Computer Vision Systems Programming VO

Approaching Computer Vision Problems

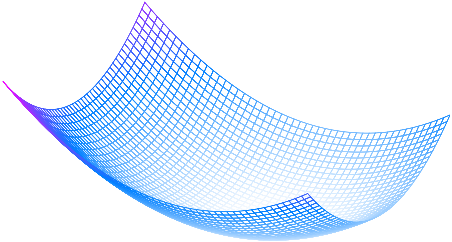
Christopher Pramerdorfer

Computer Vision Lab, Vienna University of Technology

Topics

Aspects of Computer Vision (CV) problems

Approaching CV problems



Aspects of CV Problems

CV is about

- Infering information about the world
- From images or videos (or extracted features)

To make things more formal, we represent

- This information as a scalar w or vector $\mathbf{w}$
- Our measurements as a feature vector $\mathbf{x}$

We then model the relationship between $\mathbf{x}$ and $\mathbf{w}$

- Usually the most challenging aspect

Aspects of CV Problems

So each CV problem consists of three things

- The information $\mathbf{w}$ we are interested in
- The features $\mathbf{x}$ from which we infer this information
- A model that describes the relationship

This model

- Is a mathematical function
- That usually has parameters that are learned from data

Approaching CV Problems

Thus we can break down every CV problem into

1. Specifying $\mathbf{w}$
2. Specifying how we obtain $\mathbf{x}$ (image processing)
3. Modeling the relationship between $\mathbf{x}$ and $\mathbf{w}$
4. Learning the particular relationship from training samples
5. Assessing the performance on test samples

We will discuss these steps using two example applications

Detect motion in videos via background subtraction

Image from Prince 2012

We want to predict whether a pixel belongs to a moving object

We are interested in a single property with two possible outcomes

- A pixel either belongs to a moving object or not
- So $\dim(\mathbf{w}) = 1$ and we just write w
- We let $w = 1$ mean yes and $w = 0$ mean no

As $w \in \{0, 1\} \subset \mathbb{N}$, this is a classification problem

Given our problem we

- Build a model independently for each pixel
- Let $\mathbf{x} = (r, g, b)$ be the corresponding pixel value

We decide that color should not matter and convert to grayscale

- We end up with a single value $x \in \{0, \dots, 255\}$

Suppose that it is reasonable to assume that

- The illumination remains constant
- The sensor noise is normally distributed
- We don't know how moving objects look like
- Moving and unmoving objects are equally likely

We build a statistical model $\mathrm{Pr}(x|w)$

- Probability that a pixel assumes value x, given a certain w

As we model x given w, this is a generative model

Given these assumptions, we have

$$
\begin{aligned}
Pr(x|w=0) &= \underset{x}{\mathrm{Norm}}(\mu, \sigma^2) \\
Pr(x|w=1) &= \underset{x}{\mathrm{Uniform}}(0, 255) \\
Pr(w) &= \underset{w}{\mathrm{Bern}}(0.5)
\end{aligned}
$$

We need to learn the model parameters $\boldsymbol{\theta} = (\mu, \sigma)$

So we collect n frames in which no motion occurs

- Let $\{x_i\}_{i=1}^{n}$ be the training set for a given pixel

We don't know anything about μ and σ so we

- Assume that the training samples are independent
- Estimate $\boldsymbol{\theta}$ purely from data (maximum likelihood)

Maximum likelihood means selecting the $\boldsymbol{\theta}$

- Under which observing $\{x_i\}_{i=1}^n$ is most likely

Or in mathematical terms

$$\begin{aligned}\hat{\boldsymbol{\theta}} &= \underset{\boldsymbol{\theta}}{\arg\max}\, \Pr(x_1, \ldots, x_n | \boldsymbol{\theta}) \\ &= \underset{\boldsymbol{\theta}}{\arg\max} \prod_{i=1}^{n} \Pr(x_i | \boldsymbol{\theta})\end{aligned}$$

Continuing from before

$$
\begin{aligned}
\hat{\boldsymbol{\theta}} &= \underset{\boldsymbol{\theta}}{\arg\max} \prod_{i=1}^{n} \underset{x_i}{\mathrm{Norm}}(\mu, \sigma^2) \\
&= \underset{\boldsymbol{\theta}}{\arg\max} \sum_{i=1}^{n} \log \underset{x_i}{\mathrm{Norm}}(\mu, \sigma^2) \\
&= \underset{\boldsymbol{\theta}}{\arg\max} \left(-0.5n \log \sigma^2 - 0.5n \log(2\pi) - 0.5 \sum_{i=1}^{n} \frac{(x_i - \mu)^2}{\sigma^2} \right)
\end{aligned}
$$

If we differentiate and equate to zero, we obtain

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n} (x_i - \hat{\mu})^2$$

We have derived algorithms for finding our model parameters

To obtain w from $\Pr(x|w)$ we use Bayes' rule

$$\Pr(w|x) = \frac{\Pr(x|w)\Pr(w)}{\Pr(x)}$$

Or if we don't need probabilities, we simply assign $w = 0$ iff

$$\Pr(x|w=0)\Pr(w=0) > \Pr(x|w=1)\Pr(w=1)$$

Assume we learned $\mu = 95, \sigma = 2$ and observe $x = 100$

We obtain

- $\Pr(x = 100 | w = 0) \Pr(w = 0) = \text{Norm}(100; 95, 2) \cdot 0.5$
- $\Pr(x = 100 | w = 1) \Pr(w = 1) = \text{Uniform}(0, 255) \cdot 0.5$
- We get 0.004 and 0.002, respectively, so $w = 0$

For probabilities we divide by $\Pr(x = 100)$ (Bayes rule)

We obtain $\Pr(x = 100)$ via marginalization

Recall from statistics lecture that

- $\Pr(x) = \sum_w \Pr(x, w)$ (discrete case)
- $\Pr(x, w) = \Pr(x|w)\Pr(w)$

So $\Pr(x = 100) = \sum_{i \in \{0,1\}} \Pr(x = 100|w = i)\Pr(w = i) \approx .006$

- We already calculated the summands in previous slide

Approaching CV Problems
Application 1 – Remarks

Model selection governed by domain knowledge

- Illumination does not change
- Sensor noise is normally distributed
- No information on moving object frequency, appearance

If you have this information, this is the way to go

- Think about how your data came into being
- Use this information for modeling

On this basis your solution *will work* unless

- The assumptions were wrong (e.g. illumination changes)
- Training data are not representative ($\hat{\theta}$ wrong)

Approaching CV Problems
Application 2

Categorize naturalistic images into thousands of classes

Image from `image-net.org`

Assume there is a single dominant object in each image

Goal is to predict which object is visible

Classification problem with c classes, $w \in \{0, \ldots, c-1\}$

What are good features for this task?

- No clear relationship between images and w
- Thus unclear how to select $\mathbf{x}$

So we just resize the images to a fixed size $u \times v$

And use a powerful model that figures out the rest

We use a Convolutional Neural Network (CNN)

- Special kind of neural network (details later)
- Known to perform exceptionally well in such cases

CNNs are discriminative models, $\Pr(w|\mathbf{x})$

- Compare to the generative model in application 1

CNNs consist of several layers we must specify manually

- Such model parameters are called hyperparameters
- More familiar example: k in k-means

Typically specified

- Based on experience, literature
- Experimentally (try and see what works best)

Assume that we again have n training samples $\{(\mathbf{x}_i, w_i)\}_{i=1}^n$

- w_i is class of visible object in image $\mathbf{x}_i$

Proceeding as in application 1 we seek

$$\begin{aligned}
\hat{\boldsymbol{\theta}} &= \operatorname*{arg\,max}_{\boldsymbol{\theta}} \Pr(w_1, \ldots, w_n | \mathbf{x}_1, \ldots, \mathbf{x}_n, \boldsymbol{\theta}) \\
&= \operatorname*{arg\,max}_{\boldsymbol{\theta}} \prod_{i=1}^n \Pr(w_i | \mathbf{x}_i, \boldsymbol{\theta}) \\
&= \operatorname*{arg\,max}_{\boldsymbol{\theta}} \sum_{i=1}^n \log \Pr(w_i | \mathbf{x}_i, \boldsymbol{\theta})
\end{aligned}$$

But our model $\mathrm{Pr}(w|\mathbf{x}, \boldsymbol{\theta})$ is very complex

- CNNs can have millions of parameters
- Computing $\mathrm{Pr}(w|\mathbf{x}, \boldsymbol{\theta})$ can require millions of operations
- There is no analytical solution

Thankfully algorithms for learning already exist

Our model is discriminative so we don't need Bayes' rule

- We have modeled $\Pr(w|\mathbf{x})$ directly

No obvious relationship between images and w

- Unclear how to select $\mathbf{x}$
- Unable to model the relationship like before

For these reasons we

- Use a powerful generic machine learning technique as model
- Incorporate feature selection into it

Approaching CV Problems
Application 2 – Remarks

CNNs are state of the art models for many CV problems

- But very complex, require lots of training data
- More on CNNs later

Approaching CV Problems

Both applications were extreme (but realistic) examples

Application 1

- Selecting and computing $\mathbf{x}$ was trivial
- Derived statistical model and algorithms

Application 2

- Unclear how to select $\mathbf{x}$
- Used a powerful model that did all the work

Approaching CV Problems

Most CV problems lie somewhere in between

- Focus on obtaining suitable $\mathbf{x}$ (image processing)
- Utilize generic machine learning algorithms as models

We will see a few examples in upcoming lectures

Approaching CV Problems

Suggestions

Think about how your data came into being

Use this information to model the relationship between $\mathbf{x}$ and $\mathbf{w}$

Ideally use probabilistic models (model uncertainty)

Don't think in terms of algorithms

"I will use a linear SVM to predict the class"

- Is this even a suitable model for the given problem?

Go the other way around

- Confirm that a linear *model* is applicable
- Select a suitable *algorithm* (e.g. linear SVM)

Approaching CV Problems
Suggestions

Embrace machine learning

- It is everywhere in modern CV (including image processing)
- Don't guess your parameters, learn them

Approaching CV Problems
Suggestions

Don't reinvent the wheel

- Others likely have worked on similar problems

Use online libraries

- IEEE (`ieeexplore.ieee.org`), ACM (`dl.acm.org`), Google Scholar

Use (good) libraries instead of starting from scratch

- We have already seen a selection

Bibliography

Prince, S.J.D. (2012). *Computer Vision: Models Learning and Inference*. Cambridge University Press.