Probabilistic Graphical Models and Their Applications

Bernt Schiele

Max Planck Institute for Informatics

slides adapted from Peter Gehler

November 4, 2020

Organization 1/2

- Lecture 2 hours/week
  - Wed: 14:15 – 16:00, via zoom
- Exercises 2 hours/week
  - Fri: 8:30 – 10:00, via zoom
  - Exercises start **this** Friday (Matlab primer)

Organization 2/2

Where to find what:

- http://www.mpi-inf.mpg.de/pgm
  - General information
- https://cms.sic.saarland/pgm20/
  - Slides
  - Recorded Lectures
  - Pointers to Books and Papers
  - Homework assignments
- “Semesterapparat” in library
- Registration: see cms webpage how to **register** (also includes mailinglist)

Exercises & Exam

- Exercises:
  - Typically one assignment per week
  - Theoretical and practical exercises
  - Starts with Matlab primer
  - Also includes programming project in the second part of the semester (you can select or propose your own topic)
  - To be done in groups of 2 – 3 students
  - Final Grade: 50% exercises, 50% oral exam (oral exam has to be passed obviously !)
- Exam
  - Oral exam at the end of the semester
  - Can be taken in English or German
- Tutors
  - Apratim Bhattacharyya (abhattac@mpi-inf.mpg.de)
  - Anna Kukleva (akukleva@mpi-inf.mpg.de)

Offers in our Research Group

- Master- and Bachelor Theses
- HiWi-positions, etc.

in

- Topics in machine learning
- Topics in computer vision
- Topics in machine learning applied to computer vision

- Come, talk to us

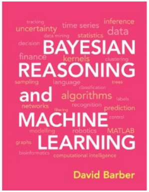
Topic overview

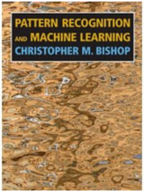
- Today: Recap – Probability and Decision theory
- Part 1: "Classic" Graphical Models
  - Basics (Directed, Undirected, Factor Graphs), Learning
  - Deterministic Inference (Sum-Prodcut, Junction Tree)
  - Approximate Inference (Loopy BP, Sampling, Variational)
- Part 2: Application to Computer Vision Problems (both classic and in the deep learning area)
  - Body Pose Estimation,
  - Semantic Segmentation,
  - Image Denoising, ...
- Part 3: Graph Neural Networks
  - Graph Convolutional Neural Networks, ...
  - and Applications ...

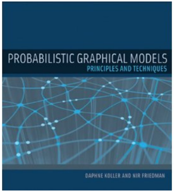
Literature (part 1)

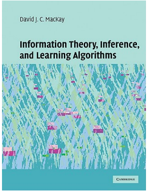
- All books in a "Semesterapparat"
- Main book for the graphical model part
  - Barber, **Bayesian Reasoning and Machine Learning**, Cambridge University Press, 2011, ISBN-13: 978-0521518147, `http://tinyurl.com/3flppuo`
- Extra References
  - Bishop, **Pattern Recognition and Machine Learning**, Springer 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 Universsity Press, 2003, ISBN-13: 978-0521642989

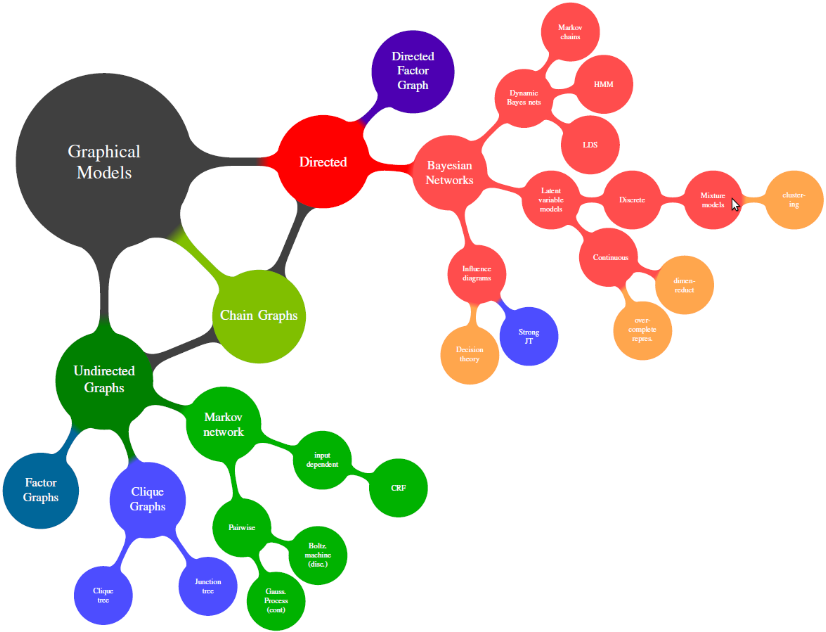
Literature (part 1)










Graphical Models
Directed
Directed Factor Graph
Bayesian Networks
Dynamic Bayes nets
Markov chains
HMM
LDS
Latent variable models
Discrete
Mixture models
cluster-ing
Continuous
dimen-reduct
over-complete repres.
Influence diagrams
Decision theory
Strong JT
Chain Graphs
Undirected Graphs
Markov network
input dependent
CRF
Pairwise
Boltz. machine (disc.)
Gauss. Process (cont)
Factor Graphs
Clique Graphs
Clique tree
Junction tree

Today's topics

- Overview: Machine Learning
  - What is machine learning ?
  - Different problem settings and examples
- Probability theory
- Decision theory, inference and decision

Machine Learning

Overview

Machine learning – what's that?

- Do you use machine learning systems already ?
- Can you think of an application ?

- Can you define the term "machine learning"?

- Goal of machine learning:
  - Machines that learn to perform a task from experience
- We can formalize this as

$$y = f(x; w) \tag{1}$$

  y is called *output variable*,
  x the *input variable* and
  w the model parameters (typically learned)
- Classification vs regression:
  - regression: y continuous
  - classification: y discrete (e.g. class membership)

- Goal of machine learning:
  - Machines that learn to perform a task from experience
- We can formalize this as

$$y = f(x; w) \tag{2}$$

  y is called *output variable*,
  x the *input variable* and
  w the model parameters (typically learned)
- learn... adjust the parameter w
- ... a task ... the function f
- ... from experience using a training dataset $\mathcal{D}$, where of either $\mathcal{D} = \{x_1, \ldots, x_n\}$ or $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$

Different Scenarios

- Unsupervised Learning
- Supervised Learning
- Reinforcement Learning

- Let's discuss

Supervised Learning

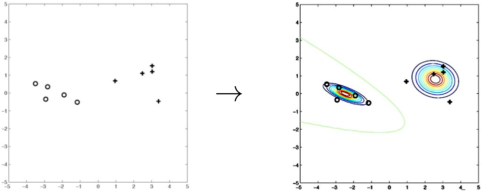
- Given are pairs of training examples from $\mathcal{X} \times \mathcal{Y}$

$$\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \tag{3}$$

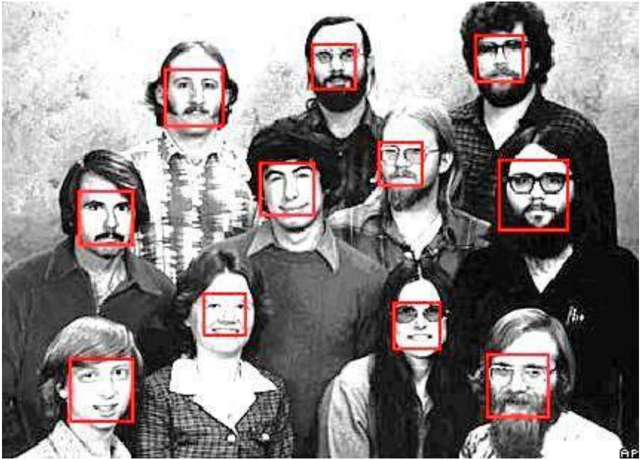
- Goal is to learn the relationship between x and y
- Given a new example point x predict y

$$y = f(x; w) \tag{4}$$

- We want to generalize to unseen data

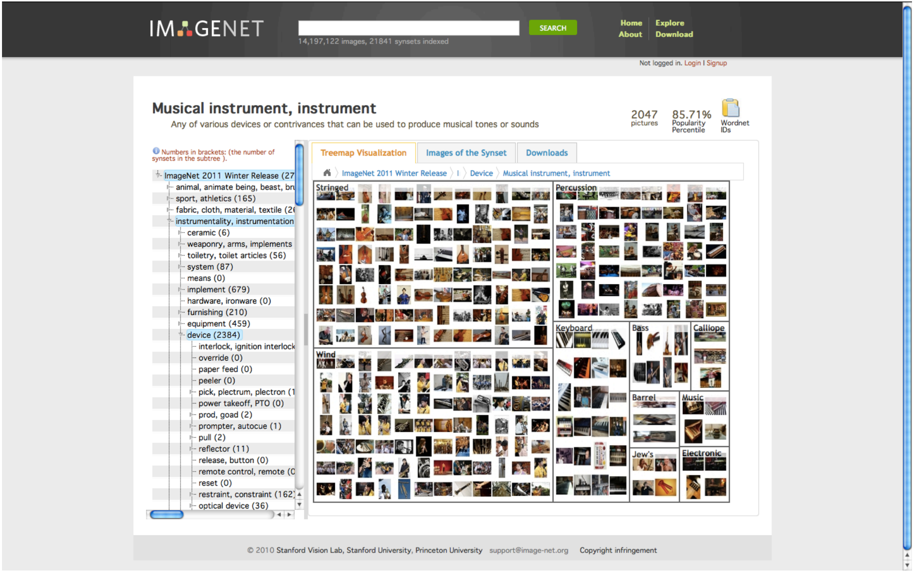


Supervised Learning – Examples



Face Detection

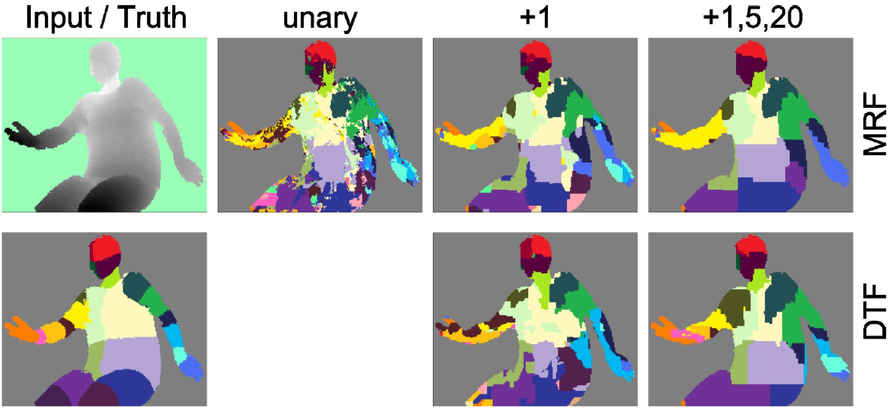
Supervised Learning – Examples



Supervised Learning – Examples

Semantic Image Segmentation

Supervised Learning – Examples



Body Part Estimation (in Kinect)

Figure from *Decision Tree Fields*, Nowozin et al., ICCV11

Supervised Learning – Examples

- Person identification
- Credit card fraud detection
- Industrial inspection
- Speech recognition
- Action classification in videos
- Human body pose estimation
- Visual object detection
- Prediction survival rate of a patient
- ...

Supervised Learning - Models

Flashing more keywords

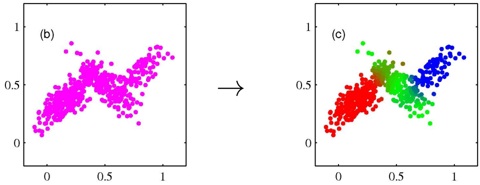
- Multilayer Perceptron (Backpropagation)
- (Deep) Convolutional Neural Networks (Backpropagation)
- Linear Regression, Logistic Regression
- Support Vector Machine (SVM)
- Boosting
- Graphical models

Unsupervised Learning

- We are given some input data points

$$\mathcal{D} = \{x_1, x_2, \ldots, x_n\} \tag{5}$$

- Goals:
  - Determine the data distribution $p(x) \rightarrow$ density estimation
  - Visualize the data by projections $\rightarrow$ dimensionality reduction
  - Find groupings of the data $\rightarrow$ clustering



Unsupervised Learning – Examples

Image Priors for Denoising

Unsupervised Learning – Examples

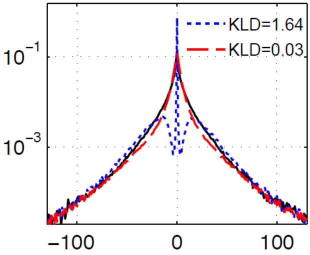
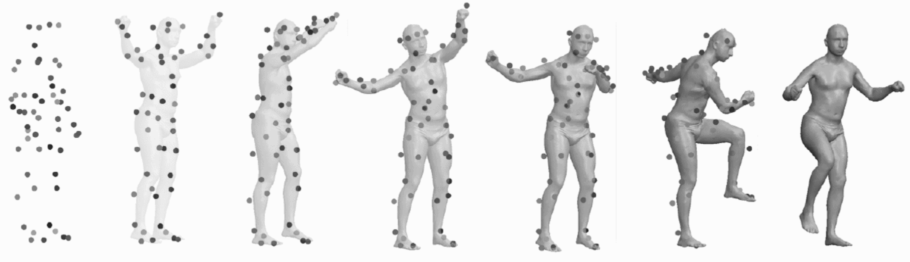


Image Priors for Inpainting

Image from *"A generative perspective on MRFs in low-level vision"*, Schmidt et al., CVPR2010

black line: statistics form original images, blue and red: statistics after applying two different algorithms

Unsupervised Learning – Examples



Human Shape Model

SCAPE: Shape Completion and Animation of People, Anguelov et al.

Unsupervised Learning – Examples

- Clustering scientific publications according to topics
- A generative model for human motion
- Generating training data for Microsoft Kinect xbox controller
- Clustering flickr images
- Novelty detection, predicting outliers
  - Anomality detection in visual inspection
  - Video surveillance

Unsupervised Learning – Models

Just *flashing* some keywords ($\rightarrow$ Machine Learning)

- Mixture Models
- Neural Networks
- K-Means
- Kernel Density Estimation
- Principal Component Analysis (PCA)
- Graphical Models (here)

Reinforcement Learning

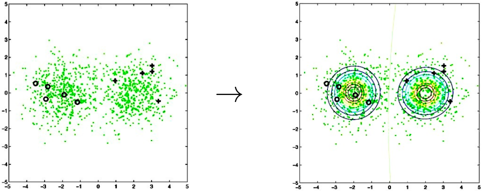
- Setting: given a situation, find an action to maximize a reward function
- Feedback:
  - we only get feedback of how well we are doing
  - we do *not* get feedback what the best action would be ("indirect teaching")
- Feedback given as reward:
  - each action yields reward, or
  - a reward is given at the end (e.g. robot has found his goal, computer has won game in Backgammon)
- **Exploration:** try out new actions
- **Exploitation:** use known actions that yield high rewards
- Find a good trade-off between exploration and exploitation

Variations of the general theme

- All problems fall in these broad categories
- But your problem will surely have some extra twists
- Many different variations of the aforementioned problems are studied separately
- Let's look at some ...

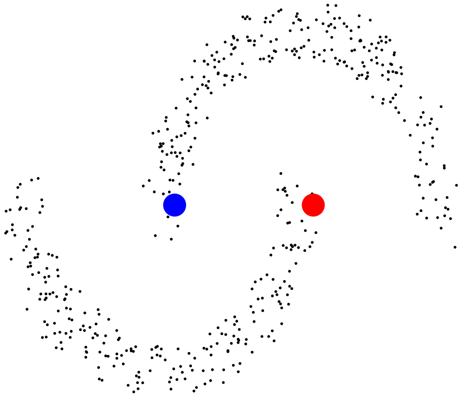
Semi-Supervised Learning

- We are given a dataset of l labeled examples
$$\mathcal{D}_l = \{(x_1, y_1), \ldots, (x_l, y_l)\}$$
as in supervised learning
- Additionally we are given a set of u unlabeled examples
$$\mathcal{D}_u = \{x_{l+1}, \ldots, x_{l+u}\}$$
as in unsupervised learning
- Goal is $y = f(x; w)$
- Question: how can we utilize the extra information in $\mathcal{D}_u$?



Semi-Supervised Learning: Two Moons

- Two labeled examples (red and blue) and additional unlabeled black dots



Two moons

Transductive Learning

- We are given a set of labeled examples

$$\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\} \tag{6}$$

- Additionally we know the **test data points** $\{x_1^{te}, \ldots, x_m^{te}\}$ (not their labels!)
- Can we do better, including this knowledge?
- This should be easier than making predictions for the entire set $\mathcal{X}$

On-line Learning

- The training data is presented step-by-step and is never available entirely
- At each time-step t we are given a new datapoint x_t (or (x_t, y_t))
- When is online learning a sensible scenario?
    - We want to continuously update the model – we can train a model with little data, but the model should become better over time when more data is available (similar to how humans learn)
    - We have limited storage for data and the model – a viable setting for large-scale datasets (e.g. the size of the internet)
- How do we learn in this scenario?

Large-Scale Learning

- Learning with millions of examples
- Study fast learning algorithms (e.g. parallelizable, special hardware)
- Problems of storing the data, computing the features, etc.
- There is no strict definition for “large-scale”
- Small-scale learning: limiting factor is number of examples
- Large-scale learning: limited by maximal time for computation (and/or maximal storage capacity)

Active Learning

- We are given a set of examples

$$\mathcal{D} = \{x_1, \ldots, x_n\} \tag{7}$$

- Goal is to learn $y = f(x; w)$
- Each label y_i **costs** something, e.g. $C_i \in \mathbb{R}_+$
- Question: How to learn well while paying little?
- This is almost always the case, labeling is expensive

Structured Output Learning

- We are given a set of training examples
$$\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\},$$
but $y \in \mathcal{Y}$ contains more structure than $y \in \mathbb{R}$ or $y \in \{-1, 1\}$
- Consider binary image segmentation
  - y is entire image labeling
  - $\mathcal{Y}$ is the set of all labelings $2^{\#pixels}$

- Other examples: y could be a graph, a tree, a ranking, ...
- Goal is to learn a function $f(x, y; w)$ and predict
$$y = \operatorname*{argmax}_{\bar{y} \in \mathcal{Y}} f(x, \bar{y}; w)$$

Some final comments

- All topics are under active development and research
- Supervised classification: basically understood
- Broad range of applications, many exciting developments
- Adopting a “ML view” has far reaching consequences, it touches problems of empirical sciences in general

Probability Theory

Brief Review

Brief Review

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

$$p(x) \text{ for } p(X = x) \quad \in [0, 1] \tag{8}$$

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

$$p(X), \tag{9}$$

  we denote the *probability distribution* over X

Brief Review

- Two random variables (RVs) are called independent if

$$p(X = x, Y = y) = p(X = x)p(Y = y) \tag{10}$$

- Joint probability (of X and Y)

$$p(x, y) \text{ instead } p(X = x, Y = y) \tag{11}$$

- Conditional probability

$$p(x|y) \text{ instead } p(X = x|Y = y) \tag{12}$$

The Rules of Probability

- Sum rule

$$p(X) = \sum_{y \in \mathcal{Y}} p(X, Y = y) \tag{13}$$

we "marginalize out y".
$p(X = x)$ is also called a marginal probability

- Product Rule

$$p(X, Y) = p(Y|X)p(X) \tag{14}$$

- And as a consequence: Bayes Theorem or Bayes Rule

$$p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)} \tag{15}$$

Vocabulary

- Joint Probability

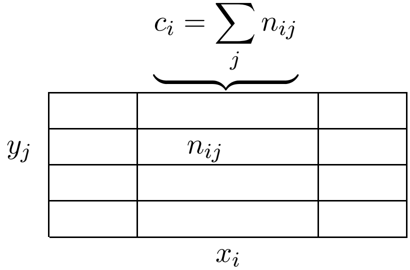
$$p(x_i, y_j) = \frac{n_{ij}}{N}$$

- Marginal Probability

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

- Conditional Probability

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



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

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

$$p(X \in (a, b)) = \int_a^b p(x) \mathrm{d}x \tag{16}$$

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

Probability Densities

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

$$p(x) \geq 0 \quad (17)$$

$$\int_{-\infty}^{\infty} p(x)\mathrm{d}x = 1 \quad (18)$$

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

$$P(z) = \int_{-\infty}^{z} p(x)\mathrm{d}x \quad (19)$$

Probability Densities

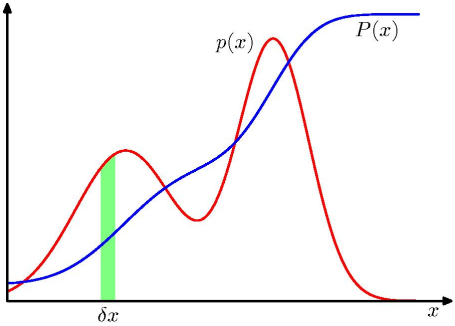


Figure: Probability density of a continuous variable

Expectation and Variances

- Expectation

$$\mathbb{E}[f] = \sum_{x \in \mathcal{X}} p(x) f(x) \tag{20}$$

$$\mathbb{E}[f] = \int_{x \in \mathcal{X}} p(x) f(x) \mathrm{d}x \tag{21}$$

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

$$\mathbb{E}_{p(\cdot|y)}[f] = \sum_{x \in \mathcal{X}} p(x|y) f(x) \tag{22}$$

- Variance

$$\text{var}[f] = \mathbb{E}\left[(f(x) - \mathbb{E}[f(x)])^2\right] \tag{23}$$

Decision Theory

Digit Classification

- Classify digits "a" versus "b"

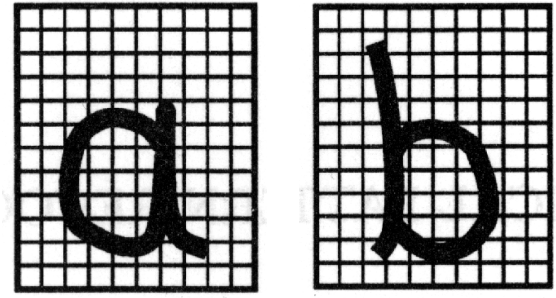


Figure: The digits "a" and "b"

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

Digit Classification - Priors

Prior Distribution

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

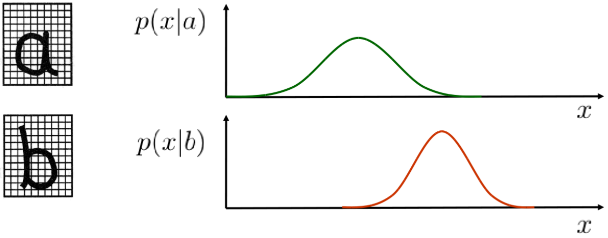
$$C_1 = a \qquad p(C_1) = 0.75 \tag{24}$$
$$C_2 = b \qquad p(C_2) = 0.25 \tag{25}$$

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

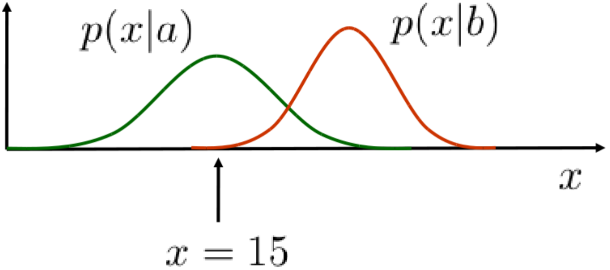
$$\sum_{k=1,2} p(C_k) = 1 \tag{26}$$

Digit Classification - Class Conditionals

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

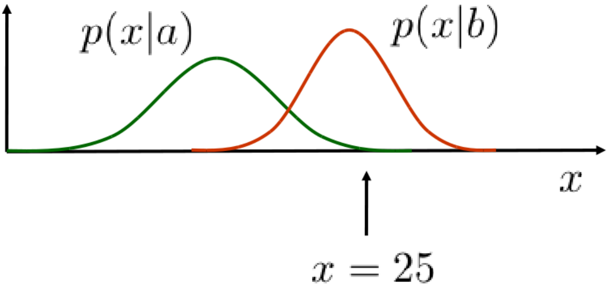


Digit Classification



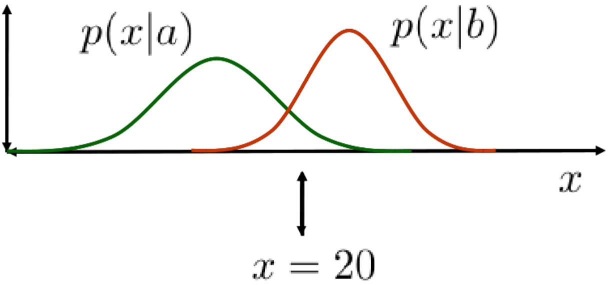
- Which class should we assign x to ?
- The answer
- Class a

Digit Classification



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

Digit Classification



- Which class should we assign x to ?
- The answer
- Class a, since p(a)=0.75

Bayes Theorem

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

$$p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)} \tag{27}$$

- Now we apply it

$$p(C_k|x) = \frac{p(x|C_k)p(C_k)}{p(x)} = \frac{p(x|C_k)p(C_k)}{\sum_j p(x|C_j)p(C_j)} \tag{28}$$

Bayes Theorem

- Some terminology! Repeated from last slide:

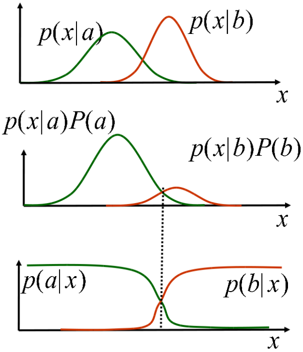
$$p(C_k|x) = \frac{p(x|C_k)p(C_k)}{p(x)} = \frac{p(x|C_k)p(C_k)}{\sum_j p(x|C_j)p(C_j)} \tag{29}$$

- We use the following names

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Normalization Factor}} \tag{30}$$

- Here the normalization factor is easy to compute. Keep an eye out for it, it will haunt us until the end of this class
(and longer :))
- It is also called the Partition Function, common symbol Z

Bayes Theorem



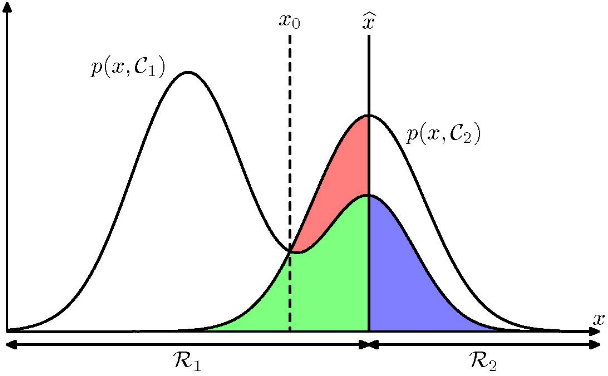
Likelihood

Likelihood × Prior

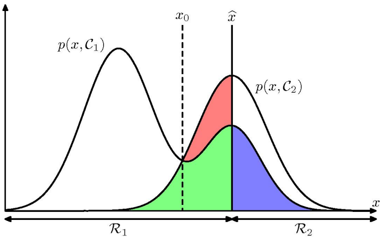
$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Normalization Factor}}$$

How to Decide?

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



Minmizing the Error



$$
\begin{aligned}
p(\text{error}) &= p(x \in R_2, C_1) + p(x \in R_1, C_2) && (31)\\
&= p(x \in R_2|C_1)p(C_1) + p(x \in R_1|C_2)p(C_2) && (32)\\
&= \int_{R_2} p(x|C_1)p(C_1)\mathrm{d}x + \int_{R_1} p(x|C_2)p(C_2)\mathrm{d}x && (33)
\end{aligned}
$$

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

$$\Delta : \quad \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+ \tag{34}$$
$$(y, \hat{y}) \mapsto \Delta(y, \hat{y}) \tag{35}$$

- 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

$$\Delta: \quad \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+ \tag{36}$$

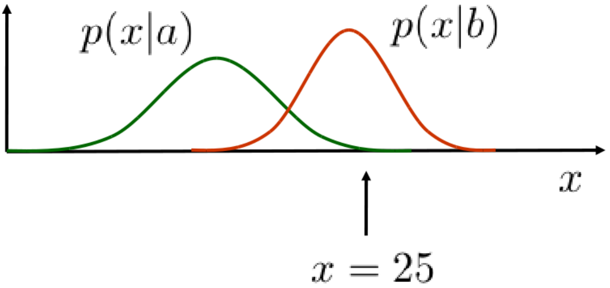
$$(y, \hat{y}) \mapsto \Delta(y, \hat{y}) \tag{37}$$

- Given: X-Ray image, Question: Cancer yes or no?
  Should we have another medical check of the patient?

		diagnosis :	
		cancer	normal
truth :	cancer	0	1000
	normal	1	0

- For discrete sets $\mathcal{Y}$ this is a loss matrix

Digit Classification

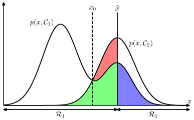


- Which class should we assign x to? $(p(a) = p(b) = 0.5)$
- The answer
- **It depends on the loss**

Minmizing Expected Loss (or Error)

- The expected loss for x (averaged over all decisions)

$$\mathbb{E}[\Delta] = \sum_{k=1,\ldots,K} \sum_{j=1,\ldots,K} \int_{R_j} \Delta(C_k, C_j) p(x, C_k) \mathrm{d}x \tag{38}$$



- And how do we predict? Decide on one y!

$$\begin{aligned} y^* &= \operatorname*{argmin}_{y \in \mathcal{Y}} \sum_{k=1,\ldots,K} \Delta(C_k, y) p(C_k|x) && (39) \\ &= \operatorname*{argmin}_{y \in \mathcal{Y}} \mathbb{E}_{p(\cdot|x)}[\Delta(\cdot, y)] && (40) \end{aligned}$$

Inference and Decision

- We broke down the process into two steps
  - Inference: obtaining the probabilities $p(C_k|x)$
  - Decision: Obtain optimal class assignment
- Two steps !!
- 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

$$p(x|\mathcal{C}_k), \quad k = 1, \ldots, K \tag{41}$$

then combine using Bayes Theorem $p(\mathcal{C}_k|x) = \frac{p(x|\mathcal{C}_k)p(\mathcal{C}_k)}{p(x)}$

2. Discriminative models: infer posterior probabilities directly

$$p(\mathcal{C}_k|x) \tag{42}$$

3. Find a discriminative function minimizing Expected Loss Δ

$$f : \mathcal{X} \rightarrow \{1, \ldots, K\} \tag{43}$$

Let's discuss these options

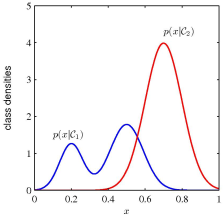
Generative Models

Pros:

- The name *generative* is because we can *generate* samples from the learnt distribution
- We can infer $p(x|\mathcal{C}_k)$ (or $p(x)$ for short)

Cons:

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



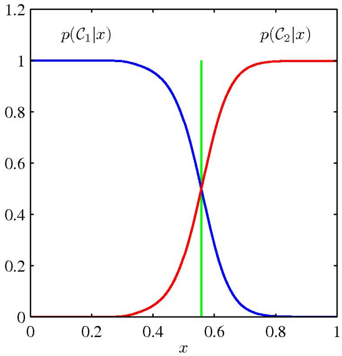
Discriminative Models

Pros:

- No need to model $p(x|\mathcal{C}_k)$ (i.e. in general easier)

Cons:

- No access to model $p(x|\mathcal{C}_k)$



Discriminative Functions

When solving a problem of interest, do not solve a harder / more general problem as an intermediate step.

– *Vladimir Vapnik*

Pros:

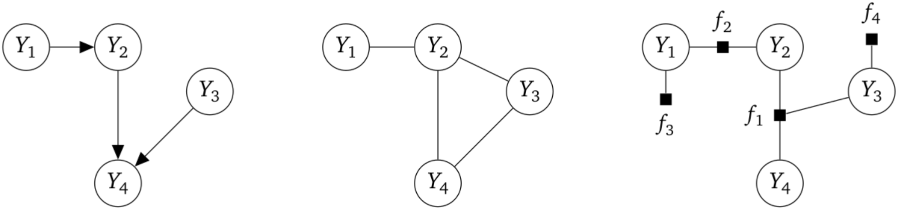
- One integrated system, we directly estimate the quantity of interest

Cons:

- Need Δ during training time – revision requires re-learning
- No access to probabilities or uncertainty, thus difficult to reject decision?
- Prominent example: Support Vector Machines (SVMs)

Next Time ...

- ... we will meet our new friends:



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