}^{9}\!/_{10} \cdot 2^{-10} \quad = \quad 0.10087890625 \quad ,$$

  since the best action is always (Bill Gates, Y), which gives gain 1 if $n = 0$ and gives gain 2^{-10} otherwise.

  Hence

$$\mathcal{L}^{\times}_{g_{\text{Gates}}}(\pi, \text{D}) \quad = \quad {}^{0.10087890625}\!/_{2^{-10}} \quad = \quad 103.3 \quad ,$$

  which is much greater than C's leakage of 11.25376. ◁

Max-case posterior g-vulnerability

Max-case posterior g-vulnerability

- As we have discussed, an alternative way to define posterior g-vulnerability of $[\pi \triangleright C]$ is to consider the maximum g-vulnerability over all of the inners.

  This represents the worst possible "world" that C can give, ignoring the question of how likely that world might be.

Definition (5.29 - max-case posterior g-vulnerability)

Given prior π, and g in $\mathbb{G}\mathcal{X}$ and channel C, let $[\pi \triangleright C] = \sum_i a_i[\delta^i]$ (recalling that the outer probabilities a_i are nonzero).

Then the **max-case posterior g-vulnerability** $V_g^{\max}[\pi \triangleright C]$ is defined as the maximum value of V_g over all the inners:

$$V_g^{\max}[\pi \triangleright C] \;:=\; \max_i V_g(\delta^i) \quad .$$

(Notice that this definition pays no attention to the outer probabilities.)

Max-case posterior g-vulnerability

- On the other hand, for many channels the max-case posterior g-vulnerability is small, and in this case it is useful in offering stronger guarantees than those provided by the usual expectation-based posterior g-vulnerability.

  This is captured in the following theorem.

Theorem (5.30)

For any $g\colon \mathbb{G}\mathcal{X}$ and π and C, we have $V_g[\pi \triangleright C] \leq V_g^{\max}[\pi \triangleright C]$.

Proof. Let $[\pi \triangleright C] = \sum a_i[\delta^i]$. Then we have

$$V_g[\pi \triangleright C] \quad = \quad \sum_i a_i V_g(\delta^i) \quad \leq \quad \sum_i a_i \max_j V_g(\delta^j) \quad = \quad V_g^{\max}[\pi \triangleright C] \quad .$$

□

Thus it's sometimes desirable to analyze max-case posterior g-vulnerability.

Max-case posterior g-vulnerability

- We can establish that max-case posterior g-vulnerability does not exceed some threshold by calculating the expected-value−based posterior vulnerability for a modified gain function g_t.

Theorem (5.31)

Given $g\colon \mathbb{G}\mathcal{X}$, define g_t for $t\in\mathbb{R}$ as g with an extra action $\perp$ such that $g_t(\perp, x) = t$ for all x in $\mathcal{X}$. Then for all π and C we have

$$V_{g_t}[\pi \triangleright C] \;=\; t \qquad \textit{iff} \qquad V_g^{\max}[\pi \triangleright C] \;\leq\; t \quad .$$

Max-case posterior g-vulnerability

- **Proof.** As usual, let $[\pi \triangleright C] = \sum_i a_i [\delta^i]$.

  By construction of g_t, we have for every i that

  $$V_{g_t}(\delta^i) \quad = \quad \max\{V_g(\delta^i), t\} \quad .$$

  For the forward direction, suppose that $V_{g_t}[\pi \triangleright C] = \sum_i a_i V_{g_t}(\delta^i) = t$.

  Then, since $V_{g_t}(\delta^i) \geq t$ and the a_i's are convex coefficients, we must have $V_{g_t}(\delta^i) = t$ for all i.

  Hence we have that $V_g(\delta^i) \leq t$ for all i, which implies that $V_g^{\max}[\pi \triangleright C] \leq t$.

  For the backwards direction, suppose that $V_g^{\max}[\pi \triangleright C] = \max_i V_g(\delta^i) \leq t$.

  Then we must have $V_{g_t}(\delta^i) = t$ for all i, hence $V_{g_t}[\pi \triangleright C] = \sum_i a_i V_{g_t}(\delta^i) = t$.
  □

  Thus we see that, to some extent, expected-value−based posterior vulnerability analysis can subsume max-case posterior vulnerability analysis.

Appendix
More properties of posterior g-vulnerability and g-leakage: A linear-programming formulation

More properties of posterior g-vulnerability and g-leakage: A linear-programming formulation

- Seeing the elements of strategy S as variables, we realize that $V_g[\pi \triangleright \mathrm{C}]$ is the solution to the linear optimization problem (for fixed π and C and G):

> Choose S to maximize $\mathbf{tr}(\mathrm{G}\lceil\pi\rfloor\mathrm{CS})$,
> subject to all rows of S being probability distributions.

Note that:

- Here S is not necessarily deterministic, meaning that the choice of action can be made probabilistically. However, Theorem 5.24 says there's always an optimal solution with deterministic S.
- This linear-programming formulation is not particularly useful if we just wish to compute $V_g[\pi \triangleright \mathrm{C}]$, since we can directly evaluate the formula in Theorem 5.7.

  The value of those linear-programming insights is that we can adapt them in order to solve some challenging algorithmic problems, as we now demonstrate.

More properties of posterior g-vulnerability and g-leakage: A linear-programming formulation

- **Algorithm 5.27** Given
  - channel matrices A: $\mathcal{X} \to\!\triangleright \mathcal{Y}$ and B: $\mathcal{X} \to\!\triangleright \mathcal{Z}$, and
  - gain function g,

  decide whether there exists a prior π such that

$$V_g[\pi \triangleright \mathsf{A}] < V_g[\pi \triangleright \mathsf{B}]$$

  and, if so, find a prior π that maximizes the difference.

  (Notice that this is equivalent to finding a prior that maximizes the difference between B's additive g-leakage and A's.)

  The challenge here is that there are infinitely many priors π to consider.

More properties of posterior g-vulnerability and g-leakage: A linear-programming formulation

- However, we can formulate that as an optimization problem for fixed A, B, and G (the matrix representation of g); and if we view the elements of π, S^A, and S^B as variables, that becomes

> Choose π to maximize $\left(\max_{S^B} \ \mathbf{tr}(G^{\lceil}\pi^{\rfloor}BS^B) - \max_{S^A} \ \mathbf{tr}(G^{\lceil}\pi^{\rfloor}AS^A)\right)$
> subject to π and all rows of S^A, S^B being probability distributions.

There are however two issues with that formulation:

1. $\mathbf{tr}(G^{\lceil}\pi^{\rfloor}BS^B)$ is quadratic in the variables π and S^B, rather than linear, and
2. it contains nested maximizations.

More properties of posterior g-vulnerability and g-leakage: A linear-programming formulation

- But we can address those issues by:
  1. first observing that we can assume without loss of generality that the strategies are deterministic, and
  2. then noting that there are only finitely many deterministic strategies S^{A} and S^{B}, namely $|\mathcal{W}|^{|\mathcal{Y}|}$ and $|\mathcal{W}|^{|\mathcal{Z}|}$ many, respectively.
- Moreover, for a fixed S^{A} we can express the property

  "π is a prior such that S^{A} is optimal"

  via a set of linear constraints that say that, for every y, the expected gain from action $\mathsf{S}^{\mathsf{A}}(y)$ is at least as good as that from any action w.

  That is, we require for all y in $\mathcal{Y}$ and w in $\mathcal{W}$ that

$$\sum_x \pi_x \mathsf{A}_{x,y} g(\mathsf{S}^{\mathsf{A}}(y), x) \quad \geq \quad \sum_x \pi_x \mathsf{A}_{x,y} g(w, x) \quad .$$

More properties of posterior g-vulnerability and g-leakage: A linear-programming formulation

- If we denote those constraints as $opt(\mathsf{S}^{\mathsf{A}})$, then the solution to our non-linear optimization above will be the maximum of the solutions to the following linear problems, over all choices of S^{A} and S^{B}:

> Choose π to maximize $\Big(\mathbf{tr}(\mathsf{G}^{\lceil}\pi^{\rfloor}\mathsf{B}\mathsf{S}^{\mathsf{B}}) - \mathbf{tr}(\mathsf{G}^{\lceil}\pi^{\rfloor}\mathsf{A}\mathsf{S}^{\mathsf{A}})\Big)$
> subject to π being a probability distribution and $opt(\mathsf{S}^{\mathsf{A}})$.

Thus we are able to solve our original problem by solving a total of $|\mathcal{W}|^{|\mathcal{Y}|+|\mathcal{Z}|}$ linear-programming problems.

(Of course that will still be prohibitively expensive unless $\mathcal{W}$, $\mathcal{Y}$, and $\mathcal{Z}$ are small.) ◁

More properties of posterior g-vulnerability and g-leakage: A linear-programming formulation

- **Algorithm 5.28** Given
  - channel matrices A: $\mathcal{X} \rightarrow\!\triangleright \mathcal{Y}$ and B: $\mathcal{X} \rightarrow\!\triangleright \mathcal{Z}$, and
  - prior π,

  decide whether there exists a gain function g such that

$$V_g[\pi \triangleright \mathsf{A}] < V_g[\pi \triangleright \mathsf{B}]$$

  and, if so, find a gain function g that maximizes the difference.

- This problem is similar to that considered in the previous algorithm.

  The difference is that here π is fixed and the elements of G (the matrix representation of g) are variables.

More properties of posterior g-vulnerability and g-leakage: A linear-programming formulation

- One issue, however, is that maximizing over all of $\mathbb{G}\mathcal{X}$ will typically lead to an unbounded solution, since (by our gain-function algebra of Thm. 5.10) scaling g by k will also scale the difference $V_g[\pi \triangleright \mathsf{B}] - V_g[\pi \triangleright \mathsf{A}]$ by k.

  Hence, we instead will maximize over one-bounded gain functions $g\colon \mathbb{G}^{\updownarrow}\mathcal{X}$.

  To do this:

  1. We constrain the gain values of g to be at most 1 (but we do not bound them from below).
  2. And to ensure that g's vulnerabilities are non-negative, we include a special action $\perp$ whose gain values are all 0.

More properties of posterior g-vulnerability and g-leakage: A linear-programming formulation

- A second issue is that the number of possible actions of g is not fixed.

  However it is easy to see that it suffices to have $|\mathcal{Y}| + |\mathcal{Z}| + 1$ possible actions, since:

  - $V_g[\pi \triangleright \mathrm{A}]$ can use at most $|\mathcal{Y}|$ actions,
  - $V_g[\pi \triangleright \mathrm{B}]$ can use at most $|\mathcal{Z}|$, and
  - we also need the $\perp$ action.

More properties of posterior g-vulnerability and g-leakage: A linear-programming formulation

- At this point, we could proceed as in Algorithm 5.27 and try exponentially many strategies S^A and S^B.

  But it turns out that here we can do enormously better — it is sufficient to consider just one S^A and S^B !

  To see this, observe that we can assume without loss of generality that:

  - the first $|\mathcal{Y}|$ rows of G contain, in order, optimal actions for the columns of A;
  - the next $|\mathcal{Z}|$ rows of G contain, in order, optimal actions for the columns of B;
  - and finally the last row of G is all zero, for the $\perp$ action.

  Achieving this just requires reordering the rows of G and possibly duplicating some actions (since the same action might be optimal for more than one column).

  Once this is done, we can use a fixed S^A that maps column j of A to row j of G, and a fixed S^B that maps column k of B to row $k+|\mathcal{Y}|$ of G.

More properties of posterior g-vulnerability and g-leakage: A linear-programming formulation

- Now we can solve a single linear programming problem for that S^A and S^B:

> Choose G to maximize $\left(\mathbf{tr}(G^{\lceil}\pi^{\rfloor}BS^B) - \mathbf{tr}(G^{\lceil}\pi^{\rfloor}AS^A)\right)$
> subject to G having elements of at most 1 and a final all-zero row, and $opt\,(S^A)$.

Remarkably, and in sharp contrast to Algorithm 5.27, this means that a maximizing gain function G can be found in polynomial time. ◁