Graphical Models in Computer Vision

Andreas Geiger

Max Planck Institute for Intelligent Systems
Perceiving Systems

April 11, 2016

MAX-PLANCK-GESELLSCHAFT

Organization

Team

Javier Romero

Andreas Geiger

Gerard Pons-Moll

Joel Janai

Naureen Mahmood

Organization

- Lecture: 2 hours/week
  - Mon: 12:15 – 14:00, Room A301
- Exercises: 2 hours/week
  - Mon: 14:00 – 16:00, Room A301
- Exception
  - 25.4.: Kleiner Hörsaal, Sand 6/7
- Course web page: `http://cv.is.tue.mpg.de/`
  - Slides
  - Pointers to Books and Papers
  - Homework assignments
- Mailing list `http://groups.google.com/d/forum/cv-is`
  - Please register!

Exercises & Exam

- Credits: 4 LP (2+2)
- Exercises:
  - Goal: Understand theory and transfer into computer experiments
  - Work in teams of up to two
  - Pen and paper exercises
  - Computing exercises
    - Will use Linux & Python
    - We provide a VirtualBox (webpage)
    - Would a brief Python tutorial be useful?
- Exam
  - Oral exam
  - English or german
  - 50 % of exercise points required!
  - Examination dates: 21.7.2016 + 25.7.2016

Joel Janai

Naureen Mahmood

Questions on organizational part?

Topics & Materials

Graphical Models...

- Models
- Inference
- Learning

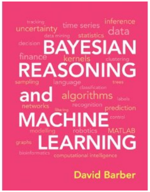
... in Computer Vision

- Image Denoising
- Human Pose Estimation
- Human Body Models
- Stereo
- Optical Flow
- Image Segmentation
- Object Detection

Syllabus

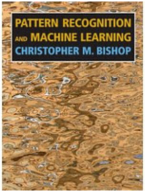
11.04.2016	Introduction
18.04.2016	Graphical Models 1
25.04.2016	Graphical Models 2 (Sand 6/7)
02.05.2016	Graphical Models 3
09.05.2016	Graphical Models 4
23.05.2016	Body Models 1
30.05.2016	Body Models 2
06.06.2016	Body Models 3
13.06.2016	Body Models 4
20.06.2016	Stereo
27.06.2016	Optical Flow
04.07.2016	Segmentation
11.07.2016	Object Detection 1
18.07.2016	Object Detection 2

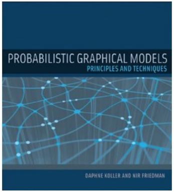
Main Book for Graphical Model Part

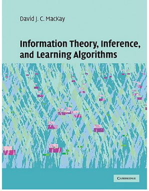


- Barber, **Bayesian Reasoning and Machine Learning**, Cambridge University Press, 2011, ISBN-13: 978-0521518147, `http://tinyurl.com/3flppuo`
- Available online **for free**
- Comes with graphical model toolbox (for Matlab)

For the curious ones ...



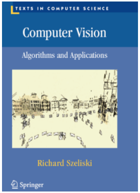


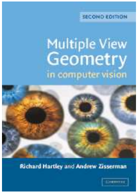


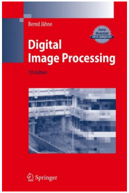
- Bishop, **Pattern Recognition and Machine Learning**, Springer New York, 2006, ISBN-13: 978-0387310732
- Koller, Friedman, **Probabilistic Graphical Models: Principles and Techniques**, The MIT Press, 2009, ISBN-13: 978-0262013192
- MacKay, **Information Theory, Inference and Learning Algorithms**, Cambridge University Press, 2003, ISBN-13: 978-0521642989

Links are available on the course website.

Computer Vision References



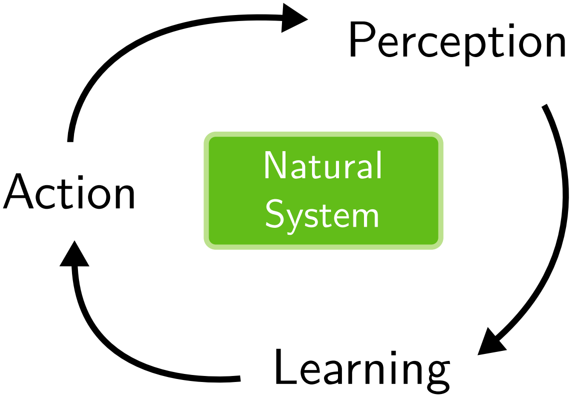


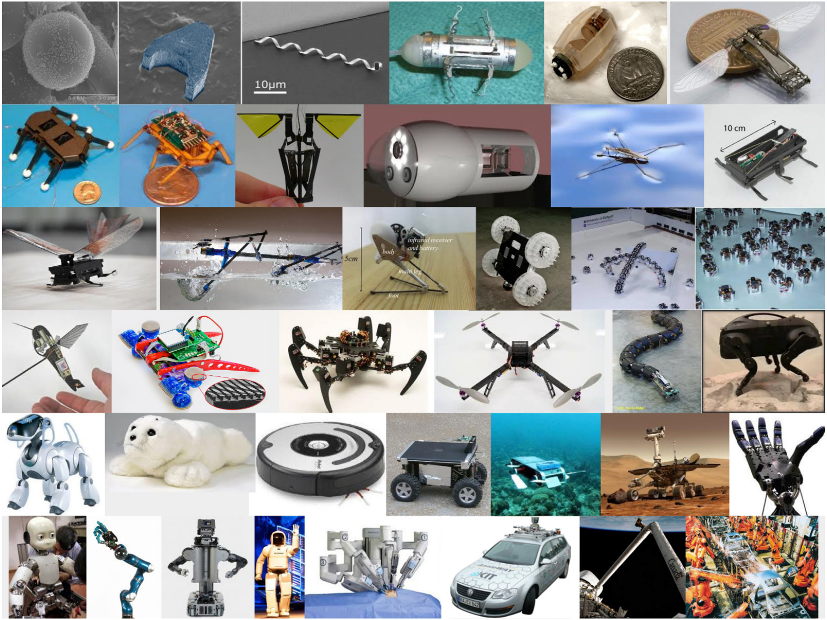


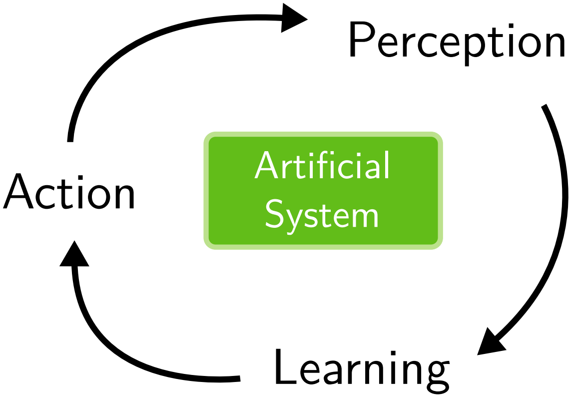
- Szeliski, **Computer Vision: Algorithms and Applications**
- Hartley & Zisserman, **Multiple View Geometry in Computer Vision**
- Bernd Jähne, **Digital Image Processing and Image Formation**

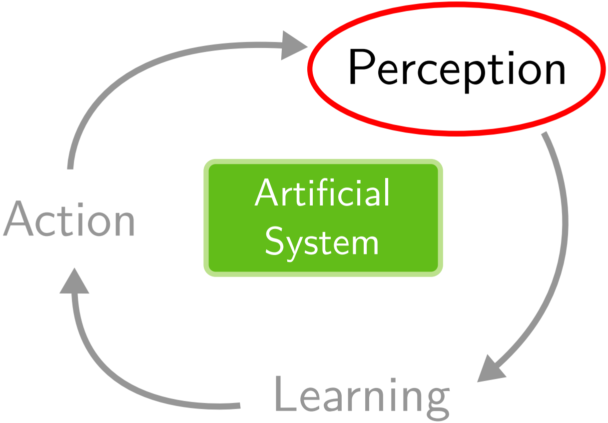
Links are available on the course website.

Introduction


Perception
Natural System
Action
Learning

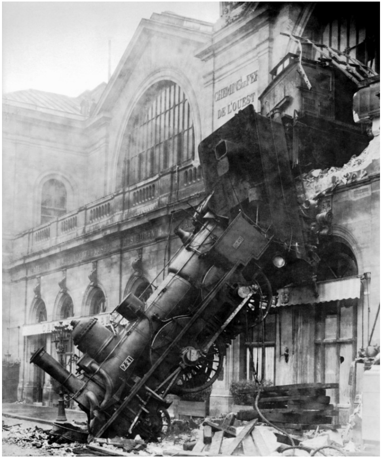

10μm
10 cm
5cm
body
infrared receiver and battery
main leg
foot


Perception
Artificial System
Action
Learning


Perception
Artificial System
Action
Learning

Why is Visual Perception hard?

Why is Visual Perception hard?



What we see

200	133	110	103	117	90	47	30	32	79	66	65
197	122	123	138	98	100	46	45	22	11	43	55
140	116	165	159	90	56	58	47	26	13	54	102
132	148	119	108	123	57	64	46	21	22	79	94
125	121	80	143	101	55	61	38	20	21	81	65
50	71	74	63	52	39	41	39	32	26	97	66
51	59	62	44	40	40	36	28	27	31	29	44
59	62	70	50	48	35	34	35	26	21	24	32
49	59	65	64	58	34	40	28	26	21	23	124
39	45	47	64	54	34	40	24	19	47	133	207
37	42	39	38	39	50	75	74	105	170	197	167
37	47	33	35	50	108	162	184	184	157	125	112
45	48	35	37	75	148	183	156	83	91	91	116
49	48	54	50	75	158	110	66	74	128	155	149
48	51	57	50	65	91	79	92	101	105	132	132
51	58	66	55	58	52	91	91	88	115	158	174
57	60	61	52	56	61	60	55	92	146	188	190
65	50	54	56	57	51	54	56	80	115	177	187
67	40	40	61	65	48	39	30	36	75	151	181
53	32	36	35	61	43	37	26	29	35	126	189
29	42	107	20	28	41	40	26	30	36	113	200
30	21	32	24	34	37	33	23	25	39	105	171
32	28	19	23	29	36	47	89	132	169	183	128
31	25	62	54	47	44	81	190	227	231	206	155
44	66	99	72	67	63	89	128	127	115	109	157
53	47	47	41	29	32	25	20	41	81	89	175
38	44	61	73	54	48	37	87	90	111	126	189
39	41	83	97	86	91	74	134	131	153	143	185
42	56	98	102	112	111	94	137	121	141	146	181
94	114	114	114	122	113	77	117	117	154	149	169
157	176	116	121	130	139	103	161	148	180	145	125
143	178	182	178	139	153	129	168	175	187	170	152
127	183	203	197	153	164	143	180	195	182	165	211
88	107	127	125	101	107	100	123	149	186	167	215

What the computer sees

Why is Computer Vision hard?

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

PROJECT MAC

Artificial Intelligence Group July 7, 1966

Vision Memo. No. 100.

<u>THE SUMMER VISION PROJECT</u>

Seymour Papert

The summer vision project is an attempt to use our summer workers effectively in the construction of a significant part of a visual system. The particular task was chosen partly because it can be segmented into sub-problems which will allow individuals to work independently and yet participate in the construction of a system complex enough to be a real landmark in the development of "pattern recognition".

Why is Visual Perception hard?

Slide credits: Antonio Torralba

Why is Visual Perception hard?

Slide credits: Antonio Torralba

Why is Visual Perception hard?

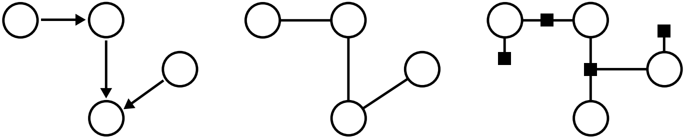
Slide credits: Antonio Torralba

Why is Visual Perception hard?

Why is Visual Perception hard?

Intelligent Systems require Robust Vision

- Feature invariance
- Good prior
- Tractable representations
- Efficient learning and inference
- Model uncertainty



Some Examples from our Lab

https://ps.is.tuebingen.mpg.de/research

Probability Theory Review

Brief Review

- A random variable X can take values from some discrete set of outcomes $\mathcal{X}$ (think six-sided dice)
- We usually use the short-hand notation

$$p(x) \text{ for } p(X = x) \in [0, 1]$$

  for the probability *that* X *takes value* x
- With

$$p(X)$$

  we denote the *probability distribution* over X
- $p(x)$ must satisfy the following conditions:

$$\begin{aligned} p(x) &\geq 0 \\ \sum_{x \in \mathcal{X}} p(x) &= 1 \end{aligned}$$

Brief Review

- Joint probability (of X and Y)

$$p(x, y) \text{ instead } p(X = x, Y = y)$$

- Conditional probability

$$p(x|y) \text{ instead } p(X = x|Y = y)$$

- Two RVs are called independent if

$$p(X = x, Y = y) = p(X = x)p(Y = y)$$

Vocabulary

- Joint Probability

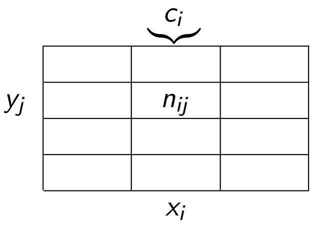
$$p(x_i, y_j) = \frac{n_{ij}}{N}$$

- Marginal Probability

$$p(x_i) = \frac{c_i}{N}$$

- Conditional Probability

$$p(y_j \mid x_i) = \frac{n_{ij}}{c_i}$$



$$c_i = \sum_j n_{ij}$$

$$N = \sum_{ij} n_{ij}$$

The Rules of Probability

- Sum rule
$$p(X) = \sum_{y \in \mathcal{Y}} p(X, Y = y)$$
we "marginalize out y". $p(X = x)$ is also called a marginal probability
- Product Rule
$$p(X, Y) = p(Y|X)p(X)$$
- And as a consequence: Bayes Theorem
$$p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)}$$

Probability Densities

- Now X is a continuous random variable, eg taking values in $\mathbb{R}$
- Probability that X takes a value in the interval (a, b) is

$$p(X \in (a, b)) = \int_a^b p(x)\,\mathrm{d}x$$

and we call $p(x)$ the probability density over x

Probability Densities

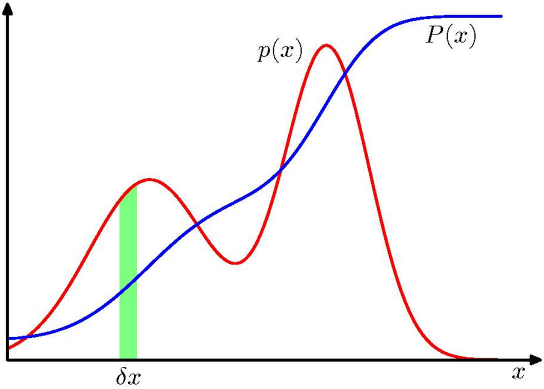
- $p(x)$ must satisfy the following conditions

$$
\begin{aligned}
p(x) &\geq 0 \\
\int_{-\infty}^{\infty} p(x) &= 1
\end{aligned}
$$

- The probability that x lies in $(-\infty, z)$ is given by the cumulative distribution function

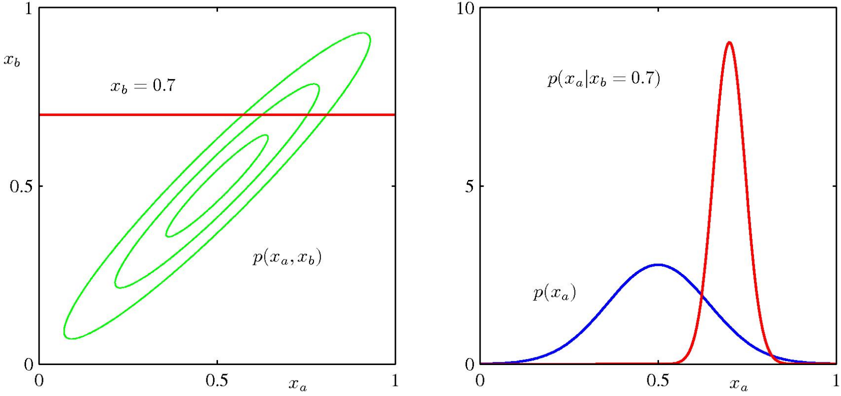
$$P(z) = \int_{-\infty}^{z} p(x) \mathrm{d}x$$

Probability Densities



Probability density of a continuous variable

Illustration



joint, marginal, conditional probability

Expectation and Variances

- Expectation

$$\mathbb{E}[f] = \sum_{x \in \mathcal{X}} p(x) f(x)$$
$$\mathbb{E}[f] = \int_{x \in \mathcal{X}} p(x) f(x) \, \mathrm{d}x$$

- Sometimes we denote the distribution that we take the expectation over as a subscript, eg

$$\mathbb{E}_p[f] = \sum_{x \in \mathcal{X}} p(x) f(x)$$

- Variance

$$\mathrm{var}[f] = \mathbb{E}\left[(f(x) - \mathbb{E}[f(x)])^2\right]$$

Structured Prediction

Standard Regression:

$$f : \mathcal{X} \rightarrow \mathbb{R}$$

- inputs $\mathcal{X}$ can be any kind of objects
    - images, text, audio, sequence of amino acids, ...
- output y is a real number
    - classification, regression, density estimation, ...

Structured Output Learning:

$$f : \mathcal{X} \rightarrow \mathcal{Y}$$

- inputs $\mathcal{X}$ can be any kind of objects
- outputs $y \in \mathcal{Y}$ are complex (structured) objects
    - images, parse trees, folds of a protein, ...

What is structured output prediction?

Ad hoc definition: predicting *structured* outputs from input data
(in contrast to predicting just a single number, like in classification or regression)

- Natural Language Processing:
  - Automatic Translation (output: sentences)
  - Sentence Parsing (output: parse trees)
- Bioinformatics:
  - Secondary Structure Prediction (output: bipartite graphs)
  - Enzyme Function Prediction (output: path in a tree)
- Speech Processing:
  - Automatic Transcription (output: sentences)
  - Text-to-Speech (output: audio signal)
- Robotics:
  - Planning (output: sequence of actions)

Graphical Models...

- Models
- Inference
- Learning

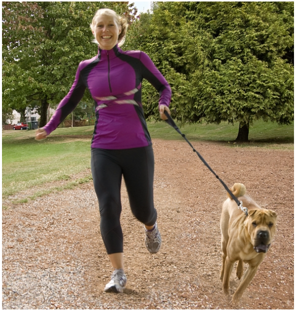
... in Computer Vision

- Object Detection
- Human Pose Estimation
- Optical Flow
- Stereo
- Image Denoising
- Segmentation
- Semantic Segmentation
- Image Stitching
- Tracking

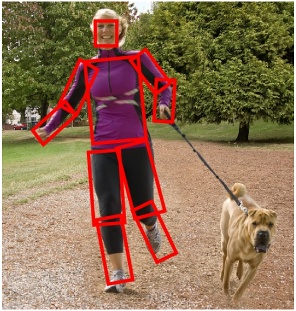
This is the language ...

... for these problems.

Example: Human Pose Estimation



$x \in \mathcal{X}$



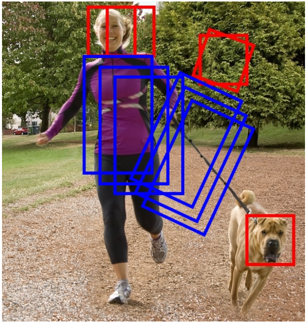
$y \in \mathcal{Y}$

- Given an image, where is a person and how is it articulated?

$$f : \mathcal{X} \to \mathcal{Y}$$

- Image x, but what is human pose $y \in \mathcal{Y}$ precisely?

Human Pose $\mathcal{Y}$



Example y_{head}

- **Body Part:** $y_{head} = (u, v, \theta)$ where (u, v) center, θ rotation
  - $(u, v) \in \{1, \ldots, M\} \times \{1, \ldots, N\}, \theta \in \{0, 45^\circ, 90^\circ, \ldots\}$
- **Entire Body:** $y = (y_{head}, y_{torso}, y_{left-lower-arm}, \ldots\} \in \mathcal{Y}$

Human Pose $\mathcal{Y}$

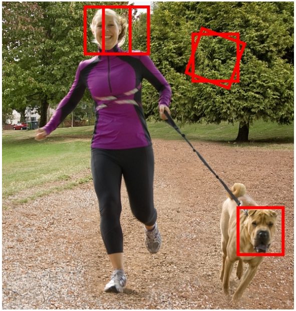
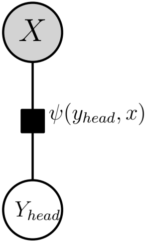


Image $x \in \mathcal{X}$

Example y_{head}



Head detector

- **Idea:** Have a head classifier (SVM, Random Forest, NN, ...)

$$\psi(y_{head}, x) \in \mathbb{R}_+$$

- Evaluate everywhere and record score
- Repeat for all body parts

Human Pose Estimation

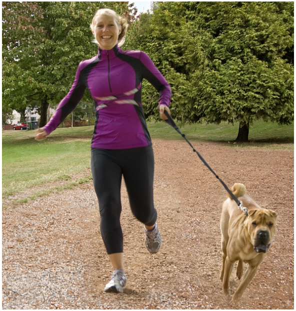
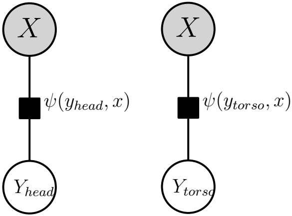


Image $x \in \mathcal{X}$



Prediction $y^* \in \mathcal{Y}$

- Compute

$$
\begin{aligned}
y^* &= (y^*_{head}, y^*_{torso}, \cdots) = \operatorname*{argmax}_{y_{head}, y_{torso}, \cdots} \psi(y_{head}, x)\, \psi(y_{torso}, x) \cdots \\
&= (\operatorname*{argmax}_{y_{head}} \psi(y_{head}, x), \operatorname*{argmax}_{y_{torso}} \psi(y_{torso}, x), \cdots)
\end{aligned}
$$

- Great! Problem solved!?

Human Pose Estimation

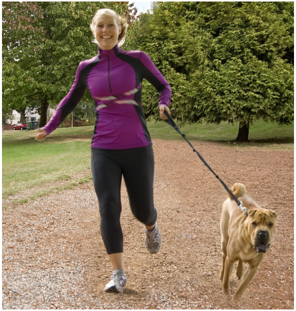
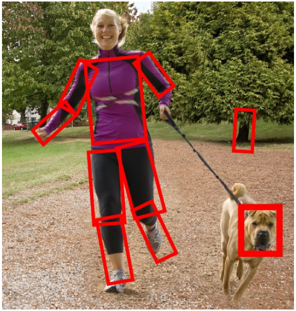


Image $x \in \mathcal{X}$



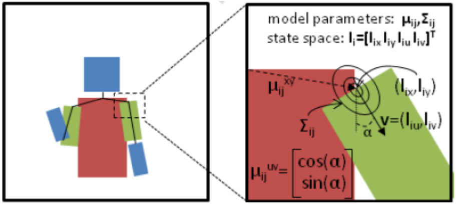
Prediction $y^* \in \mathcal{Y}$

- Compute

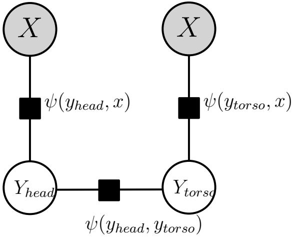
$$
\begin{aligned}
y^* &= (y^*_{head}, y^*_{torso}, \cdots) = \operatorname*{argmax}_{y_{head}, y_{torso}, \cdots} \psi(y_{head}, x)\, \psi(y_{torso}, x) \cdots \\
&= (\operatorname*{argmax}_{y_{head}} \psi(y_{head}, x), \operatorname*{argmax}_{y_{torso}} \psi(y_{torso}, x), \cdots)
\end{aligned}
$$

- Great! Problem solved!?

Idea: Connect Body Parts



$\psi(y_{torso}, y_{arm})$



Head-Torso Model

- Ensure *head* is on top of *torso*

$$\psi(y_{head}, y_{torso}) \in \mathbb{R}_+$$

- Compute

$$y^* = \operatorname*{argmax}_{y_{head}, y_{torso}, \cdots} \psi(y_{head}, x)\psi(y_{torso}, x)\psi(y_{head}, y_{torso}) \cdots$$

Problem? Does not decompose anymore!

The General Recipe

Structured output function: $\mathcal{X}$ = anything $\to \mathcal{Y}$ = anything

1) Define auxiliary function $g : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$:

$$e.g. \qquad g(x,y) = \prod_i \psi_i(y_i, x) \prod_{i \sim j} \psi_{ij}(y_i, y_j, x)$$

2) Obtain $f : \mathcal{X} \to \mathcal{Y}$ by *maximimization*:

$$f(x) = \operatorname*{argmax}_{y \in \mathcal{Y}} g(x,y)$$

A Probabilistic View

Computer Vision problems usually deal with *uncertain* information

- Incomplete information (observe static images, projections, etc)
- Annotation is "noisy" (wrong or ambiguous cases)

Uncertainty is captured by (conditional) probability distributions: $p(y|x)$

- for input $x \in \mathcal{X}$, how *likely* is $y \in \mathcal{Y}$ the correct output?

We can also phrase this as

- what's the probability of observing y given x?
- how strong is our *belief* in y if we know x?

A Probabilistic View on $f : \mathcal{X} \to \mathcal{Y}$

Structured output function $\mathcal{X}$ = anything $\to$ $\mathcal{Y}$ = anything

We need to define an auxiliary function, $g : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$.

$$e.g. \qquad g(x, y) := p(y|x).$$

Then *maximimization*

$$f(x) = \operatorname*{argmax}_{y \in \mathcal{Y}} g(x, y) = \operatorname*{argmax}_{y \in \mathcal{Y}} p(y|x)$$

becomes *maximum a posteriori (MAP) prediction*.

Interpretation: The MAP estimate $y \in \mathcal{Y}$, is the most probable value (there can be multiple).

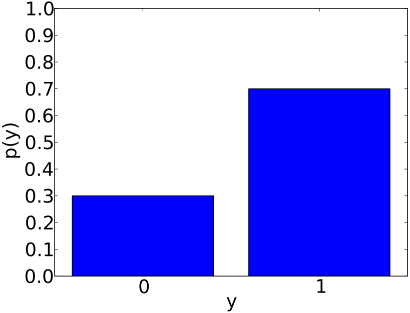
Probability Distributions

$$\forall y \in \mathcal{Y} \quad p(y) \geq 0 \qquad \text{(positivity)}$$

$$\sum_{y \in \mathcal{Y}} p(y) = 1 \qquad \text{(normalization)}$$

Example: binary ("Bernoulli") variable $y \in \mathcal{Y} = \{0, 1\}$

- 2 values,
- 1 degree of freedom



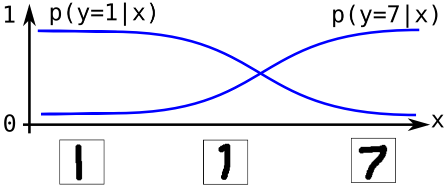
Conditional Probability Distributions

$$\forall x \in \mathcal{X}\ \forall y \in \mathcal{Y} \quad p(y|x) \geq 0 \qquad \text{(positivity)}$$

$$\forall x \in \mathcal{X}\ \sum_{y \in \mathcal{Y}} p(y|x) = 1 \qquad \text{(normalization w.r.t. } y\text{)}$$

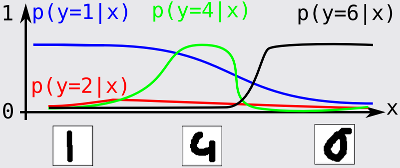
For example: **binary** prediction
$\mathcal{X} = \{\text{images}\}$, $y \in \mathcal{Y} = \{0, 1\}$

- each x: 2 values, 1 d.o.f.
  $\rightarrow$ two functions



Multi-class prediction, $y \in \mathcal{Y} = \{1, \ldots, K\}$

- each x: K values, $K-1$ d.o.f.
  $\rightarrow K-1$ *functions*
- or 1 vector-valued function with
  $K-1$ outputs

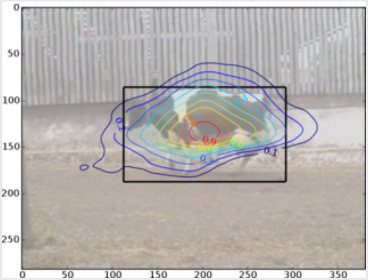


Typically: K functions, plus explicit normalization

Example: **predicting the center point of an object**

$y \in \mathcal{Y} = \{(1,1), \ldots, (width, height)\}$
- for each x: $|\mathcal{Y}| = W \cdot H$ values,

$y = (y_1, y_2) \in \mathcal{Y}_1 \times \mathcal{Y}_2$ with
$\mathcal{Y}_1 = \{1, \ldots, width\}$ and
$\mathcal{Y}_2 = \{1, \ldots, height\}$.
- each x: $|\mathcal{Y}_1| \cdot |\mathcal{Y}_2| = W \cdot H$ values,

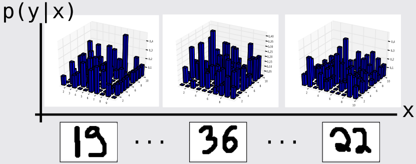


Structured objects: **predicting M variables jointly**

$\mathcal{Y} = \{1, K\} \times \{1, K\} \cdots \times \{1, K\}$

For each x:

- K^M values, $K^M - 1$ d.o.f.
  $\rightarrow K^M$ functions



Example: Object detection with **variable size bounding box**

$$\mathcal{Y} \subset \{1, \ldots, W\} \times \{1, \ldots, H\} \times \{1, \ldots, W\} \times \{1, \ldots, H\}$$

$$y = (\textit{left}, \textit{top}, \textit{right}, \textit{bottom})$$

For each x:

- $\frac{1}{4} W(W-1)H(H-1)$ values (millions to billions...)

Example: image denoising

$$\mathcal{Y} = \{640 \times 480 \text{ RGB images}\}$$

For each x:

- 16777216^{307200} values in $p(y|x)$
- $\geq 10^{2000000}$ functions
- How many atoms in universe?

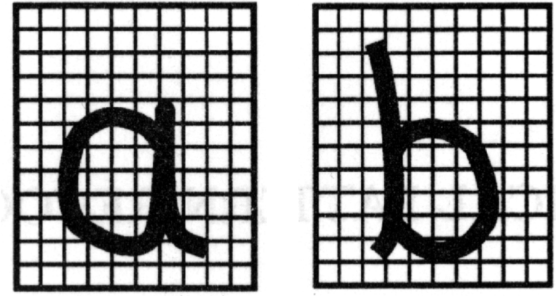
too much!

We cannot consider all possible distributions, we must impose **structure**.

Decision Theory

Digit classification

- Classify digits "a" versus "b"



The digits "a" and "b"

- **Goal:** classify new digits such that probability of error is minimized

Digit classification - Priors

Prior Distribution?

- How often do the letters “a” and “b” occur ?
- Let us assume

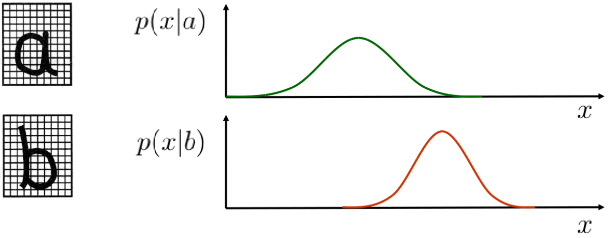
$$
\begin{aligned}
C_1 &= a \qquad p(C_1) = 0.75 \\
C_2 &= b \qquad p(C_2) = 0.25
\end{aligned}
$$

- The *prior* has to be a distribution, in particular

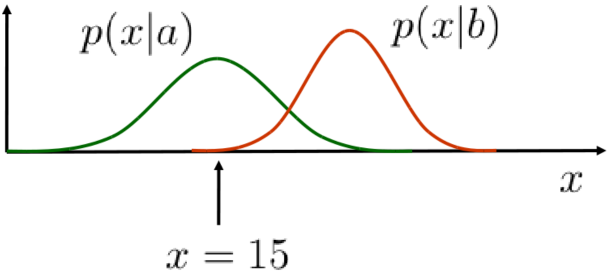
$$\sum_{k=1,2} p(C_k) = 1$$

Digit classification - Class conditionals

- We describe every digit using some feature vector x
  - the number of black pixels in each box
  - relation between width and height
- **Likelihood:** How likely has x been generated from $p(x \mid a)$ or $p(x \mid b)$?

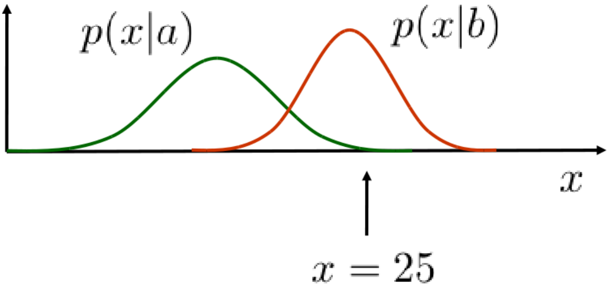


Digit classification



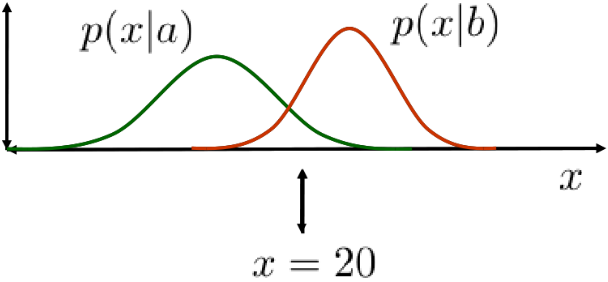
- Which class should we assign x to?
- Class a

Digit classification



- Which class should we assign x to ?
- Class b

Digit classification



- Which class should we assign x to ?
- The answer?

Bayes Theorem

- How do we formalize this?
- We already mentioned Bayes Theorem

$$p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)}$$

- Now we apply it:

$$p(C_k|x) = \frac{p(x|C_k)p(C_k)}{p(x)} = \frac{p(x|C_k)p(C_k)}{\sum_j p(x|C_j)p(C_j)}$$

Bayes Theorem

- Some terminology! Repeated from last slide:

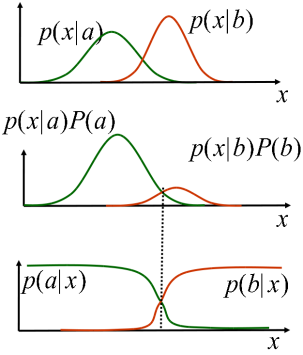
$$p(C_k|x) = \frac{p(x|C_k)p(C_k)}{p(x)} = \frac{p(x|C_k)p(C_k)}{\sum_j p(x|C_j)p(C_j)}$$

- We use the following names

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Normalization Factor}}$$

- Normalization Factor is also called the Partition Function or Evidence (commonly denoted with the symbol 'Z')

Bayes Theorem



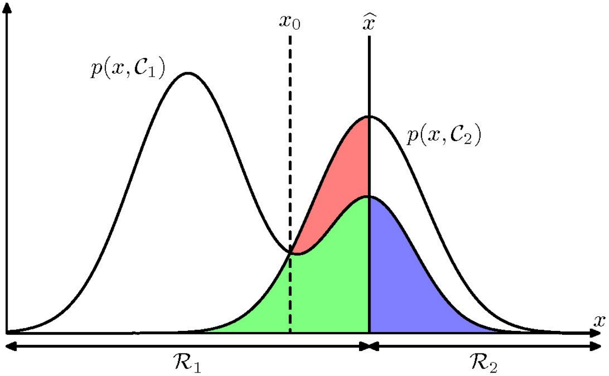
Likelihood

Likelihood $\times$ Prior

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Normalization Factor}}$$

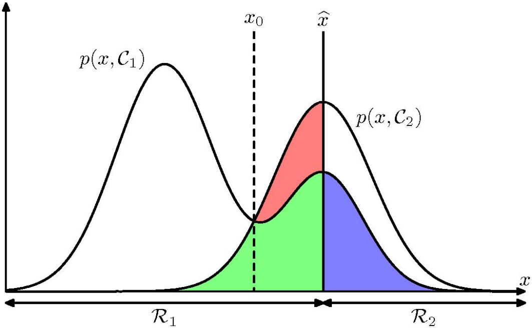
How to decide?

- Two class problem C_1, C_2, plotting Likelihood $\times$ Prior



- What is the probability of making an error?

Minmizing the Error



$$
\begin{aligned}
p(\text{error}) &= p(x \in R_2, C_1) + p(x \in R_1, C_2) \\
&= p(x \in R_2 | C_1) p(C_1) + p(x \in R_1 | C_2) p(C_2) \\
&= \int_{R_2} p(x|C_1) p(C_1) \mathrm{d}x + \int_{R_1} p(x|C_2) p(C_2) \mathrm{d}x
\end{aligned}
$$

General Loss Functions

- So far we considered misclassification error only
- This is also referred to as 0/1 loss
- Now suppose we are given a more general loss function

$$\begin{aligned} \Delta: \quad & \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+ \\ & (y, \hat{y}) \mapsto \Delta(y, \hat{y}) \end{aligned}$$

- How do we read this?
- $\Delta(y, \hat{y})$ is the cost we have to pay if y is the true class, but we predict $\hat{y}$ instead

Example: Predicting Cancer

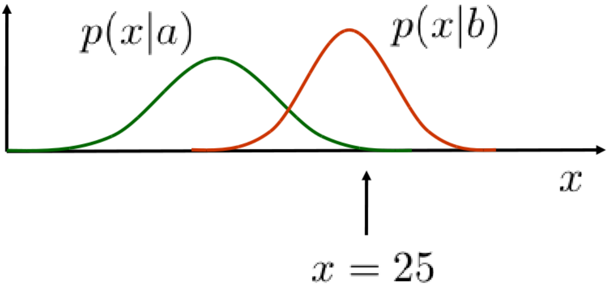
- General loss function:

$$\begin{aligned} \Delta : \quad & \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+ \\ & (y, \hat{y}) \mapsto \Delta(y, \hat{y}) \end{aligned}$$

- Given: X-Ray image
  - Question: Cancer yes or no?
  - Should we have a medical doctor check the patient?
- For discrete sets $\mathcal{Y}$ this is a loss matrix. How does it look?
- Loss function:

	cancer	normal
cancer	0	1000
normal	1	0

Digit Classification

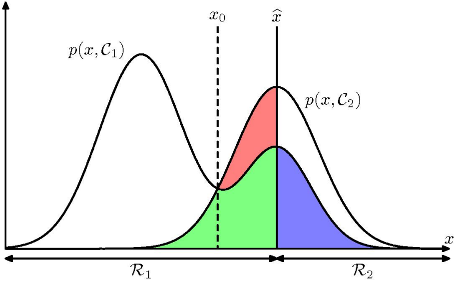


- Which class should we assign x to? ($p(a) = p(b) = 0.5$)
- The answer
- **It depends on the loss**

Minimizing Expected Error

- But we do not know the correct class y
- The expected error for x (averaged over all decisions):

$$\mathbb{E}[\Delta] = \sum_{k=1,\dots,K} \sum_{j=1,\dots,K} \int_{R_j} \Delta(C_k, C_j) p(x, C_k) \mathrm{d}x$$



Minimizing Expected Error

- But we do not know the correct class y
- The expected error for x (averaged over all decisions):

$$\mathbb{E}[\Delta] = \sum_{k=1,\ldots,K} \sum_{j=1,\ldots,K} \int_{R_j} \Delta(C_k, C_j) p(x, C_k) \mathrm{d}x$$

- And how do we predict, given an x? Decide on one y!

$$\begin{aligned} y^* &= \operatorname*{argmin}_{y \in \mathcal{Y}} \sum_{k=1,\ldots,K} \Delta(C_k, y) p(C_k | x) \\ &= \operatorname*{argmin}_{y \in \mathcal{Y}} \mathbb{E}_{p(\cdot|x)}[\Delta(\cdot, y)] \end{aligned}$$

Inference and Decision

- We broke down the process into two steps
  - Inference: obtaining the probabilities $p(C_k|x)$
  - Decision: Obtain optimal class assignment
- The probabilites $p(\cdot|x)$ represent our belief of the world
- The loss Δ tells us what to do with it!
- 0/1 loss implies deciding for max probability (exercise)

Three approaches to solve decision problems

1. Generative models: infer the class conditionals

$$p(x|\mathcal{C}_k), \quad k = 1, \dots, K$$

   then combine using Bayes Theorem

2. Discriminative models: infer posterior probabilities directly

$$p(\mathcal{C}_k|x)$$

3. Find discriminative function minimizing expected loss Δ

$$f : \mathcal{X} \to \{1, \dots, K\}$$

Let's discuss these options ...

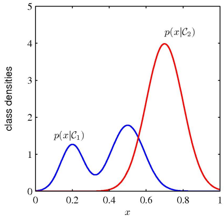
Generative Models

Pros:

- The name *generative* is because we explain the *generative* process of the data
- Intuitive, "understand" your process
- We can generate samples x from $p(x)$

Cons:

- With high dimensionality of $x \in \mathcal{X}$ we need large training set to determine the class-conditionals
- We may just not be interested in all quantities



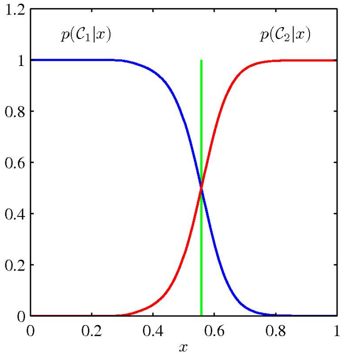
Discriminative Models

Pros:

- No need to model $p(x|C_k)$
  $\Rightarrow$ easier

Cons:

- No access to model $p(x|C_k)$



Discriminative Functions

Pros:

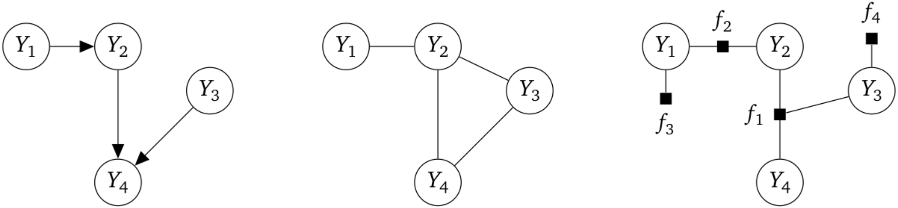
- One integrated system
- Directly estimate the quantity of interest $f(x)$
- *When solving a problem of interest, do not solve a harder / more general problem as an intermediate step. [Vladimir Vapnik]*

Cons:

- Need Δ during training time
- Revision of Δ requires re-learning
- No probabilities, no uncertainty, no reject?
- Prominent example: SVMs

Next Time ...

- ... we will meet our new friends:



(a) Bayesian Network **(b)** Markov Random Field **(c)** Factor Graph