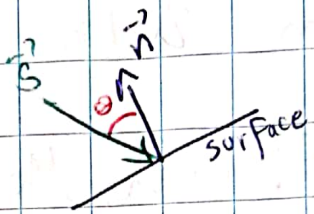
Exam 2018

1) Reflection Models

Lambertian reflection model

$$R = e\,\rho_R \cos\theta$$
$$G = e\,\rho_G \cos\theta$$
$$B = e\,\rho_B \cos\theta$$



a) The Lambertian model assumes a matte surface with only difuse reflection.
Glossy and mirror surfaces have specularities that the Lambertian model does not account for.

b) No, the reflected light is independent of the view point.
Imagine that the sun has to move just because you tilted your head to the left! ~~That~~ You are not the center of the universe! Grow up, get a real job and stop embarassing your family.
If you don't beleove me just look at the formula:

Where is the view point?

$$R = \underbrace{e}_{\text{intensity and color of the light source}} \cdot \underbrace{\rho_r}_{\text{red albedo component}} \cdot \underbrace{\underbrace{\vec{n}}_{\text{surface normal}} \cdot \underbrace{\vec{s}}_{\text{light source direction}}}_{=\cos(\theta)}$$

24/03/2019

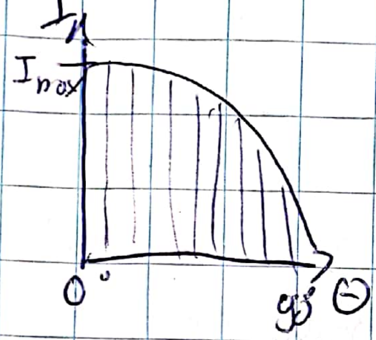
CVi

Victor Zhananezy

24/03/2019

c) According to the model $\cos(\Theta) = 1 \Rightarrow \Theta = 0°$. When the light source shines straight on the object.

Alternativelly, $\cos(\Theta) = 0 \Rightarrow \Theta = 90°$, which means that the light rays are parallel to the surface, do not touch it, thus no light is reflected. BLACK!



CV1

d) Intrinsic image decomposition factorizes the image into shading, ~~and~~ albedo and reflectance. As we have a Lambertian model, there are no reflectances (specularities).

Shading component: $\cos\Theta = \vec{n} \cdot \vec{s}$

Albedo component: ρ_R, ρ_G, ρ_B

Victor Ouanaz zi

Scanned by CamScanner

1e) We have a tridimensional world; so $\vec{s} = \begin{bmatrix} s_x \\ s_y \\ s_z \end{bmatrix}$, three variables to be calculated. By definition $\|\vec{s}\| = 1$

We also have from the lambertian model:

$$R = e \cdot \rho_r \cdot \vec{n} \cdot s = e \cdot \rho_r \cdot [n_x \ n_y \ n_z] \begin{bmatrix} s_x \\ s_y \\ s_z \end{bmatrix} \Rightarrow$$

$$s_x n_x + s_y n_y + s_z n_z = \frac{R}{e \rho_r}$$

We have to know the reflected red component R, the light source e and the albedo red component of the object. and the surface normal at that point.

If we know those components ~~for~~ of 2 points of the object, we have found 3 eq to solve for the three components of $\vec{s}$.

In math:

$$\|\vec{s}\| = 1$$

$$\vec{n}_1 \vec{s} = \frac{R_1}{e \rho_{r1}} \rightarrow \text{for one point}$$

$$\vec{n}_2 \vec{s} = \frac{R_2}{e \rho_{r2}} \rightarrow \text{for another point}$$

assumes parallel rays.

For all that to work out we assume that the light source is located in the infinity and such that all insisive light rays are parallel.

24/03/2019

CV1

Victor Grabarski

Scanned by CamScanner

24/03/2019

CVI

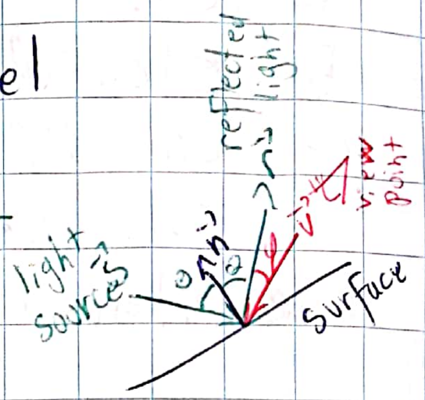
Victor Zuanazzi

1f) $\begin{bmatrix} R^e \\ G^e \\ B^e \end{bmatrix} = \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix} \begin{bmatrix} R^{\hat{e}} \\ G^{\hat{e}} \\ B^{\hat{e}} \end{bmatrix} \Rightarrow a = \frac{R^e}{R^{\hat{e}}}, b = \frac{G^e}{G^{\hat{e}}}, c = \frac{B^e}{B^{\hat{e}}}$

1g) Dichromatic reflection model

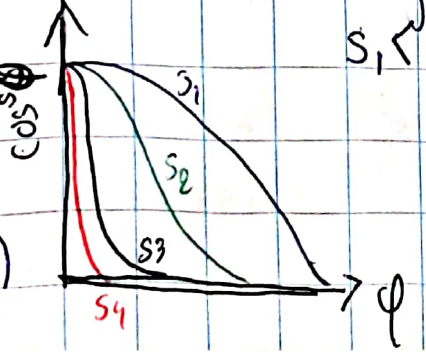
↳ that is not a desease!

$$R = \underbrace{e \rho_r \cos\theta}_{\text{matte component}} + \underbrace{e \cos^s \varphi}_{\text{glossy component}}$$
$$= e \rho_r \vec{n} \cdot \vec{s} + e(\vec{r} \cdot \vec{v})$$



$\cos^s \varphi$ models the specularity of a glossy surface. φ is the angle between the reflected light, $\vec{r}$ (as in a mirror) and the point of view of the observer, $\vec{v}$. The specularity experienced by the observer is the hieghest when its view point is on the direction of $\vec{r}$, in other words $\cos^s(0) = 1$. The specular component s is a proprierty of the object and defines how concentrated the specularity is. For a miror $s = 0$, the hole surface directly reflects light and for S large the specularity is seen as a small reagion of the object.

$s_1 < s_2 < S_3 < S_4$



1h)

Scanned by CamScanner

1i) The color of the highlight is dependent on the color of the light source.

Color of the highlight:

$R_n = \boxed{e} \cos^s \varphi$
$B_n = \boxed{e} \cos^s \varphi$
$G_n = \boxed{e} \cos^s \varphi$

↳ all tree reflected components depend on the intensity and color of the light source.

2) Photometric Invariants

$$\frac{2R - B - G}{B - G} = $$

$$= \frac{2e\rho_r \cos\theta + 2e\cos^s\varphi - e\rho_B \cos\theta - e\cos^s\varphi - e\rho_G \cos\theta - e\cos^s\varphi}{e\rho_B \cos\theta + e\cos^s\varphi - e\rho_G \cos\theta - e\cos^s\varphi}$$

$$= \frac{e\cos\theta(2\rho_r - \rho_B - \rho_G) + \underbrace{e\cos^s\varphi(2-1-1)}_{=0}}{e\cos\theta(\rho_B - \rho_G)\, \underbrace{e\cos^s\varphi(1-1)}_{=0}}$$

Blue pen died. 1 min of silence to pay respect.

$$= \frac{e\cos\theta(2\rho_r - \rho_B - \rho_G)}{e\cos\theta(\rho_B - \rho_G)}$$

that assumes that e is white light.

$$= \frac{2\rho_r - \rho_B - \rho_G}{\rho_B - \rho_G}$$

24/03/2019

CV 1

Victor Bouwens

Scanned by CamScanner

24/03/2019

Under the assumption that e is a white light source, at a pixel, $\frac{2R-B-G}{B-G}$ is ~~invar~~ a color invariant to shape (geometry), light source intensity and direction.

2b) Lambertian reflection model

$$\frac{R_{x_1} G_{x_2}}{R_{x_2} G_{x_1}} = \frac{\cancel{e_r} \rho_{r_1} \cancel{\vec{n}_1 \vec{s}} \cdot \cancel{e_G} \rho_{G_2} \cancel{\vec{n}_2 \vec{s}}}{\cancel{e_r} \rho_{r_2} \cancel{\vec{n}_2 \vec{s}} \cdot \cancel{e_G} \rho_{G_1} \cancel{\vec{n}_1 \vec{s}}}$$

$$= \frac{\rho_{r_1} \rho_{G_2}}{\rho_{r_2} \rho_{G_1}} \rightarrow \text{ops!}$$

This color invariant is independent of geometry and it is independent of light sourse intensity, direction and <u>color</u>.

CV1

2c) $\frac{2R-B-G}{B-G}$ assumes a white light source, therefore it is not appropriated to use under sirconstances in which the color of the light source changes.

$\frac{R_{x_1} G_{x_2}}{R_{x_2} G_{x_1}}$ is invariant to light source color, so that is the one we should use. (however, it assumes matte objects, the choice is also not appropriated if the objects have high specularity).

Victor Zuanazzi

24/03/2019

CVI

Victor Zvanoezi

2d) $R = 20,\ G = 40,\ B = 60,\quad \sigma = 4 = \sigma_{u_i} \cdot \sigma_w$

Color model $2R + 4B = q(R, B)$

$\cancel{\sigma_q = \Delta}\quad \frac{\partial q}{\partial R} = 2\ ,\ \frac{\partial q}{\partial B} = 4$

$$\sigma_q = \sqrt{\left(\frac{\partial q}{\partial R} \cdot \sigma\right)^2 + \left(\frac{\partial q}{\partial B} \cdot \sigma\right)^2}$$

$$= \sqrt{(2 \cdot 4)^2 + (4 \cdot 4)^2} = \sqrt{64 + 256} = 17{,}89$$

Uncertainty of the color model

The color model is stable because in no situation $\sigma_q \to \infty$.

2e) $q = \frac{R+B}{2G} \Rightarrow \frac{\partial q}{\partial R} = \frac{1}{2G},\ \frac{\partial q}{\partial B} = \frac{1}{2G},\ \frac{\partial q}{\partial G} = \frac{-1}{4G^2}$

$$\sigma_q = \sqrt{\left(\frac{\partial q}{\partial R}\sigma\right)^2 + \left(\frac{\partial q}{\partial B}\sigma\right)^2 + \left(\frac{\partial q}{\partial G}\sigma\right)^2}$$

$$= \sqrt{\frac{\sigma^2}{4G^2} + \frac{\sigma^2}{4G^2} + \frac{(R+B)^2\sigma^2}{16G^4}} = \frac{\sigma}{2G}\sqrt{2 + \frac{(R+B)^2}{4G^2}}$$

$$= \frac{4}{80}\sqrt{2 + \frac{(R+B)^2}{4 \cdot 40^2}} =$$

$$= \frac{4}{80}\sqrt{2 + \frac{80^2}{4 \cdot 40^2}} = 0{,}086$$

This model is not stable when the intensity of the green component aproaches zero, $\sigma_q \to \infty$

Scanned by CamScanner

24/03/2019

When $R = G = B$

$$\frac{\sigma}{2G}\sqrt{2+\frac{(R+B)^2}{4G^2}} = \frac{\sigma}{2G}\sqrt{2+\frac{\cancel{(G+G)^2}}{4\cancel{G^2}}} = = \frac{\sigma}{2G}\sqrt{\frac{9}{4}} = \frac{3\sigma}{4G}$$

if the intensity of the light aproaches zero

$$\sigma_\varphi \to \infty \Leftarrow \frac{3\sigma}{4 G\to 0}$$

3 Filters and Image Features

3a) assuming padding = 1 so the position (2, 2) is in the center of the convoluted patch:

CVI

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \underset{\text{convolution}}{*} \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \underset{\text{cross correlation}}{\otimes} \begin{bmatrix} 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{bmatrix}$$

$$= 9a + 8b + 7c + 6d + 5e + 4f + 3g + 2h + 1i$$

Victor Zvanozzi

3 b) $g = [-1 \;\; \underline{5} \;\; -1]$

← center of the filter

g does sharpening and scaling when not normalized.

The sharpening happens by intensifing the diferences between the center pixel and its neighbors.

Scaling happens because $-1+5-1 = 3 > 1$ the filter is not normalized. Thus the average value of the image is higher.

Zero padding

$$\underbrace{[1 \;\; 1 \;\; 1]}_{\text{image}} \underset{\text{convolution}}{*} \underbrace{[-1 \;\; 5 \;\; -1]}_{\text{filter}} = \underbrace{[3 \;\; 3 \;\; 3]}_{\text{resulting image}} \leftarrow \text{scaled}$$

$$[1 \;\; 0 \;\; 2] * [-1 \;\; 5 \;\; -1] = [5 \;\; -3 \;\; 10] \rightarrow \text{scaled and sharpened.}$$

$-1 \quad +2 \qquad\qquad -8 \quad +13$

If g is normilized then only ~~the~~ a mild sharpening is done because most of the weight is applied on the center pixel

$$[1 \;\; 0 \;\; 2] * \frac{1}{3}[-1 \;\; 5 \;\; -1] = \left[\frac{5}{3} \;\; -1 \;\; \frac{10}{3}\right] \quad \text{Mild sharpening}$$

$-8/3 \quad 13/3$

$-6 \quad 9$

$$[1 \;\; 0 \;\; 2] * [-1 \;\; 3 \;\; -1] = [3 \;\; -3 \;\; 6] \quad \text{Strong sharpening}$$

24/03/2019

CV I

Victor Buendía

24/03/2019

3c) h= [1 3 6 7 6 3 1] → it is also an approximation of gaussian filter.

This filter smooths the image.

When applied not normalized, it also scales the pixel values.

~~This~~ The name is lowpass filter because it removes high frequencies from the image.

3d) This filter can be obtained by convolving a box filter with itself 3 times

$[1\ 1\ 1] * [1\ 1\ 1] = [1\ 2\ 3\ 2\ 1]$

$[1\ 2\ 3\ 2\ 1] * [1\ 1\ 1] = [1\ 3\ 6\ 7\ 6\ 3\ 1]$

CV1

As oposite to a low pass filter, we can aply a high pass filter:

$[-1\ 0\ 1] * [-1\ 0\ 1] = [-1\ 0\ 2\ 0\ -1]$

$[-1\ 0\ 2\ 0\ -1] * [-1\ 0\ 1] = [1\ 0\ -3\ 0\ -3\ 0\ +1]$

Which is a derivative and only keeps the edges of the image.

Alternativelly, we can use a band pass filter tha filters out high and low frequencies, and keeps mid range frequencies.

$\underbrace{[-1\ 0\ 1]}_{\text{high pass}} * \underbrace{[1\ 1\ 1]}_{\text{low pass}} = \underbrace{[-1\ -1\ 0\ 1\ 1]}_{\text{band pass}}$

Victor Zuanazzi

Scanned by CamScanner

3e) $i = [-1\ 2\ -1]$

It is a second order derivative $[-1\ 1] * [1\ \ 1]$
$= [-1\ 2\ -1]$

It is useful for edge detection, but it is very sensitive to noise.

$[\underbrace{1\ 1\ 0\ 0}_{\text{edge}}\ \underset{\substack{\uparrow \\ \text{noise}}}{1}\ 0] * [1\ 2\ -1] = [\underset{\substack{\uparrow \\ \text{copy} \\ \text{padding}}}{0}\ 1\ -1\ \underbrace{-1\ 2\ -1}_{\text{answer of the noise is higher than the answer to the edge}}]$

3f) $i * h =$

$[-1\ 2\ -1] * [1\ 3\ 6\ 7\ 6\ 3\ 1]$

$= [-1\ -1\ -1\ -1\ 2\ -1\ -1\ -1\ -1]$

If the image is noisy, it may be a good idea to use $i * h$ because it smooths ~~and~~ the image and detect edges (the ones that survive that amout of smoothing!).

24/02/2019

CV

Victor Bueno

24/03/2019

3g) $P * F_x = [0]_{5\times5}$ Using copy padding.

$P * F_y = \begin{bmatrix} 0 \\ 0 \\ 2 \\ 0 \\ 0 \end{bmatrix}_{\times 5}$ $Q * F_x = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \end{bmatrix}$, $Q * F_y = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$

For the above, I am using coppy pading of apropriated size and $P * F_x = P \otimes [\underline{1} \ -1]$

3h) $M = \begin{bmatrix} \sum_i I_x^2 & \sum_i I_x I_y \\ \sum_i I_x I_y & \sum_i I_y^2 \end{bmatrix}$

$M_Q = \begin{bmatrix} 3 \cdot 2^2 & 2 \cdot 2 \\ 2 \cdot 2 & 3 \cdot 2^2 \end{bmatrix} = \begin{bmatrix} 12 & 4 \\ 4 & 12 \end{bmatrix}$ $M_P = \begin{bmatrix} 0 & 0 \\ 0 & 20 \end{bmatrix}$

3i) $\det(M - \lambda I) = 0$ to find eigen values

$(12 - \lambda)^2 - 16 = 0$

$\lambda^2 - 24\lambda + 128 = 0$

$\lambda = \dfrac{+24 \pm \sqrt{24^2 - 4 \cdot 128}}{2} = \begin{cases} \lambda_1 = 16 \\ \lambda_2 = 8 \end{cases}$

The eigen values are large, which meas it is likely there is a corner in the patch. If one of them was zero, then it is indicative of an edge.

Alternatively, we can check the Harris corner response:

$R = \det(M) - 0{,}04\, tr^2 M = 12^2 - 4^2 - 0{,}04 \cdot 24^2 = 104{,}9620$

indicative of corner.

CV

Victor Zwanzger

Scanned by CamScanner

Just for fun we can do the same for P

$\det(P - \lambda I) = 0$

$\det\begin{pmatrix} -\lambda & 0 \\ 0 & 20-\lambda \end{pmatrix} = -\lambda(20-\lambda) = 0 \Rightarrow \begin{cases} \lambda_1 = 20 \\ \lambda_2 = 0 \end{cases}$

Which is indicative of an edge because $\lambda_2 = 0$ and $\lambda_1 > 0$.

For the Harris corner response

$R = \det(P) - 0{,}04\, tr^2(P) = 0 - 0{,}04 \cdot 20^2 = \underbrace{-1{,}6 < 0}_{\text{indicative of not a corner.}}$

R indicates an edge.

3j) Eigen vectors are fun!

$M\vec{x} = \lambda \vec{x}$

$\begin{bmatrix} 12 & 4 \\ 4 & 12 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 16 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

$12x_1 + 4x_2 = 16x_1$

$4x_1 + 12x_2 = 16x_2$

$\div 4$

$3x_1 + 1x_2 = 4x_1$

$x_1 + 3x_2 = 4x_2$

$\cancel{\Rightarrow 8x_1 = 2 - 8x_1}$

$x_1 = +x_2$

$\vec{x}_1 = \frac{1}{\sqrt{2}}\begin{bmatrix} +1 \\ +1 \end{bmatrix}$

$\begin{bmatrix} 12 & 4 \\ 4 & 12 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 8 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

$12x_1 + 4x_2 = 8x_1$

$4x_1 + 12x_2 = 8x_2$

$3x_1 + x_2 = 2x_1$

$x_1 + 3x_2 = 2x_2$

$x_1 = -x_2$

$\vec{x}_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -1 \end{bmatrix}$

Sanety check, $\vec{x}_1$ and $\vec{x}_2$ must be orthogonal

24/07/2019

24/02/2019

Those eigen vectors show the fastest and slowest changes (respectively to the highest and lowest eigen values). In the case of a corner, $\vec{x}_1$ shows the direction of the corner.

4) Object Classification

4a) The multi-class NN has to classify I, L, O and X into one of the 4 classes: I L O and X!

What are the predictions for I and X?

$$I = [0\ 1\ 0\ 0\ 1\ 0\ 0\ 1\ 0]^T$$

$$y_I = \text{softmax}(M \cdot \vec{I}) = \text{softmax}\begin{bmatrix} 0.5 + 0.5 + 1 \\ 0.5 + 1 + 0.5 \\ 0.5 \\ 1 \end{bmatrix} = \text{softmax}\begin{bmatrix} 2 \\ 2 \\ 0.5 \\ 1 \end{bmatrix}$$

y_I is a tie between the classes I and L!

$$X = [1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1]$$

$$y_X = \text{softmax}(M\vec{X}) = \text{softmax}\begin{bmatrix} 0.5 \\ \cancel{0.5}\ 1 + 1 \\ 0 \\ 0.9 + 1 + 1 + 1 + 0.9 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 2 \\ 0 \\ \boxed{4.8} \end{bmatrix}$$

y_X is correctly classified as class X.

CV1

Victor Zúñiga [illegible]

Scanned by CamScanner

4 b) There are two main issues. Using fully connected layers requires the model to learn a realy large number of parameters. A small image of 100×100 leads to 10^4 params per neuron.

The second problem is that objects can be present in many places in different images. Fully connected layers imply in data that is not translation variant.

4 c) Image: $\underbrace{32}_{h} \times \underbrace{32}_{w} \times \underbrace{5}_{\text{chanels}}$

10 Filters ~~10 x 10 x~~ 5 $\quad \underbrace{5 \times 5}_{k} \times 5 \rightarrow$ Chanel global

$p=0$, $s=1$

$$dim_o = \frac{dim_i - k + 2p}{S} + 1$$

$$= \frac{32 - 5 + 0}{1} + 1 = 27 + 1 = 28$$

The output feature map is: $28 \times 28 \times \underbrace{10}_{\text{n° filters}}$

24/03/2019 almost 25

CV I

Victor [illegible]

Scanned by CamScanner

24/03/2019

4d) Standard CNN:
- convolutional layer
- activation function (ReLu)
- Max pool
- Normalization
- Fully connected layer

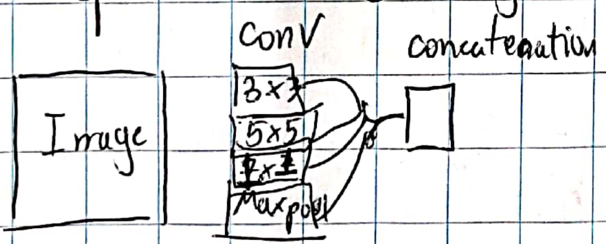
} creates a feature map

→ does something with the feature map.

Alex Net:
- convolution layer with different sizes and strides
- activation function (ReLu)
- Max Pooling
- Normalization
- Fully conected layer.

CVI

Inception (GoogLeNet)



4e) What is max pooling? It takes the max value in a evaluation window.

It is useful to reduce computational time as it decreases the size of the volume for the next convolutional layer.
The intuition behind it is that the max response is the most relevant one in that window.

Victor Zuanazzi

Scanned by CamScanner

4 f) It uses 3 steps:

1- Region (or window) proposal: an algorithm that returns bounding boxes that potentially contain a object.

2- Object detector: a classifier that outputs a score for how likely it is that there is an object (any object) inside a given bounding box

3- Multi-class classifier: A classifier that will classify the object in the bounding box.

4(g) Sliding Window	Selective Search
+ Easy to implement	- hard to implement
- Many evaluations are necessary: $w_s \cdot w_r \cdot p$	+ Requires less evaluations than Sliding window
- Most windows have no entire object in it	+ sorts and discards bounding boxes so only the ones that are likely to contain an entire object are returned.

24/03/2019 even classnto 25 CVI

Victor Bruno Zen

Scanned by CamScanner

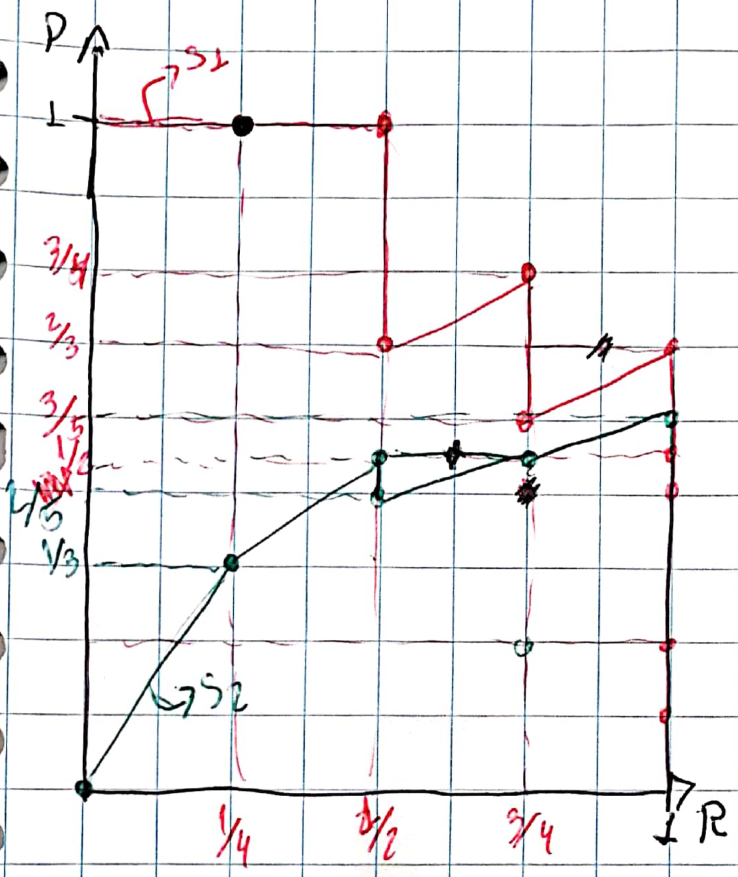
24/03/2019

1g)

For S_1

$P_{S_1} = 1, \frac{2}{2}, \frac{2}{3}, \frac{3}{4}, \frac{3}{5}, \frac{4}{6}, \frac{4}{7}, \frac{4}{8}, \frac{4}{9}, \frac{4}{10}$

$R_{S_1} = \frac{1}{4}, \frac{1}{2}, \frac{1}{2}, \frac{3}{4}, \frac{3}{4}, 1, 1, 1, 1, 1$



S_1 clearly outperforms S_2.

For S_2

$P_{S_2} = 0\ 0\ \frac{1}{3}, \frac{2}{4}, \frac{2}{5}, \frac{3}{6}, \frac{4}{7}, \frac{4}{8}, \frac{4}{9}, \frac{4}{10}$

$R_{S_2} = 0\ 0\ \frac{1}{4}, \frac{1}{2}, \frac{1}{2}, \frac{3}{4}, 1, 1, 1, 1$

$$AP_{S_2} = \frac{1}{4}\left(\frac{1}{3} + \frac{2}{4} + \frac{3}{6} + \frac{4}{7}\right) = 0{,}48$$

S_2 sucks!

$$AP_{S_1} = \frac{1}{4}\left(1 + 1 + \frac{3}{4} + \frac{4}{6}\right) = \frac{3{,}42}{4} = 0{,}85$$