Lecture 1.

Note Title 1/16/2010

Introduction to Vision.

- Vision is the task of interpreting the world from the rays of light that reach our eyes (or a camera).
- This is an inverse problem – the forward problem is computer graphics (i.e. to form an image by specifying the "world" and the lighting conditions).
- Vision is extremely difficult – Humans are vision machines, between 30-50% of the human cortex is devoted to doing vision.
- The difficulty of vision was first realized when researchers started trying to make computer vision systems.
- Why is vision so hard? Because images are big, complex, and ambiguous.
- The total number of possible images is huge $(1024 \times 1024)^{256}$ (256 intensity levels, per pixel; 1024×1024 pixels in an image)
- By constrast, statisticians usually work on data which lies in less than 10 dimensions.
- Vision is only possible because the set of images that occur in reality is a lot smaller. The structure of the world puts constraints of the types of images that occur – natural constraints (Marr), ecological constraints (Gibson)

Image I, State of the World W

E.G. W labels positions and properties of all objects in a visual scene.

Probability distribution

$P(I \mid W)$	generative model.
$P(W \mid I)$	discriminative model.
$P(W)$	world prior
$P(I)$	image prior

These are related by the identity (joint distribution.)

$$P(I \mid W) P(W) = P(W \mid I) P(I) = P(W, I)$$

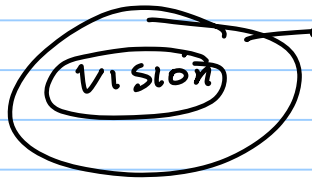
which gives Bayes Theorem:

$$P(W \mid I) = \frac{P(I \mid W) P(W)}{P(I)}$$

(2) The probability distributions $P(I|W), P(W|I), P(W), P(I)$ are defined over structured representations.
→ grammars, graphs,

Key Elements of the course.
(1) representations.
(2) inference. (eg. estimate W from $P(W|I)$)
(3) learning. using data to learn the distribution

Bigger picture: vision is part of larger enterprises



← models of intelligence,
~ includes natural language, reasoning, etc.

← data
~ includes statistics, machine learning

This is interdisciplinary research involving Computer Science (representations & algorithms) Statistics (probabilities, data analysis), Mathematics, and Engineering.

Probability distributions over structured representation (including learning and inference) offers the most promising approach to all these enterprises.

There are trade-offs between generative and discriminative methods which will keep arising during this course.

Discriminative: formulate vision by specifying / learning $P(W|I)$.

But the dimension and complexity of W makes it unrealistic to simple apply standard machine learning methods. It is important to introduce structure and take the complexity of inference and learning into account.

(3) Generative (Bayesian)

Specify a generative model $P(I|W)$ $P(W)$

use Bayes theorem to form

$$P(W|I) = \frac{P(I|W)P(W)}{P(I)} \quad \text{with } P(I) = \sum_W P(I|W)P(W)$$

This is conceptually attractive, but also extremely complicated. It requires developing probability models capable of generating real images (i.e. solving computer graphics). This strategy is called "analysis by synthesis."

The roles of $P(I|W)$ & $P(W)$. (Sinha's example)

Image: I 

$P(I|W)$ rules out interpretations W that are inconsistent with the image

→ e.g. W cannot be a sphere ○ or a cone △

But ambiguities remain, there are many three-dimensional objects consistent with the image

E.G. distorted shapes like

are consistent with the image from certain viewpoints

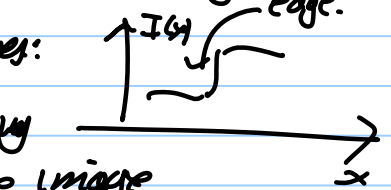
The prior $P(W)$ resolves the ambiguity by requiring that a cube is the most likely object W.

Claim is that there are more cubes that these other types of objects

(4) Example: Edge Detection.

Edges correspond to the boundaries of objects or significant changes in texture.

They typically occur at places in the image where there are large intensity changes:



These can be measured by applying filter F (linear or non-linear) to the image

E.g. derivative operator $|\nabla I|$ ∇-gradient

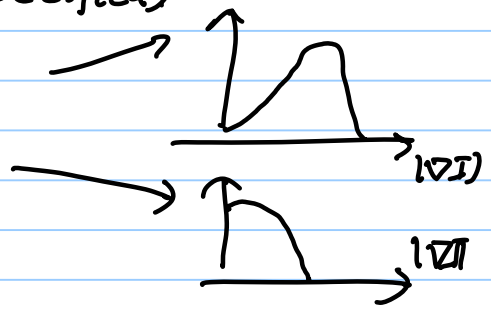
$(F*I)_{\underline{x}}$ is the value of the filter when evaluated at pixel $\underline{x}$.

Define a label $w(\underline{x}) \in \{0,1\}$ for each pixel. (non-edge, edge)

Let $p(w(\underline{x}) | (F*I)_{\underline{x}})$ be the probability distribution of the labels — this can be learnt from training data (i.e. images with the ground truth values of the edges specified)

$p(|\nabla I|(\underline{x}) \mid w(\underline{x})=1)$

$p(|\nabla I|(\underline{x}) \mid w(\underline{x})=0)$



Use a prior $p(w(\underline{x}))$ to determine $p(w(\underline{x}) | (F*I)_{\underline{x}})$

Alternatively, apply AdaBoost (or other machine learning) to learn $P(w(\underline{x}) | (F*I)_{\underline{x}})$.

Factorizability: assume conditional independence

$$P(w|I) = \prod_{\underline{x}} P(w(\underline{x}) | (F*I)_{\underline{x}})$$

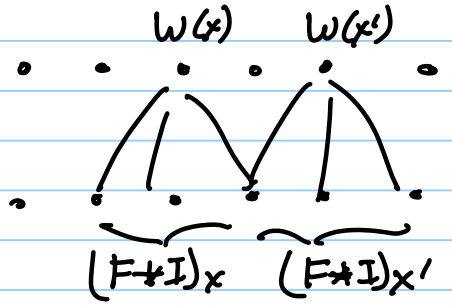
this makes inference easy

$$\hat{w} = \text{ARG MAX}_{w} P(w|I)$$

is given by $\hat{w} = \{\hat{w}(\underline{x})\}$, where $\hat{w}(\underline{x}) = \text{ARG MAX}_{w(\underline{x})} P(w(\underline{x}) | (F*I)_{\underline{x}})$

But this ignores the fact that neighbouring pixels may be more likely to be edges.

(5) Graph Structure.



No sideways relations between W(x) & W(x')

i.e. the model ignores the spatial context

Historical Context

→ edge detection was first formulated in this way by Konishi & Yuille (~1999).

→ why so late in history of computer vision? impossible to learn these distributions until we had image datasets with edges marked as ground truth (ie to learn $P(F*I_x|W(x))$.

→ later, the Berkeley dataset (Malik etal) define new filters (F*I) and learn by AdaBoost

Note: this is a discriminative model. It is very hard to model this problem by a generative model.

Key Idea:

- Representation → graph structure
- Learning → probs $P(F*I|W)$
  supervised learning (ground truth)
  note → learning must be checked by cross-validation to prevent over-generalization.
- Inference — very simple in this case

(6) Extension to image labeling

multiple classes

Label each image pixel as $w(x) \in \{$ edge, sky, road, vegetation, building, $\}$

As before, assume independence.

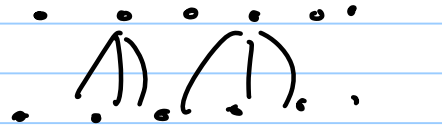
$$P(w|I) = \prod_x P(w(x)|(F*I)_x)$$

where, $P(w(x)|(F*I)_x)$ is a probability distribution for multiple classes

This can also be formulated by learning the distribution $P((F*I)_x|w(x))$ from training data and learning $P(w(x))$.

or by doing discriminative learning to get $P(w(x)|(F*I)_x)$ directly. (eg multiple-class AdaBoost)

(•) Representation →



(•) Learning → learn $P(F*I|w)$ or $P(w|F*I)$

(•) Inference → easy $\hat{w}(x) = \underset{w(x)}{\text{ARG MAX}}\ P(w(x)|(F*I)_x)$

For both problems:

where to get the filters F?

- select from dictionary of filters
- hand-design
- learn automatically

Why factorization:

It greatly simplifies the inference algorithm and the learning algorithm.

How to generalize to new datasets with different statistics — meta-model (later in course).