Machine Learning II:
(Applications)

Prepared by:
Prof. Dr. Visvanathan Ramesh

References and Sources: Nils Bertschinger (ML II lecture slides)
Simon Prince (Learning and Vision)

Outline

- **Recap – Nil Bertschinger Lectures**
  - Approaches to Machine Learning
  - Introduction to Probability, Bayes rule, Probability Distributions
  - Bayesian Machine Learning
  - Parametric/Non-parametric Bayesian methods
  - Gaussian Processes
  - Link to Neural Networks
  - Bayesian Nonparametrics
    - Dirichlet Processes, Chinese Restaurant Process, Indian Buffet Process
  - Inference by Sampling , MCMC – Metropolis-Hastings, Gibbs Sampler, HMC sampler

  - What is not covered yet? (And what is planned for the rest of the weeks)
    - Recap of past classes , discussion
    - Example application case studies in computer vision
    - Variational Methods

Brief Recap

Approaches to Machine Learning

- Data-driven
  - Very large data sets ... "Big Data"
  - Non-parametric models, e.g. k-NN
- Model-driven
  - Can be used for small data sets
  - Parametric models

Note: As models become more complex any data set is "small"
$\Longrightarrow$ Recent rise of model based machine learning

Model based Machine Learning

General setup of model based ML:

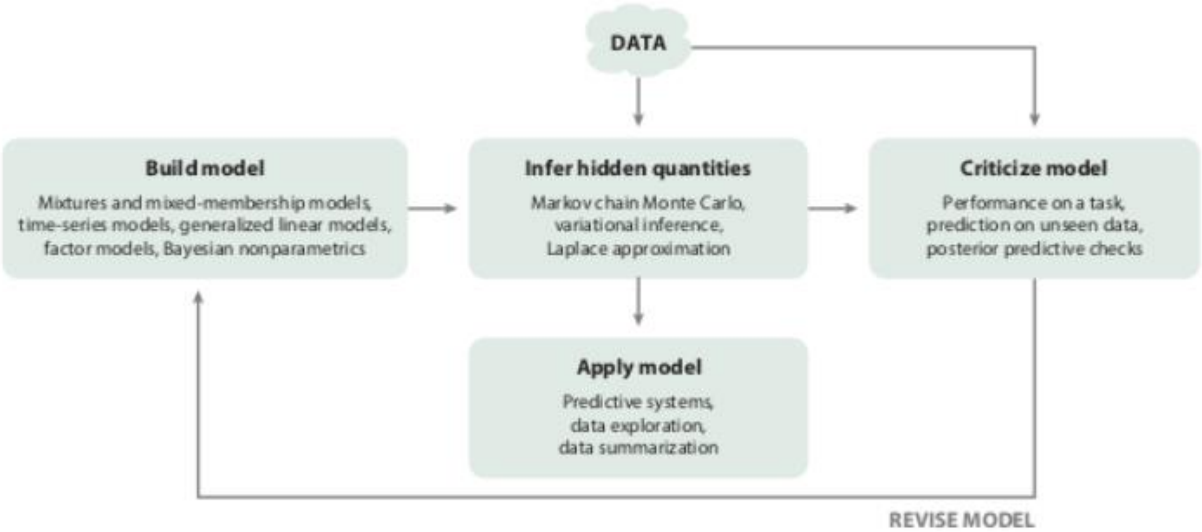


Fig. from: David M. Blei, *Build, Compute, Critique, Repeat: Data Analysis with Latent Variable Models*, Annu. Rev. Stat. Appl. 2014. 1:20332

Terminology

- **Supervised:** Patterns whose class/output is known a-priori are used for training (*labelled training data*)
  - *Regression:* Real-valued output
    Typical examples: Interpolation, (Time-series) Prediction
  - *Classification:* Categorical output
    Typical examples: Face recognition, Identity authentification, Speech recognition
- **Unsupervised:** Number of classes is (in general) unknown and no labelled data are available
  Typical examples: Cluster analysis, Recommendation systems

Bayesian Machine Learning

Bayesian statistics:

- Principled and logically consistent way to reason under uncertainty
  Prior $\xrightarrow{\textit{Data}}$ Posterior (belief update)
- Especially useful when taking decisions or making predictions

Bayesian machine learning:

- Statistical modeling:

$$p(\underbrace{\mathbf{x}}_{\text{Data}}, \underbrace{\mathbf{z}}_{\substack{\text{Latent variables} \\ \text{Parameters}}}) = \underbrace{p(\mathbf{z})}_{\textit{prior}} \underbrace{p(\mathbf{x}|\mathbf{z})}_{\textit{likelihood}}$$

- Conceptually simple, but computationally challenging

Bayesian Machine Learning

Bayesian machine learning:

- Bayesian modeling requires prior assumptions:
  - Parametric models, e.g. linear regression
  - Bayesian non-parametrics:
    - Flexible models with infinite-dimensional parameter spaces
    - Effective number of parameters grow with amount of data

  But, explicit about prior assumptions
  - *No free lunch theorem*: Assumption-free learning is impossible!
- Takes uncertainty into account

  Bayesian Occam's razor: Automatic penalty for model complexity
- Computational challenge: Posterior $p(\mathbf{z}|\mathbf{x})$ often intractable
  - Sampling algorithms
  - Variational approximations

Course Outline:

Machine Learning II course ... Focus on Bayesian methods

- Motivation: Bayesian vs frequentist statistics
- Decision theory: Handling uncertainty, loss functions
- Probability theory: Conjugate priors
- Modeling: Latent variables, hierarchical models, Bayesian non-parametrics
- Model selection: Marginal likelihood, sparsity priors
- Algorithms: Variational Bayes (ELBO), sampling methods

Potential applications

- Social data: Voting results, network models
- Economic data: GDP forecasting, volatility modeling
- Computer vision: Detection, tracking, recognition, segmentation
- ...

Computer vision: models, learning and inference

Source: Chapter 6 , 7

Computer Vision: Models, Learning and Inference (Simon Prince)

Structure

Computer vision models

- Two types of model

Worked example 1: Regression

Worked example 2: Classification

Which type should we choose?

Applications

Observe measured data, x

Draw inferences from it about state of world, w

Examples:

- Observe adjacent frames in video sequence
- Infer camera motion

- Observe image of face
- Infer identity

- Observe images from two displaced cameras
- Infer 3d structure of scene

Regression vs. Classification

Observe measured data, x

Draw inferences from it about world, w

When the world state w is continuous we'll call this regression

When the world state w is discrete we call this classification

Ambiguity of visual world

Unfortunately visual measurements may be compatible with more than one world state w

- Measurement process is noisy
- Inherent ambiguity in visual data

Conclusion: the best we can do is compute a probability distribution Pr(w|x) over possible states of world

Refined goal of computer vision

Take observations x

Return probability distribution Pr(w|x) over possible worlds compatible with data

(not always tractable – might have to settle for an approximation to this distribution, samples from it, or the best (MAP) solution for w)

Components of solution

We need

A model that mathematically relates the visual data x to the world state w. Model specifies family of relationships, particular relationship depends on parameters θ

A learning algorithm: fits parameters θ from paired training examples x_i, w_i

An inference algorithm: uses model to return $Pr(w|x)$ given new observed data x.

Types of Model

The model mathematically relates the visual data x to the world state w. Two main categories of model

1. **Model contingency of the world on the data Pr(w|x)**
2. **Model contingency of data on world Pr(x|w)**

Generative vs. Discriminative

1. **Model contingency of the world on the data Pr(w|x)**

   (DISCRIMINATIVE MODEL)

2. **Model contingency of data on world Pr(x|w)**

   (GENERATIVE MODELS)

Generative as probability model over data and so when we draw samples from model, we GENERATE new data

Type 1: Model Pr(w|x) - Discriminative

How to model Pr(w|x)?

1. Choose an appropriate form for Pr(**w**)
2. Make parameters a function of **x**
3. Function takes parameters θ that define its shape

Learning algorithm: learn parameters θ from training data x,w

Inference algorithm: just evaluate Pr(w|x)

Type 2: Pr(x|w) - Generative

How to model Pr(x|w)?

1. Choose an appropriate form for Pr(**x**)
2. Make parameters a function of **w**
3. Function takes parameters θ that define its shape

Learning algorithm: learn parameters θ from training data x,w

Inference algorithm: Define prior Pr(w) and then compute Pr(w|x) using Bayes' rule

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

Summary

Two different types of model depend on the quantity of interest:

1. **Pr(w|x)** Discriminative
2. **Pr(w|x)** Generative

Inference in discriminative models easy as we directly model posterior Pr(w|x). Generative models require more complex inference process using Bayes' rule

Structure

Computer vision models

- Two types of model

Worked example 1: Regression

Worked example 2: Classification

Which type should we choose?

Applications

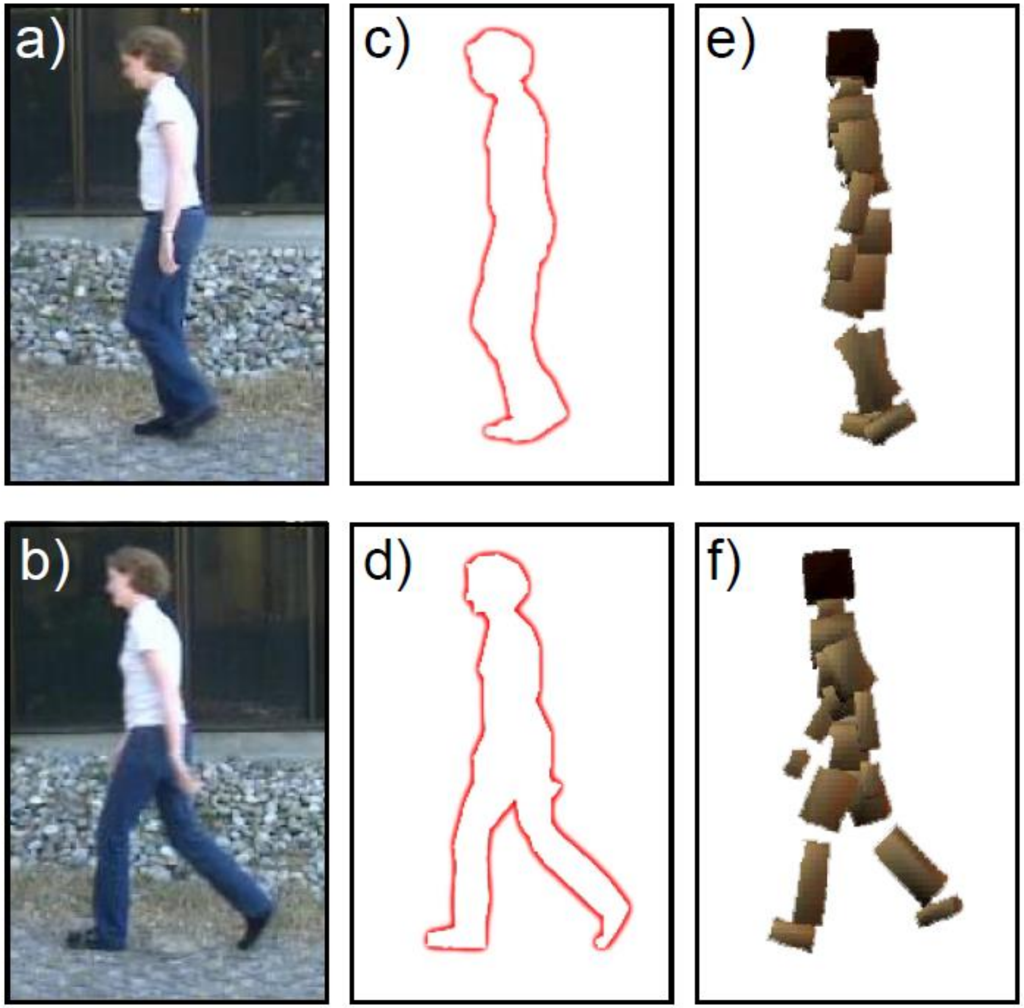
Worked example 1: Regression

Consider simple case where

- we make a univariate continuous measurement **x**
- use this to predict a univariate continuous state **w**

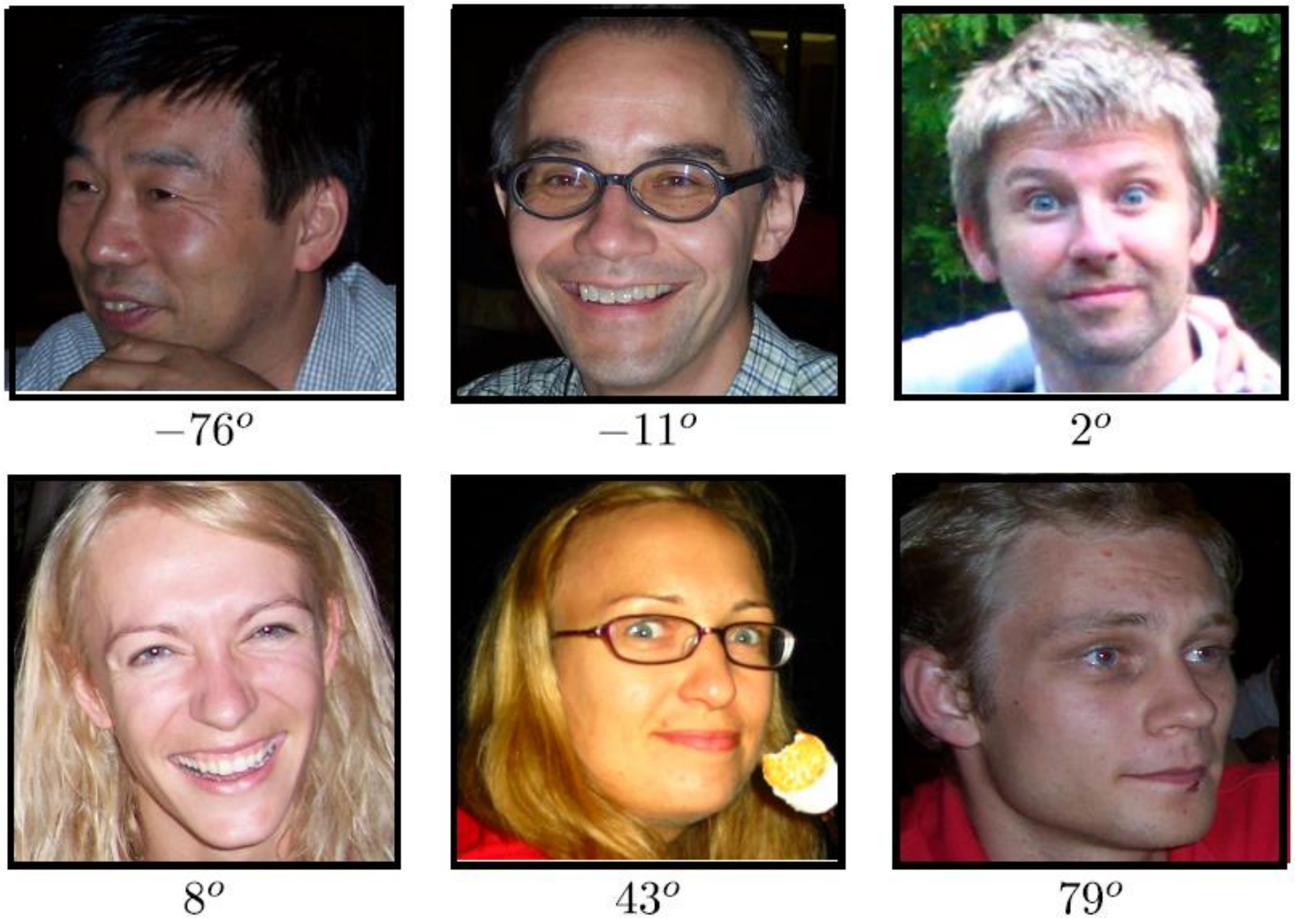
(regression as world state is continuous)

Regression application 1: Pose from Silhouette



Computer vision: models, learning and inference. ©2011 Simon J.D. Prince 23

Regression application 2: Head pose estimation



Worked example 1: Regression

Consider simple case where

- we make a univariate continuous measurement **x**
- use this to predict a univariate continuous state **w**

(regression as world state is continuous)

Type 1: Model Pr(w|x) - Discriminative

How to model Pr(w|x)?

1. Choose an appropriate form for Pr(**w**)
2. Make parameters a function of **x**
3. Function takes parameters θ that define its shape

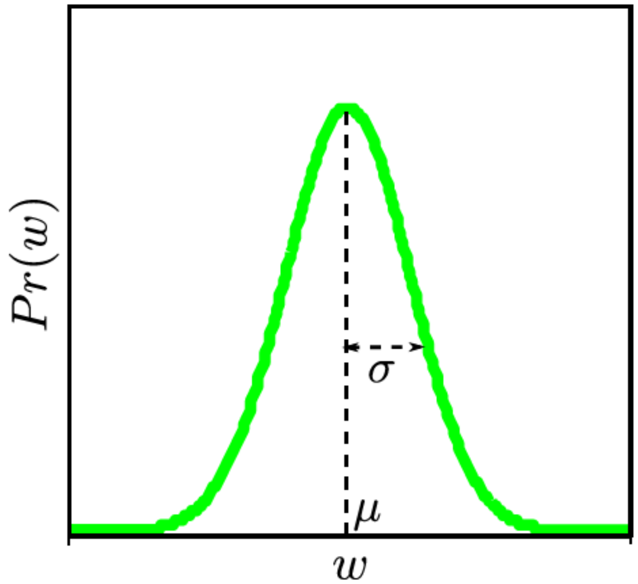
Learning algorithm: learn parameters θ from training data x,w

Inference algorithm: just evaluate Pr(w|x)

Type 1: Model Pr(w|x) - Discriminative

How to model Pr(w|x)?

1. Choose an appropriate form for Pr(**w**)
2. Make parameters a function of **x**
3. Function takes parameters θ that define its shape



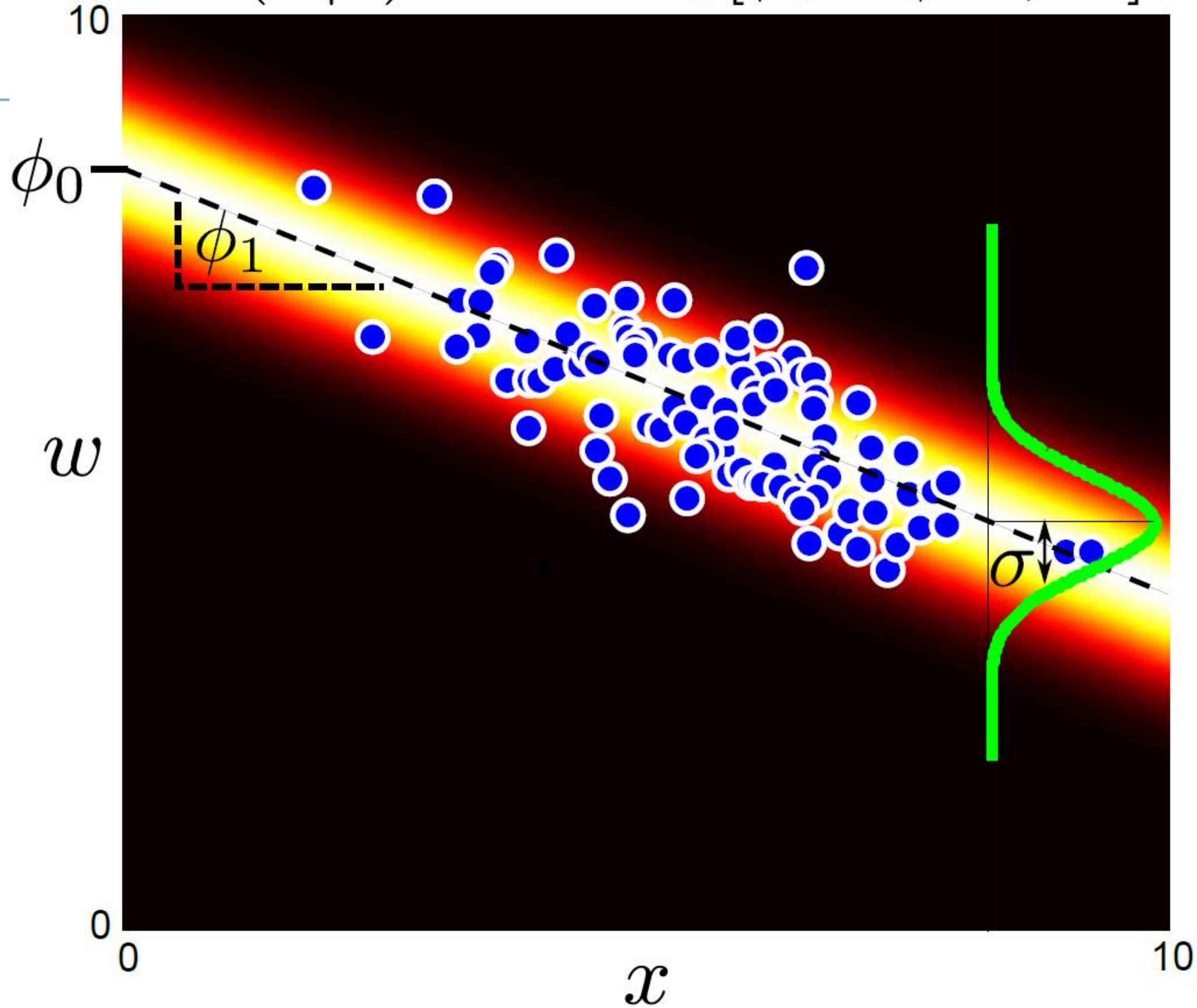
1. Choose normal distribution over w
2. Make mean μ linear function of x (variance constant)

$$Pr(w|x,\boldsymbol{\theta}) = \text{Norm}_w\left[\phi_0 + \phi_1 x, \sigma^2\right]$$

3. Parameters are ϕ_0, ϕ_1, σ^2.

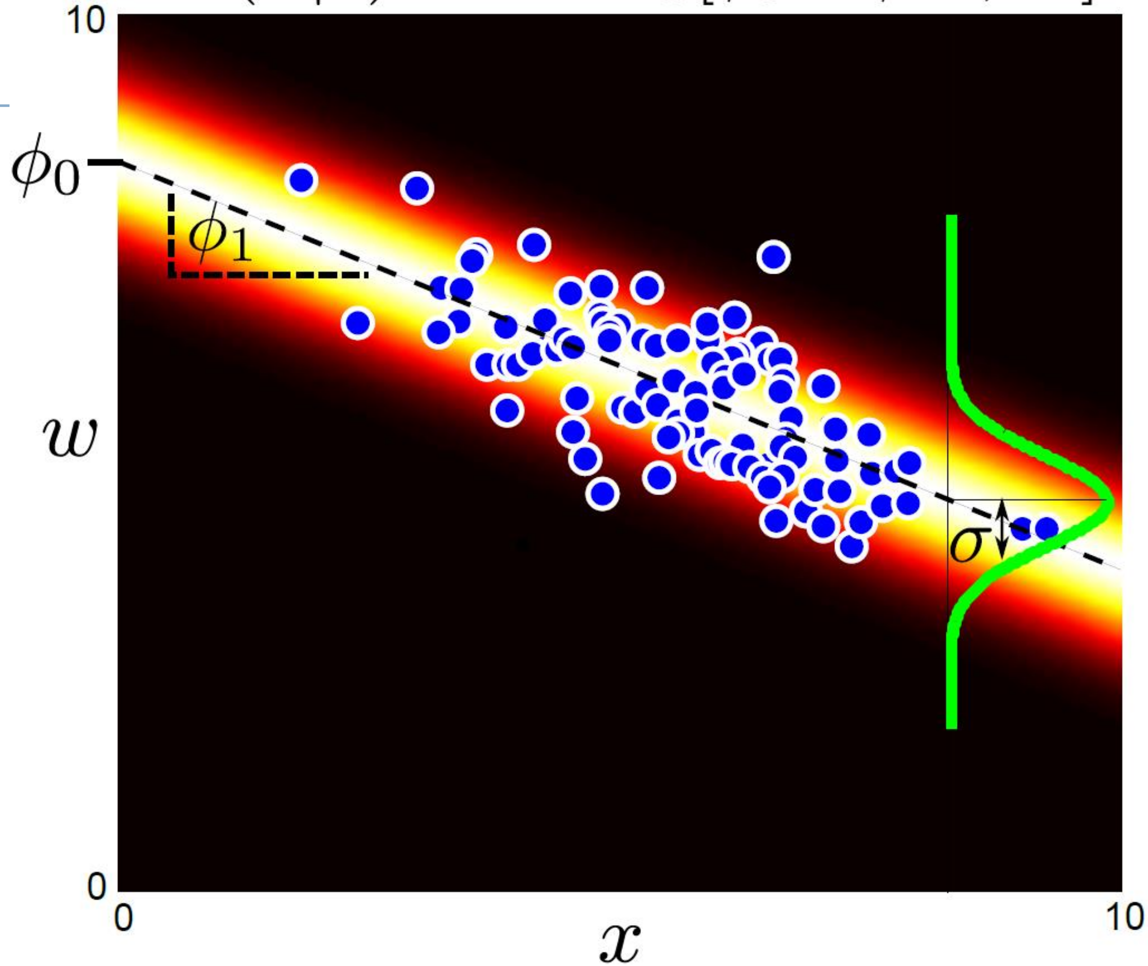
This model is called *linear regression.*

$$Pr(w|x) = \text{Norm}_w[\phi_0 + \phi_1 x, \sigma^2]$$



Parameters $\theta = \{\phi_0, \phi_1, \sigma^2\}$ are y-offset, slope and variance

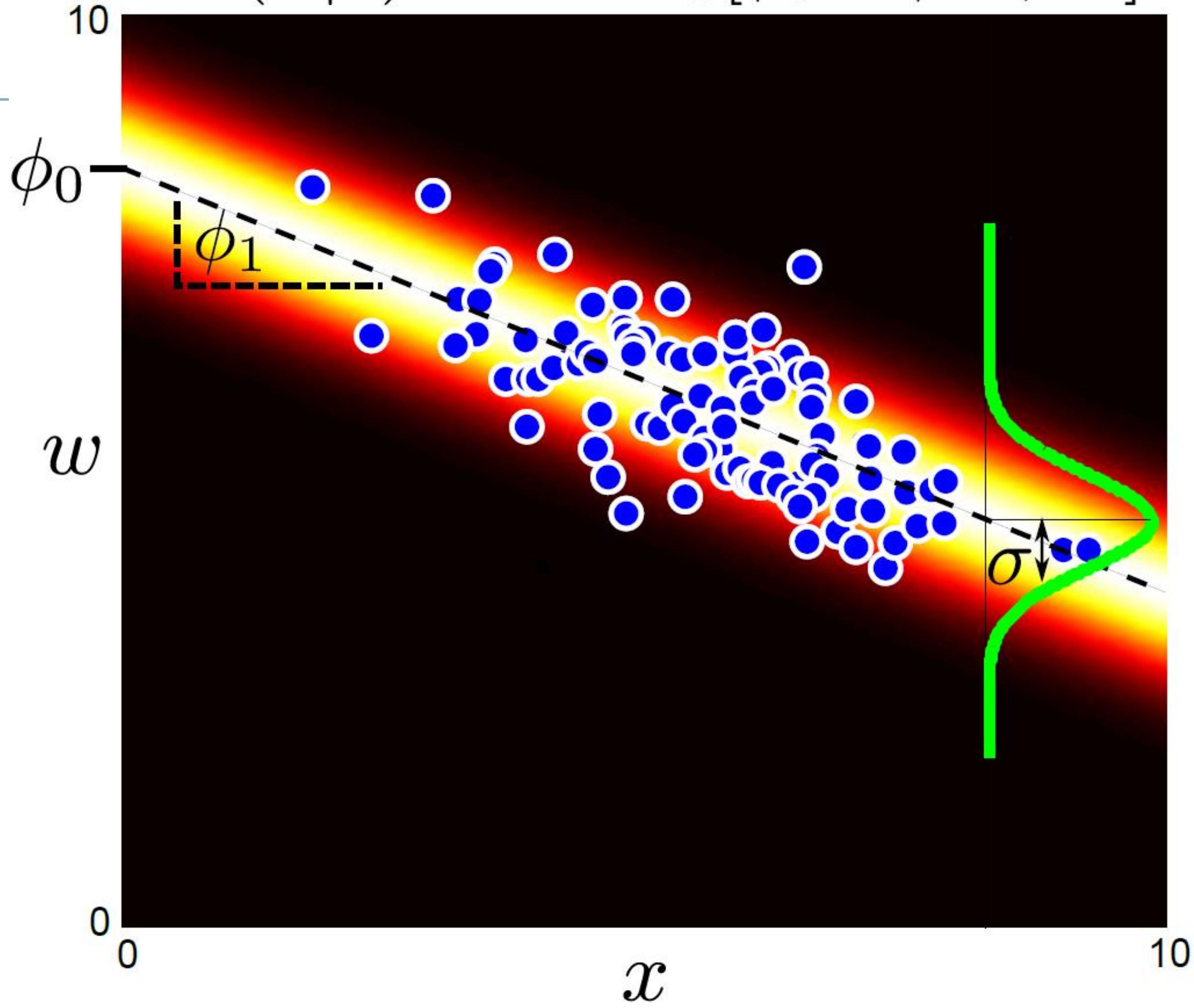
$$Pr(w|x) = \text{Norm}_w[\phi_0 + \phi_1 x, \sigma^2]$$



Learning algorithm: learn θ from training data **x,y**. E.g.

$$\begin{aligned}\hat{\boldsymbol{\theta}} &= \underset{\boldsymbol{\theta}}{\arg\max}\, Pr(\boldsymbol{\theta}|w_{1\ldots I}, x_{1\ldots I}) \\ &= \underset{\boldsymbol{\theta}}{\arg\max}\, Pr(w_{1\ldots I}|x_{1\ldots I}, \boldsymbol{\theta}) Pr(\boldsymbol{\theta})\end{aligned} \quad = \underset{\boldsymbol{\theta}}{\arg\max} \prod_{i=1}^{I} Pr(w_i|x_i, \boldsymbol{\theta}) Pr(\boldsymbol{\theta}),$$

$$Pr(w|x) = \text{Norm}_w[\phi_0 + \phi_1 x, \sigma^2]$$



Inference algorithm: just evaluate Pr(**w**|**x**) for new data

Type 2: Pr(x|w) - Generative

How to model Pr(x|w)?

1. Choose an appropriate form for Pr(**x**)
2. Make parameters a function of **w**
3. Function takes parameters θ that define its shape

Learning algorithm: learn parameters θ from training data x,w

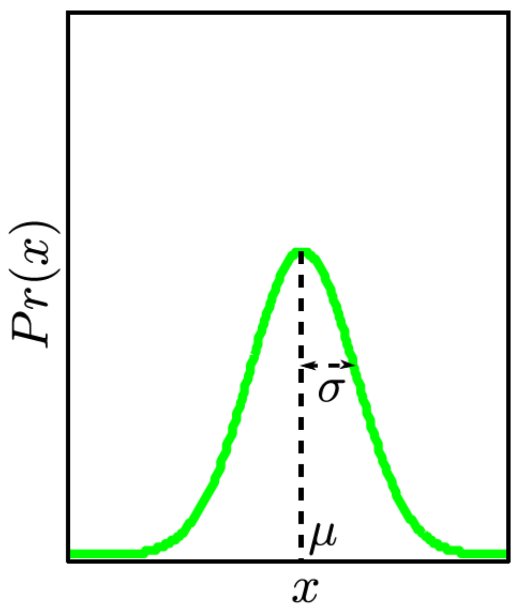
Inference algorithm: Define prior Pr(w) and then compute Pr(w|x) using Bayes' rule

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

Type 2: Pr(x|w) - Generative

How to model Pr(x|w)?

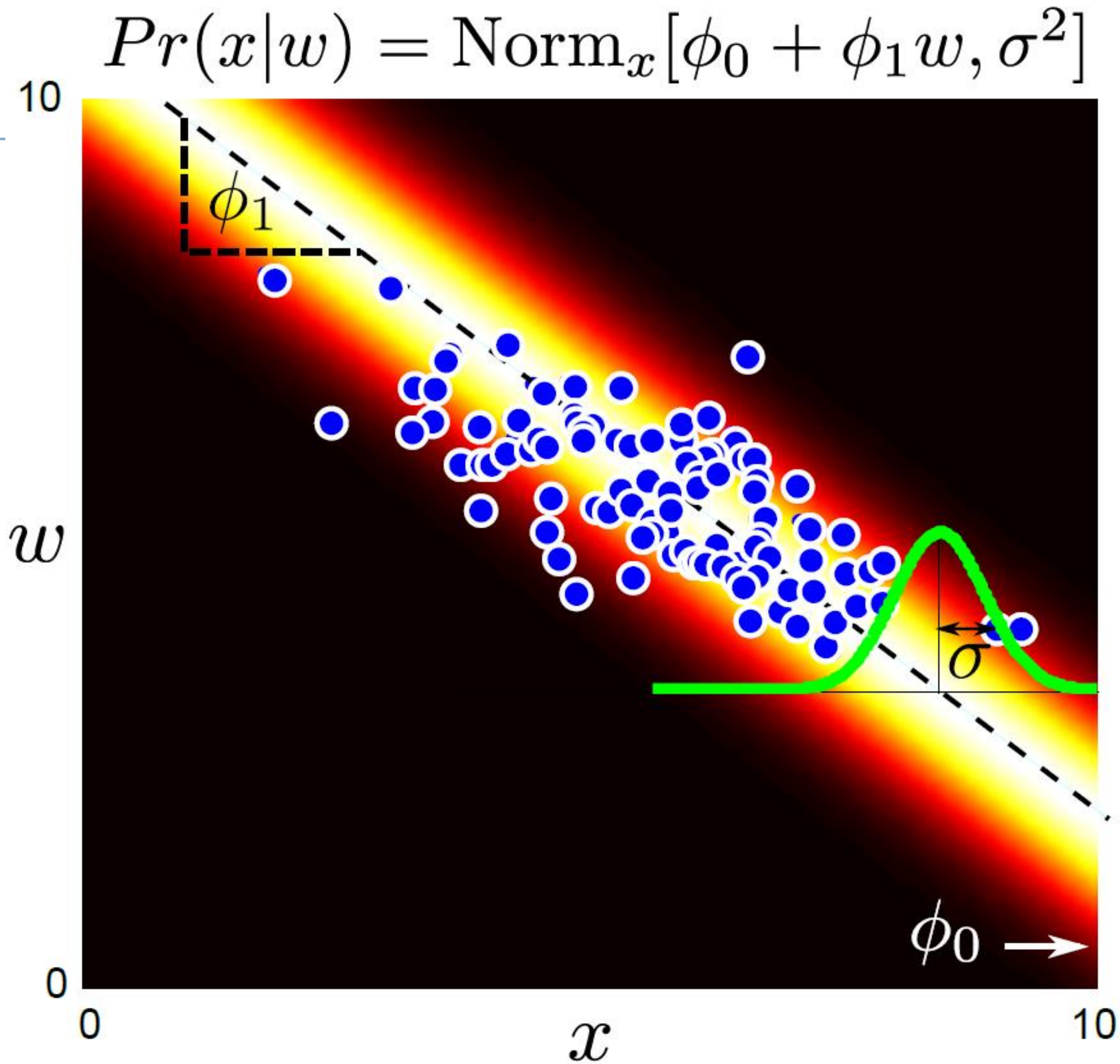
1. Choose an appropriate form for Pr(**x**)
2. Make parameters a function of **w**
3. Function takes parameters **θ** that define its shape



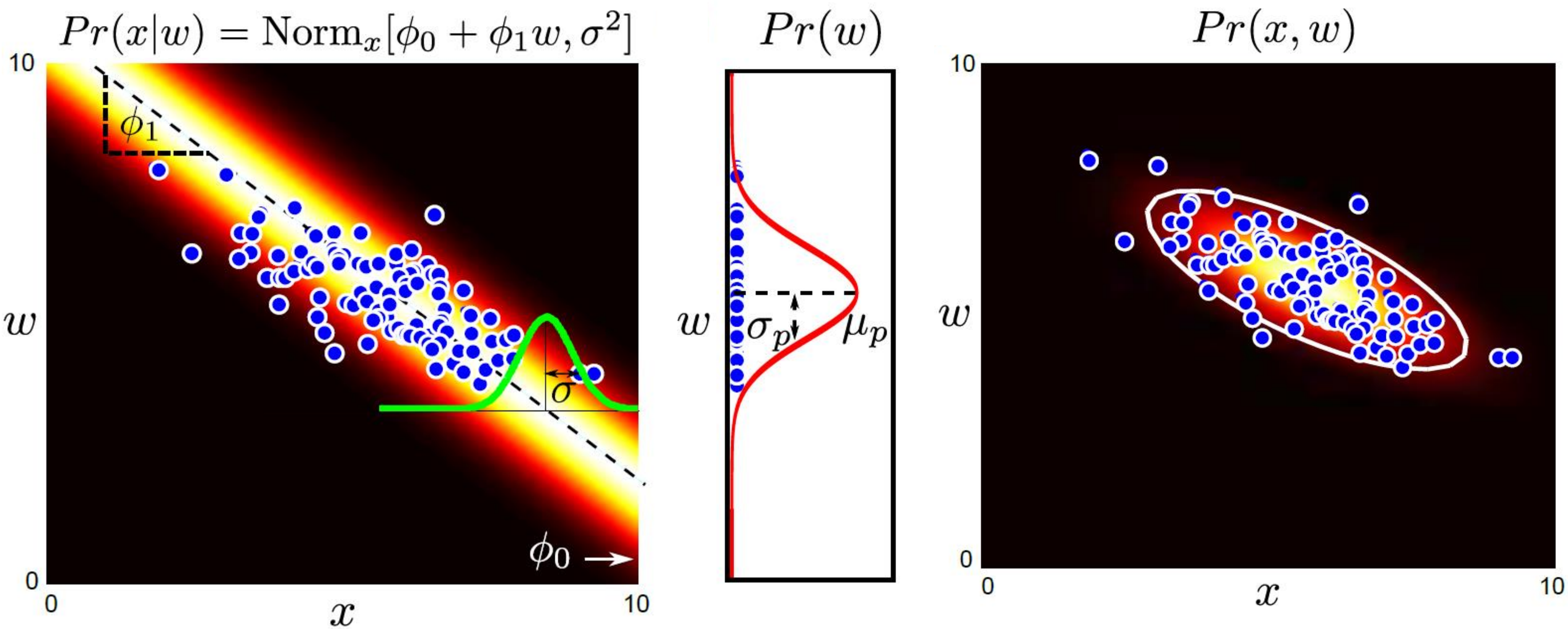
1. Choose normal distribution over x
2. Make mean μ linear function of w (variance constant)

$$Pr(x|w,\boldsymbol{\theta}) = \text{Norm}_x\left[\phi_0 + \phi_1 w, \sigma^2\right]$$

3. Parameter are ϕ_0, ϕ_1, σ^2.

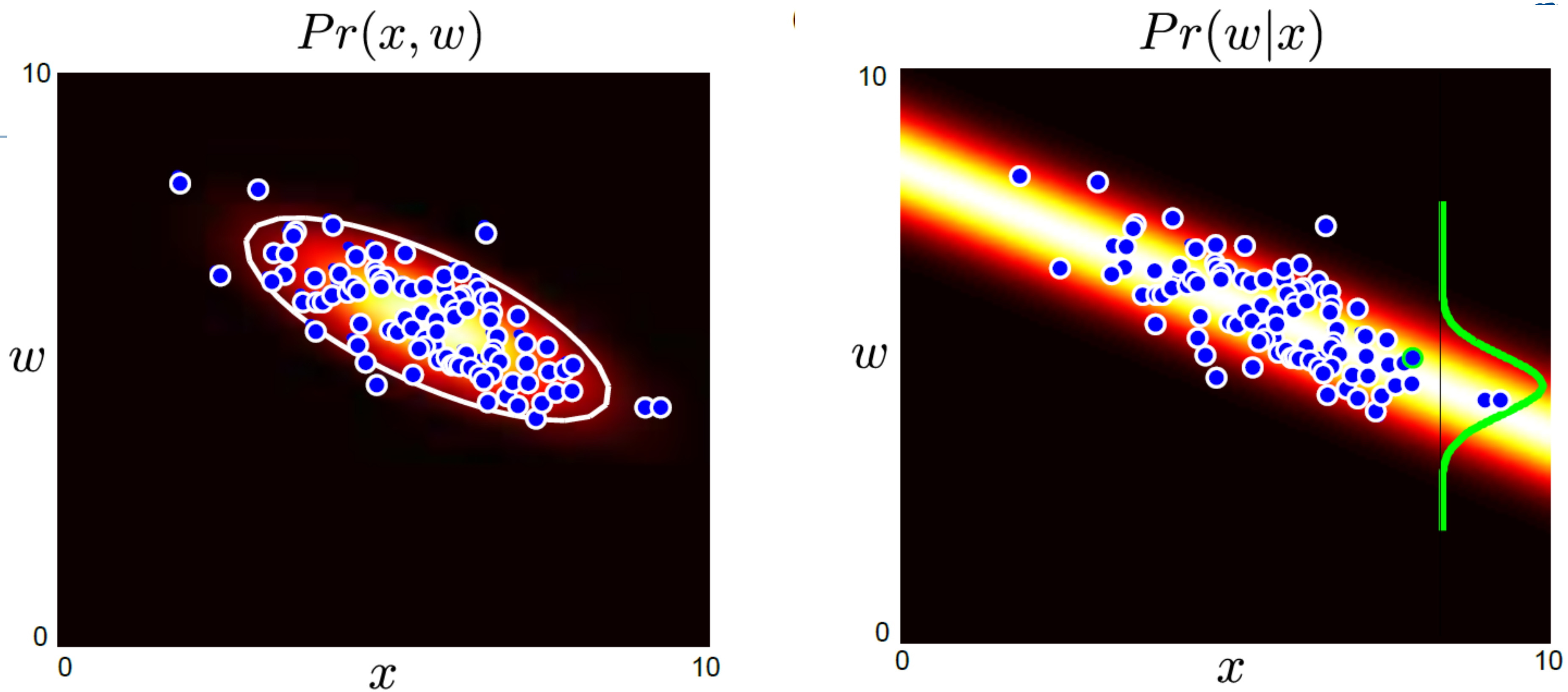


Learning algorithm: learn θ from training data **x**,**w**. e.g. MAP



Pr(x|w) x Pr(w) = Pr(x,w)

Can get back to joint probability Pr(x,y)



Inference algorithm: compute Pr(**w**|**x**) using Bayes rule

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

Structure

Computer vision models

- Three types of model

Worked example 1: Regression

Worked example 2: Classification

Which type should we choose?

Applications

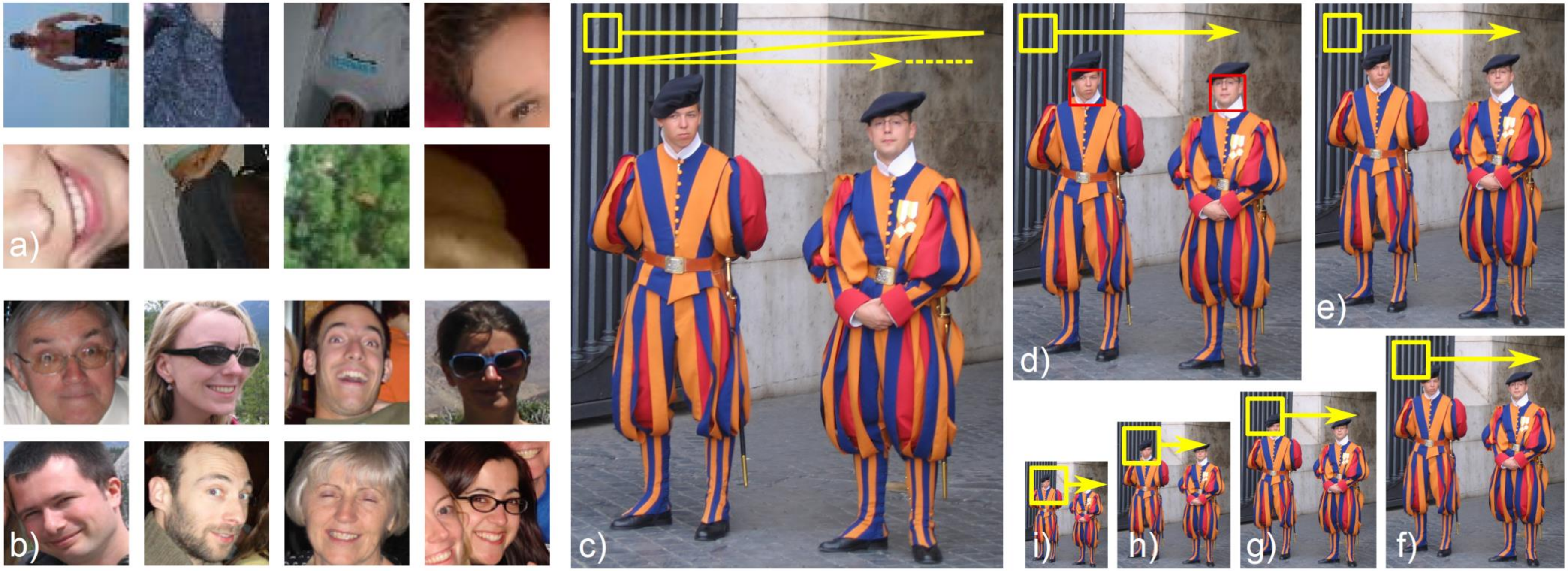
Worked example 2: Classification

Consider simple case where

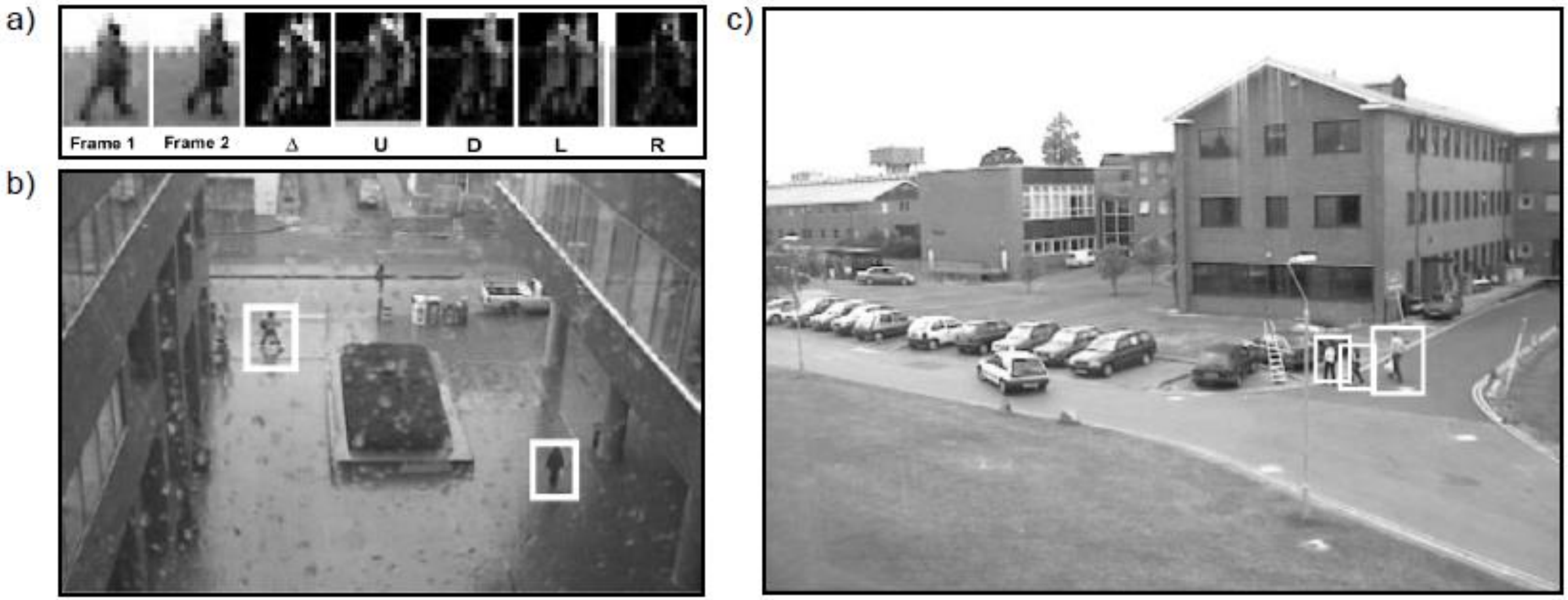
- we make a univariate continuous measurement x
- use this to predict a discrete binary $w \in \{0, 1\}$

(classification as world state is discrete)

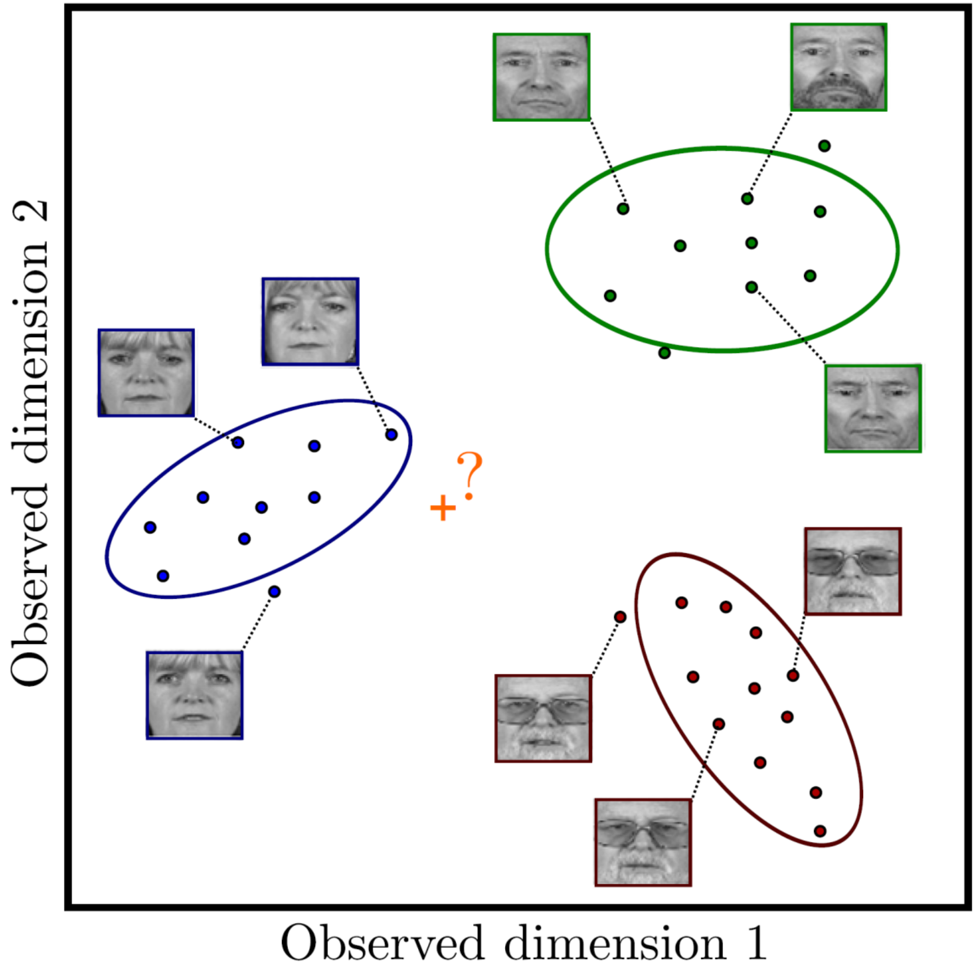
Classification Example 1: Face Detection



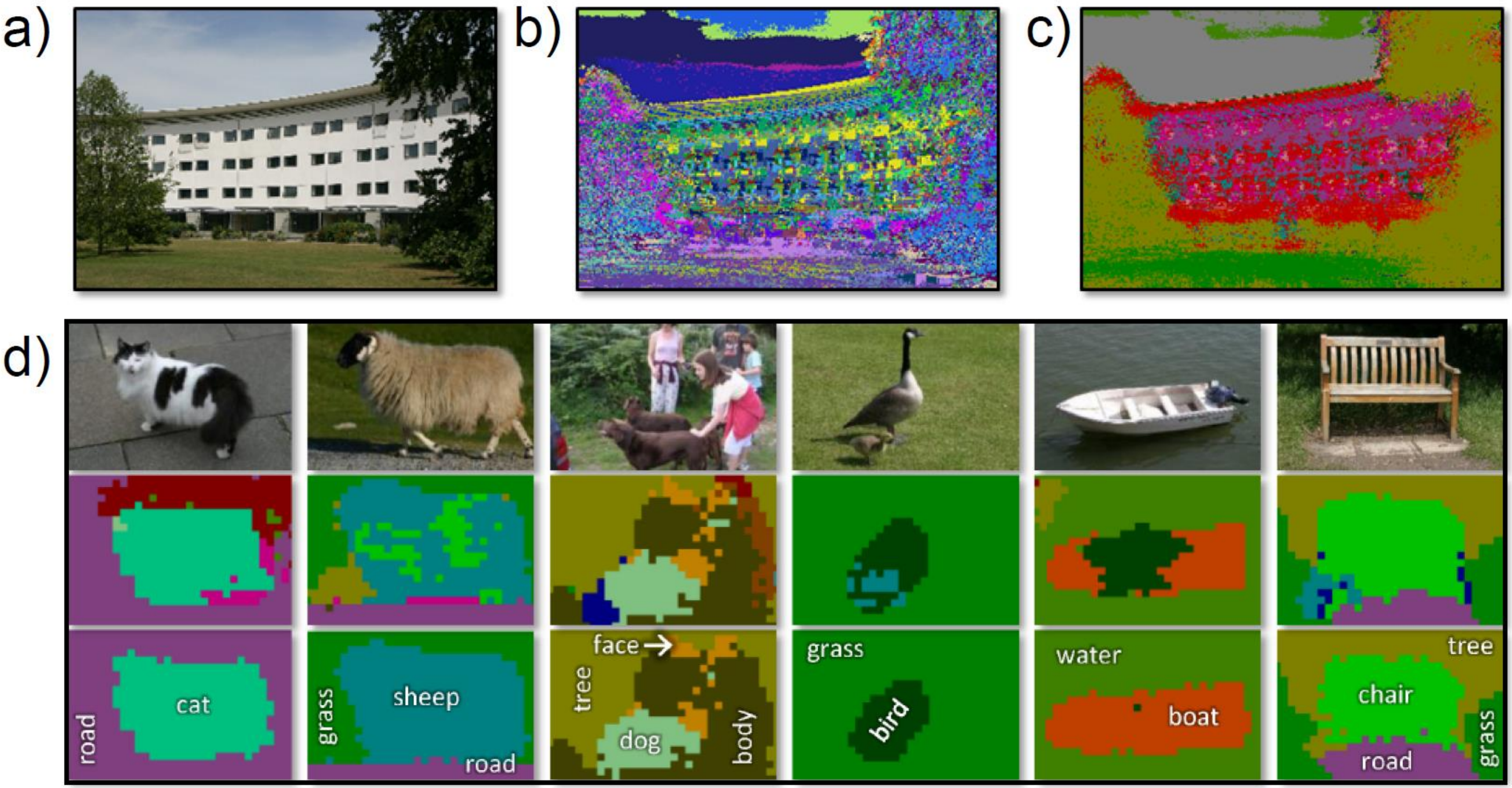
Classification Example 2: Pedestrian Detection



Classification Example 3: Face Recognition



Classification Example 4: Semantic Segmentation



Worked example 2: Classification

Consider simple case where

- we make a univariate continuous measurement x
- use this to predict a discrete binary world w $\in \{0, 1\}$

(classification as world state is discrete)

Type 1: Model Pr(w|x) - Discriminative

How to model Pr(w|x)?

- Choose an appropriate form for Pr(**w**)
- Make parameters a function of **x**
- Function takes parameters θ that define its shape

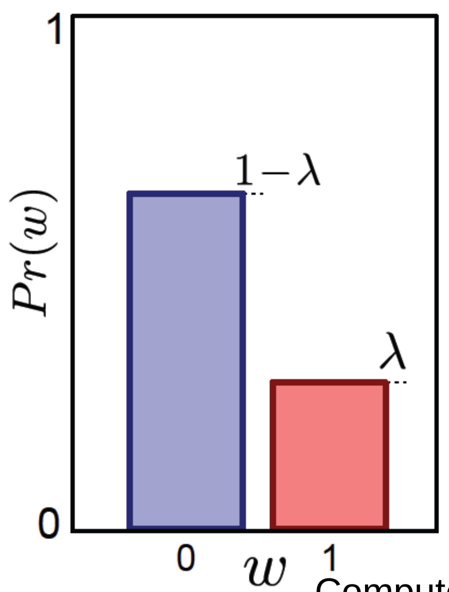
Learning algorithm: learn parameters θ from training data x,w

Inference algorithm: just evaluate Pr(w|x)

Type 1: Model Pr(w|x) - Discriminative

How to model Pr(w|x)?

1. Choose an appropriate form for Pr(**w**)
2. Make parameters a function of **x**
3. Function takes parameters θ that define its shape

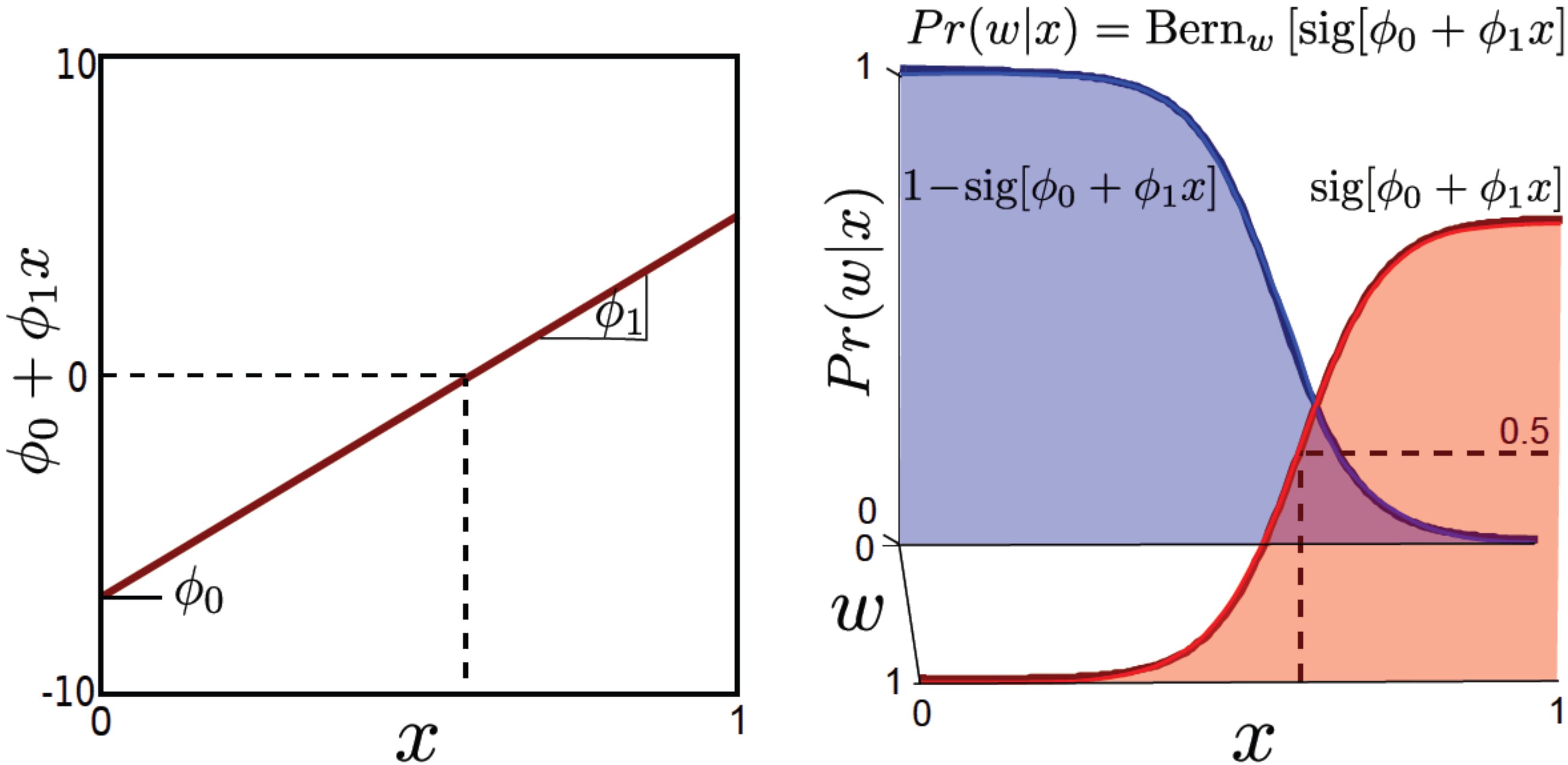


1. Choose Bernoulli dist. for Pr(**w**)
2. Make parameters a function of **x**

$$Pr(w|x) = \text{Bern}_w\left[\text{sig}[\phi_0 + \phi_1 x]\right]$$
$$= \text{Bern}_w\left[\frac{1}{1+\exp[-\phi_0 - \phi_1 x]}\right]$$

3. Function takes parameters ϕ_0 and ϕ_1

This model is called *logistic regression.*



Two parameters $\boldsymbol{\theta} = \{\phi_0, \phi_1\}$

Learning by standard methods (ML,MAP, Bayesian)

Inference: Just evaluate Pr(w|x)

Type 2: Pr(x|w) - Generative

How to model Pr(x|w)?

1. Choose an appropriate form for Pr(**x**)
2. Make parameters a function of **w**
3. Function takes parameters θ that define its shape

Learning algorithm: learn parameters θ from training data x,w

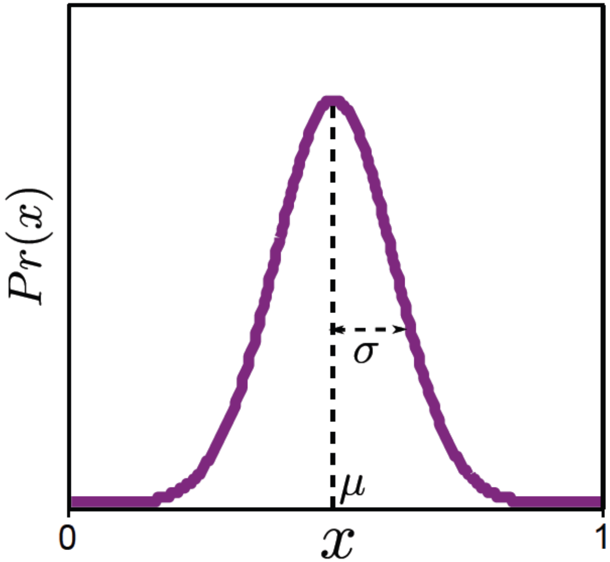
Inference algorithm: Define prior Pr(w) and then compute Pr(w|x) using Bayes' rule

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

Type 2: Pr(x|w) - Generative

How to model Pr(x|w)?

1. Choose an appropriate form for Pr(**x**)
2. Make parameters a function of **w**
3. Function takes parameters θ that define its shape

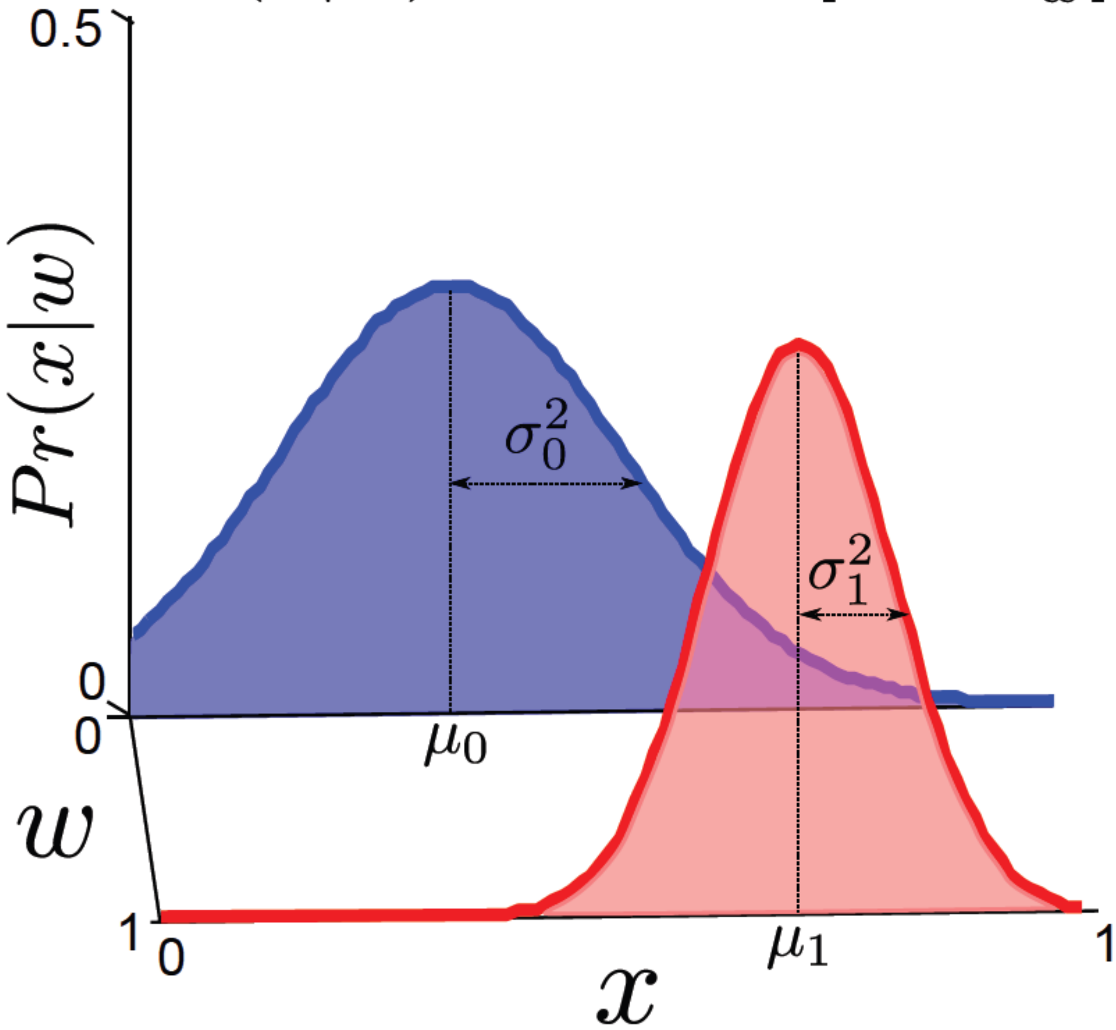


1. Choose a Gaussian distribution for Pr(**x**)
2. Make parameters a function of discrete binary **w**

$$Pr(x|w) = \text{Norm}_x[\mu_w, \sigma^2_w]$$

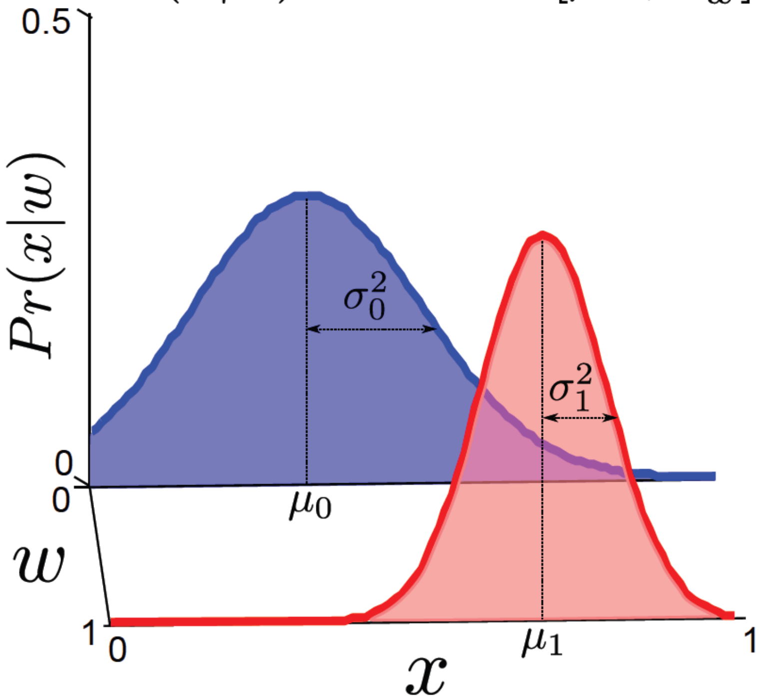
3. Function takes parameters μ_0, μ_1, σ^2_0, σ^2_1 that define its shape

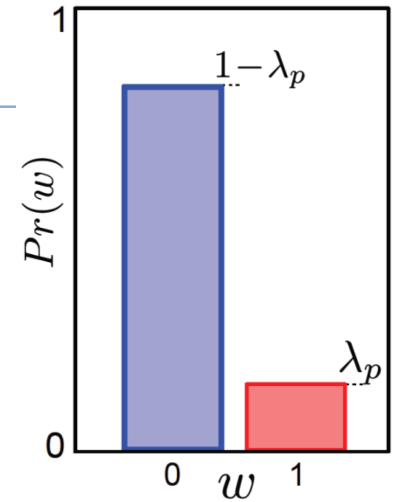
$$Pr(x|w) = \mathrm{Norm}_x[\mu_w, \sigma^2_w]$$

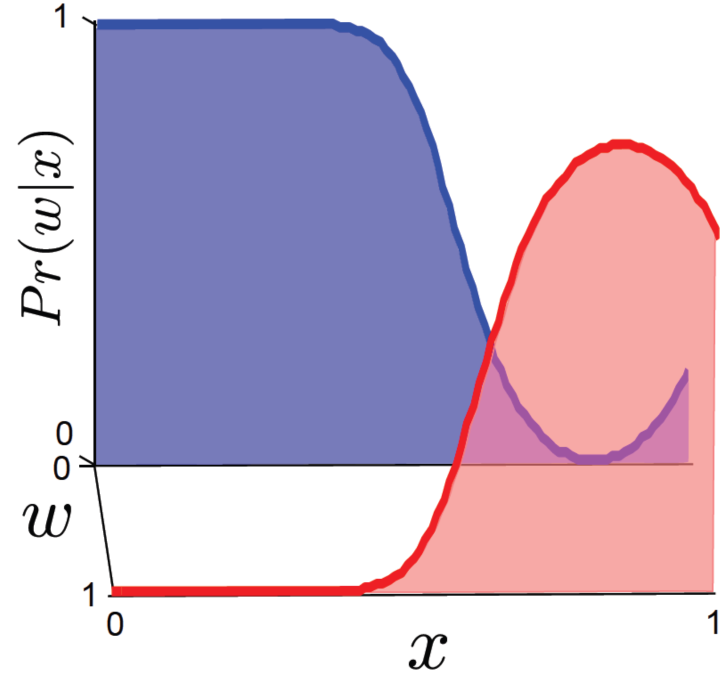


Learn parameters μ_0, μ_1, $\sigma^2{}_0$, $\sigma^2{}_1$ that define its shape

$$Pr(x|w) = \text{Norm}_x[\mu_w, \sigma_w^2]$$







Inference algorithm: Define prior Pr(**w**) and then compute Pr(**w**|**x**) using Bayes' rule

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

Structure

Computer vision models

- Three types of model

Worked example 1: Regression

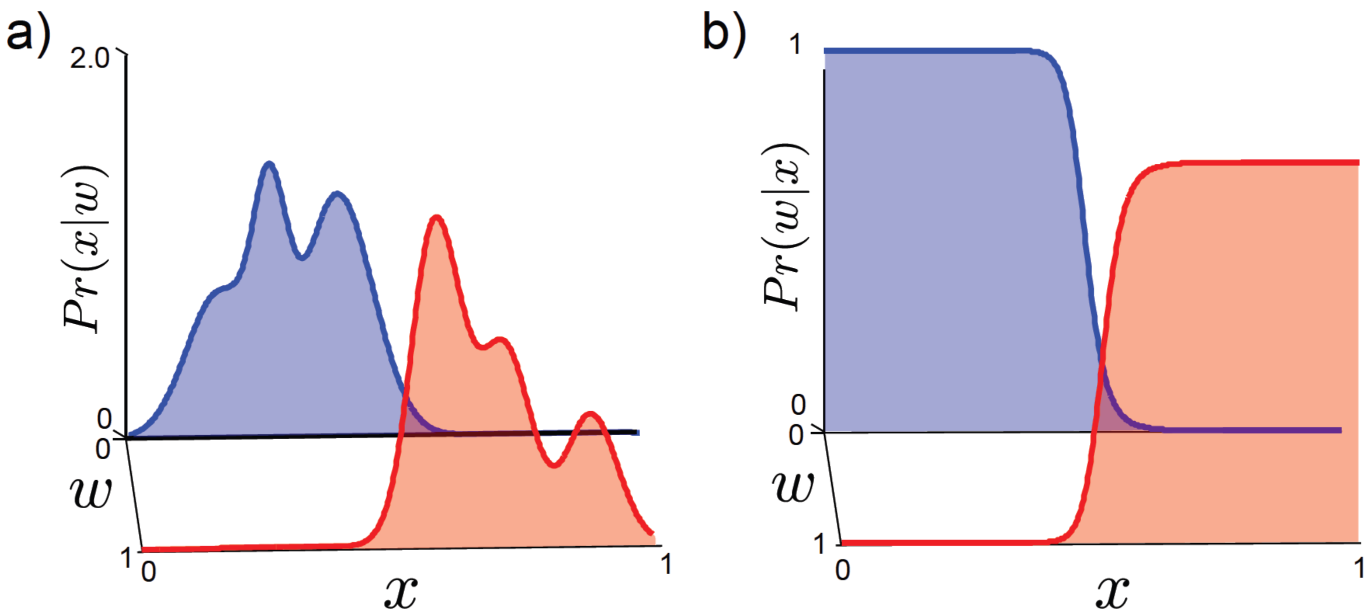
Worked example 2: Classification

Which type should we choose?

Applications

Which type of model to use?

1. **Generative methods model data – costly and many aspects of data may have no influence on world state**



Which type of model to use?

2. **Inference simple in discriminative models**
3. **Data really is generated from world – generative matches this**
4. **If missing data, then generative preferred**
5. **Generative allows imposition of prior knowledge specified by user**

Structure

Computer vision models

- Three types of model

Worked example 1: Regression

Worked example 2: Classification

Which type should we choose?

Applications

Application: Skin Detection

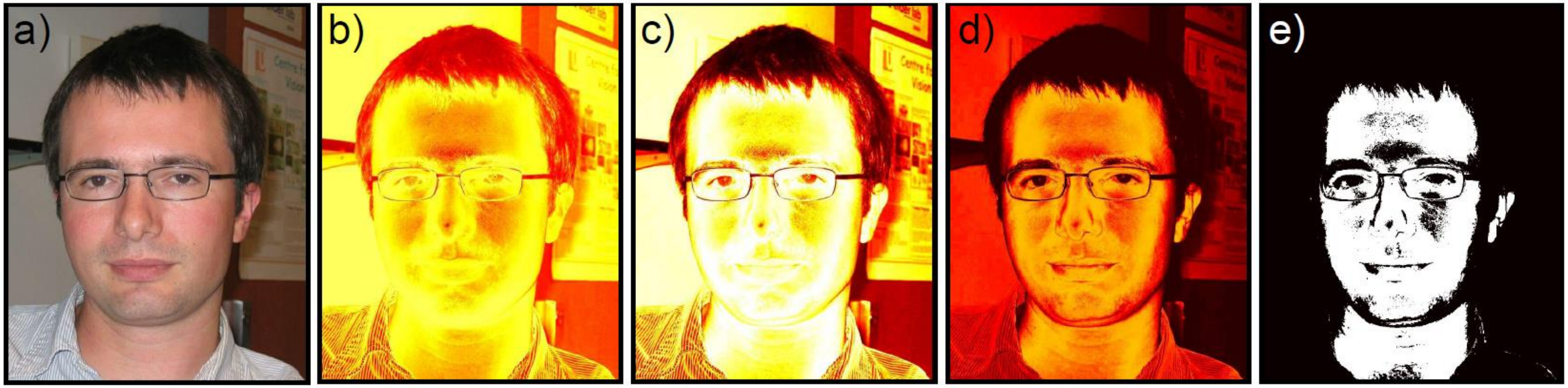
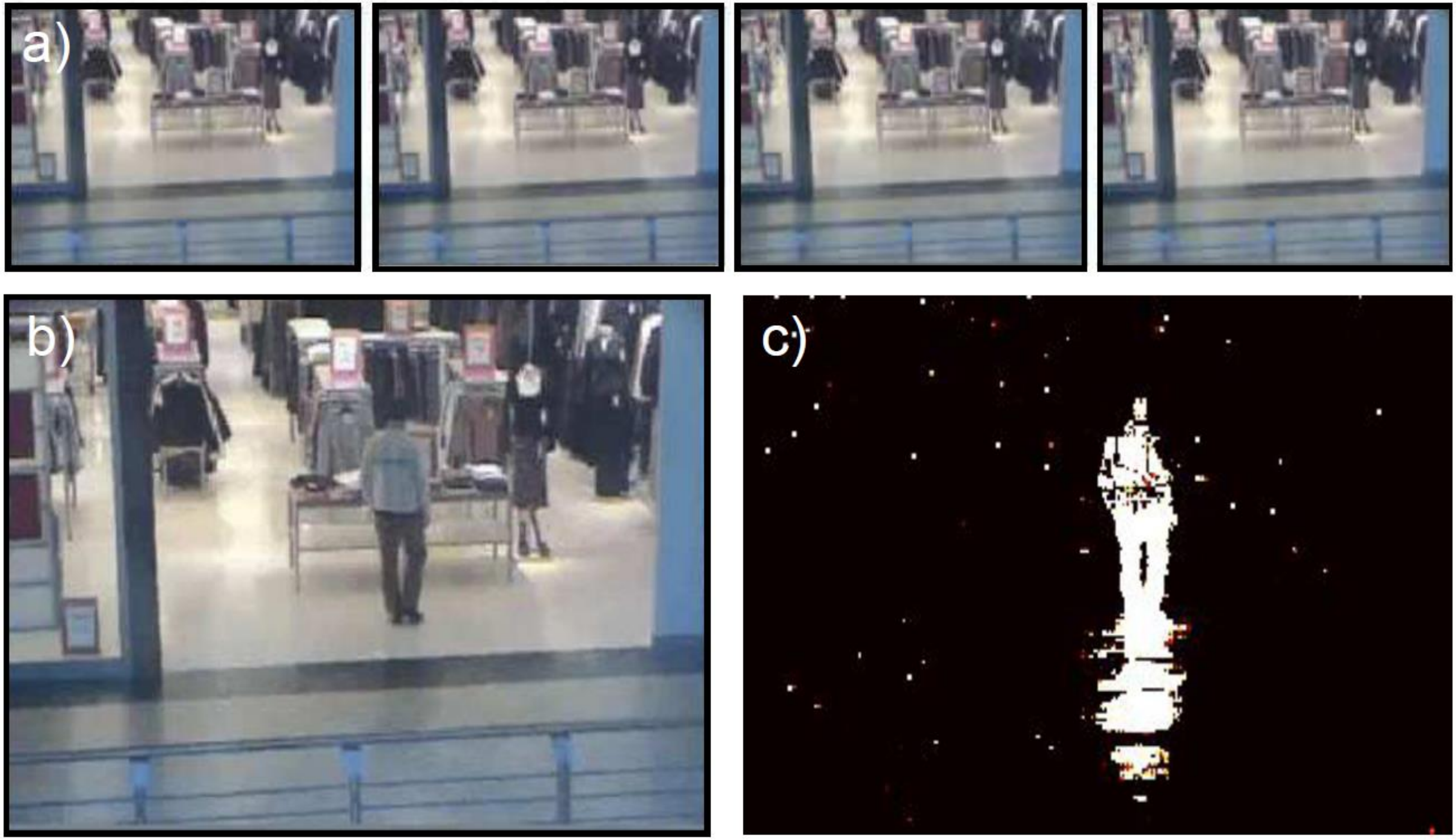
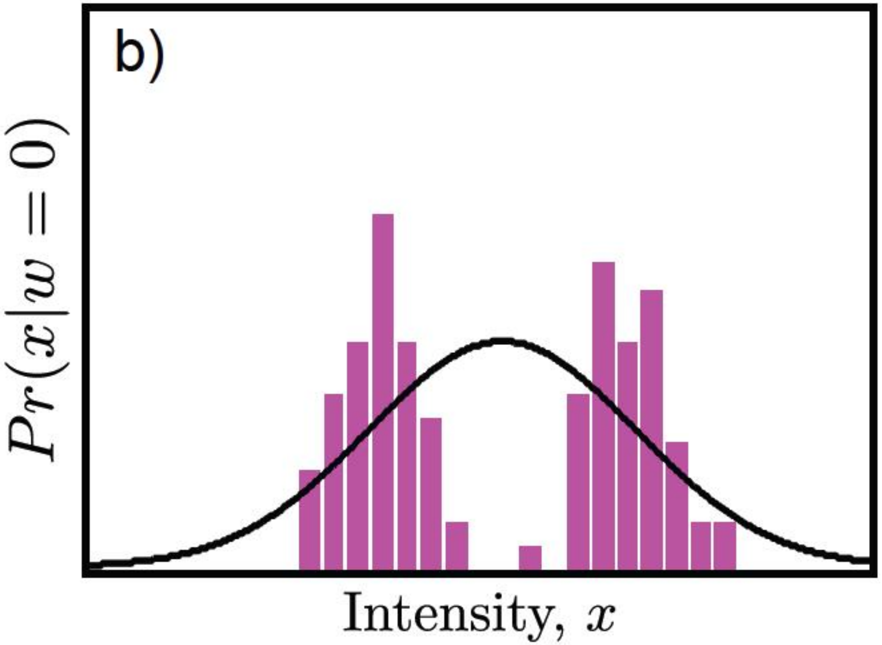


Figure 6.7 Skin detection. For each pixel we aim to infer a label $w \in \{0, 1\}$ denoting the absence or presence of skin based on the RGB triple $\mathbf{x}$. Here we modeled the class conditional density functions $Pr(\mathbf{x}|w)$ as normal distributions. a) Original image. b) Log likelihood (log of data assessed under class-conditional density function) for non-skin. c) Log likelihood for skin. d) Posterior probability of belonging to skin class. e) Thresholded posterior probability $Pr(w|\mathbf{x}) > 0.5$ gives estimate of w.

Application: Background subtraction

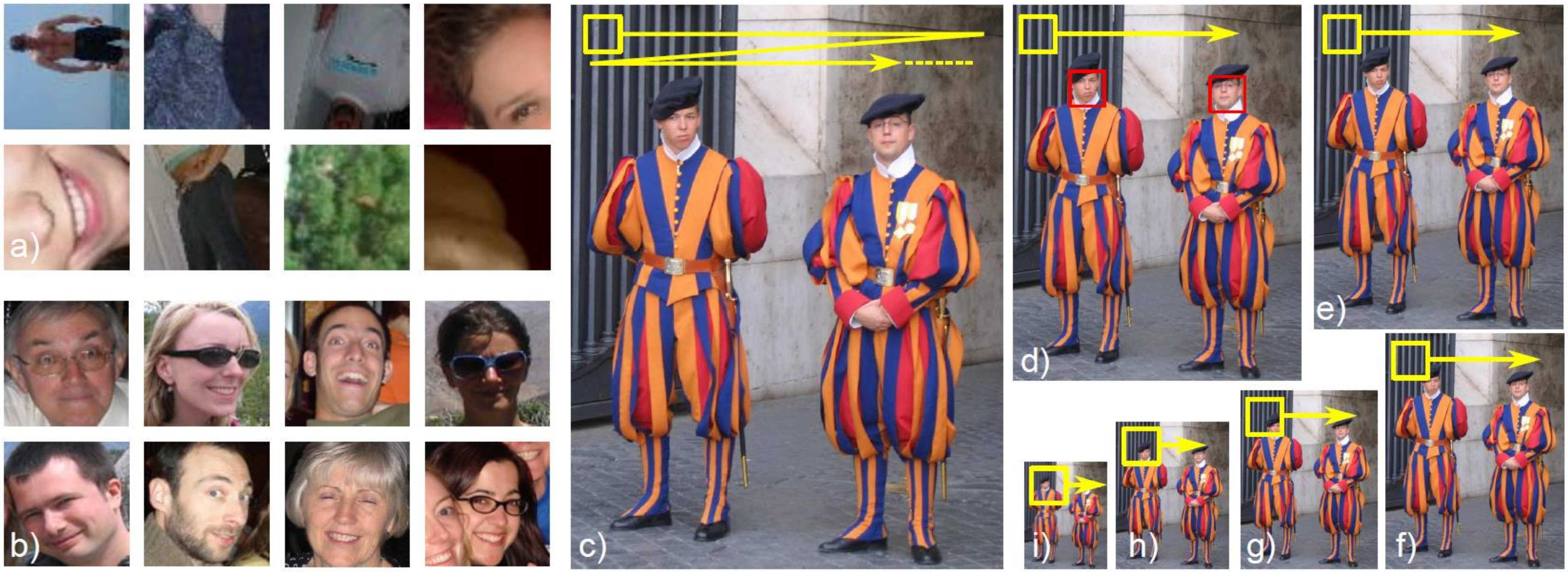


Application: Background subtraction



But consider this scene in which the foliage is blowing in the wind. A normal distribution is not good enough! Need a way to make more complex distributions

Face Detection



Type 3: Pr(x|w) - Generative

How to model Pr(x|w)?

- Choose an appropriate form for Pr(**x**)
- Make parameters a function of **w**
- Function takes parameters θ that define its shape

Learning algorithm: learn parameters θ from training data x,w

Inference algorithm: Define prior Pr(w) and then compute Pr(w|x) using Bayes' rule

$$Pr(w=1|\mathbf{x}) = \frac{Pr(\mathbf{x}|w=1)Pr(w=1)}{\sum_{k=0}^{1} Pr(\mathbf{x}|w=k)Pr(w=k)}$$

Classification Model

$$Pr(\mathbf{x}|w) = \text{Norm}_{\mathbf{x}}[\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w]$$

Or writing in terms of class conditional density functions

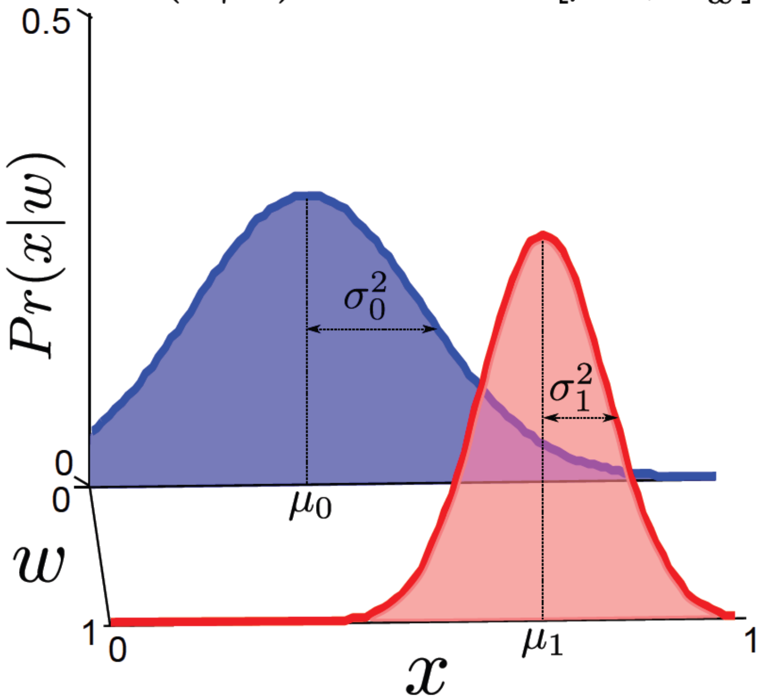
$$\begin{aligned} Pr(\mathbf{x}|w=0) &= \text{Norm}_{\mathbf{x}}[\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0] \\ Pr(\mathbf{x}|w=1) &= \text{Norm}_{\mathbf{x}}[\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1] \end{aligned}$$

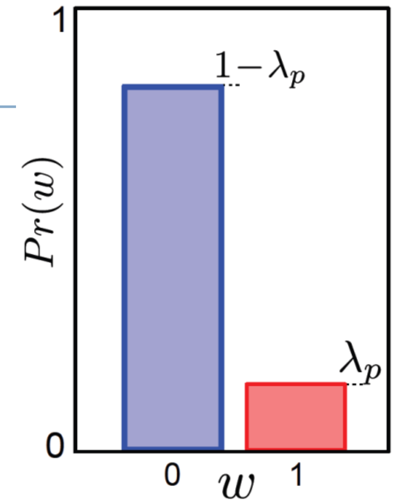
Parameters $\boldsymbol{\mu}_0$, $\boldsymbol{\Sigma}_0$ learnt just from data S_0 where w=0

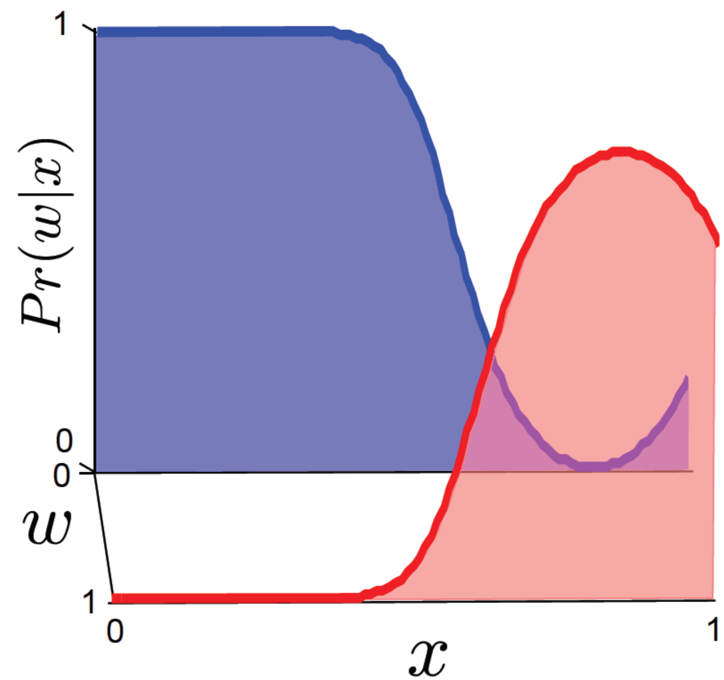
$$\begin{aligned} \hat{\boldsymbol{\mu}}_0, \hat{\boldsymbol{\Sigma}}_0 &= \underset{\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0}{\operatorname{argmax}} \left[\prod_{i \in \mathcal{S}_0} Pr(\mathbf{x}_i | \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) \right] \\ &= \underset{\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0}{\operatorname{argmax}} \left[\prod_{i \in \mathcal{S}_0} \text{Norm}_{\mathbf{x}_i}[\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0] \right] \end{aligned}$$

Similarly, parameters μ_1, Σ_1 learnt just from data S_1 where w=1

$$Pr(x|w) = \text{Norm}_x[\mu_w, \sigma_w^2]$$







Inference algorithm: Define prior Pr(**w**) and then compute Pr(**w**|**x**) using Bayes' rule

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

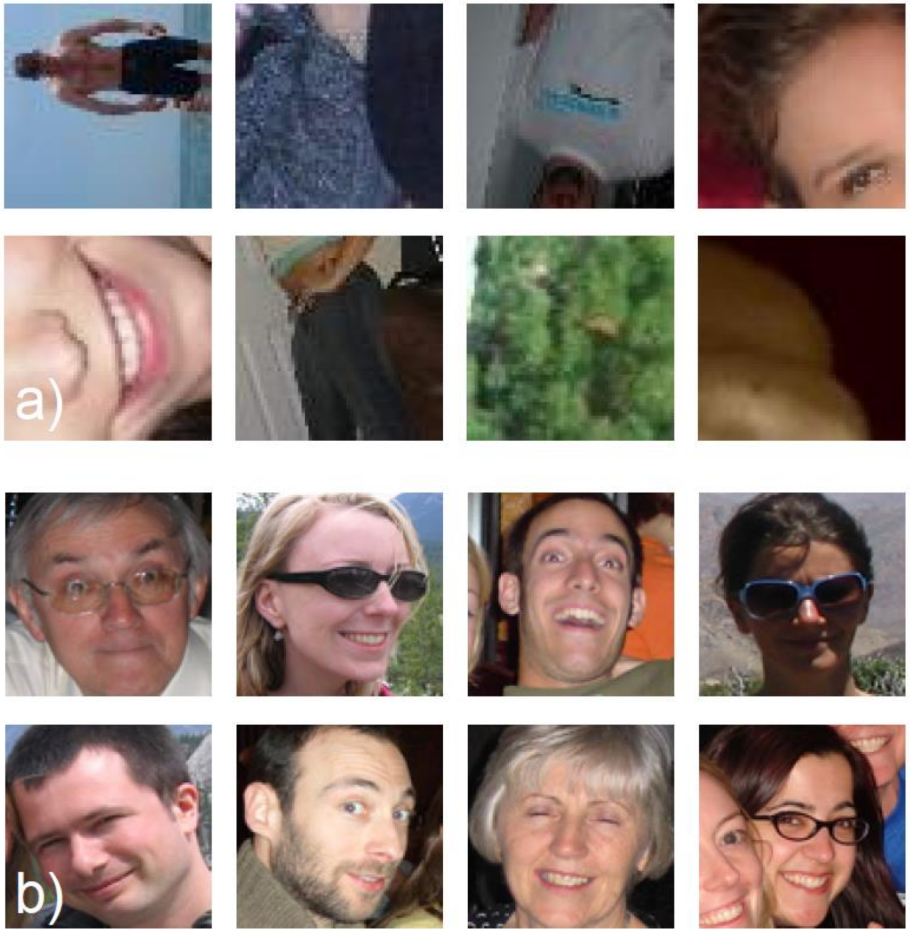
Experiment

1000 non-faces
1000 faces

60x60x3 Images =10800 x1 vectors

Equal priors Pr(y=1)=Pr(y=0) = 0.5

75% performance on test set. Not very good!



Results (diagonal covariance)

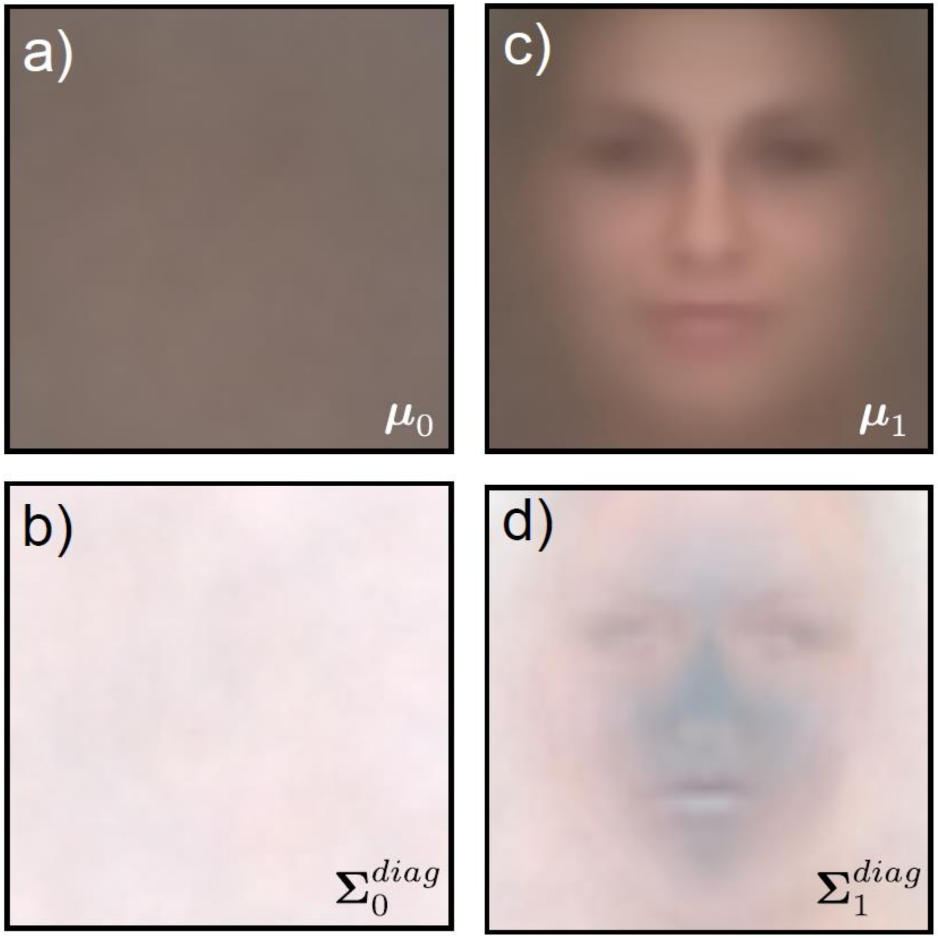
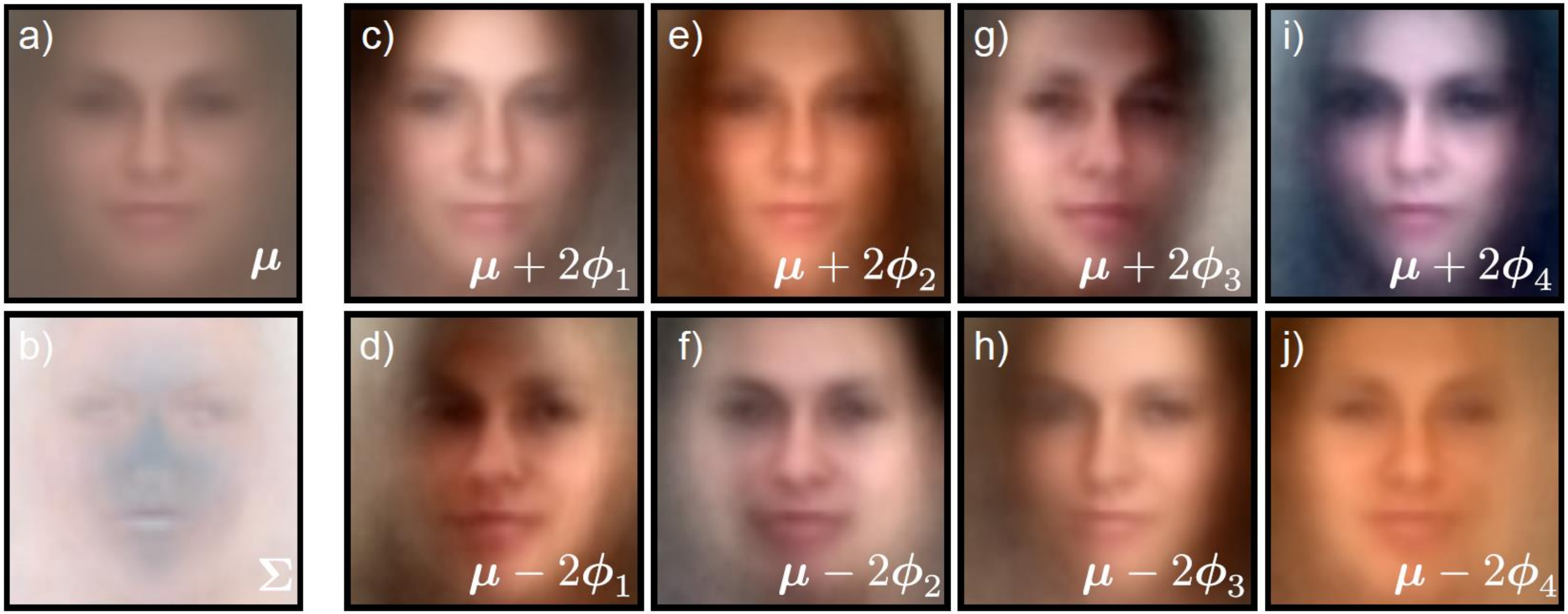


Figure 7.2 Class conditional density functions for normal model with diagonal covariance. Maximum likelihood fits based on 1000 training examples per class. a) Mean for background data $\boldsymbol{\mu}_0$ (reshaped from 10800×1 vector to 60×60 RGB image). b) Reshaped square root of diagonal covariance for background data $\boldsymbol{\Sigma}_0$. c) Mean for face data $\boldsymbol{\mu}_1$ d) Covariance for face data $\boldsymbol{\Sigma}_1$. The background model has little structure: the mean is uniform and the variance is high everywhere. The mean of the face model clearly captures class-specific information. The covariance of the face is larger at the edges of the image which usually contain hair or background.

Means of face/non-face model

Classification → 84% (9% improvement!)

Face model



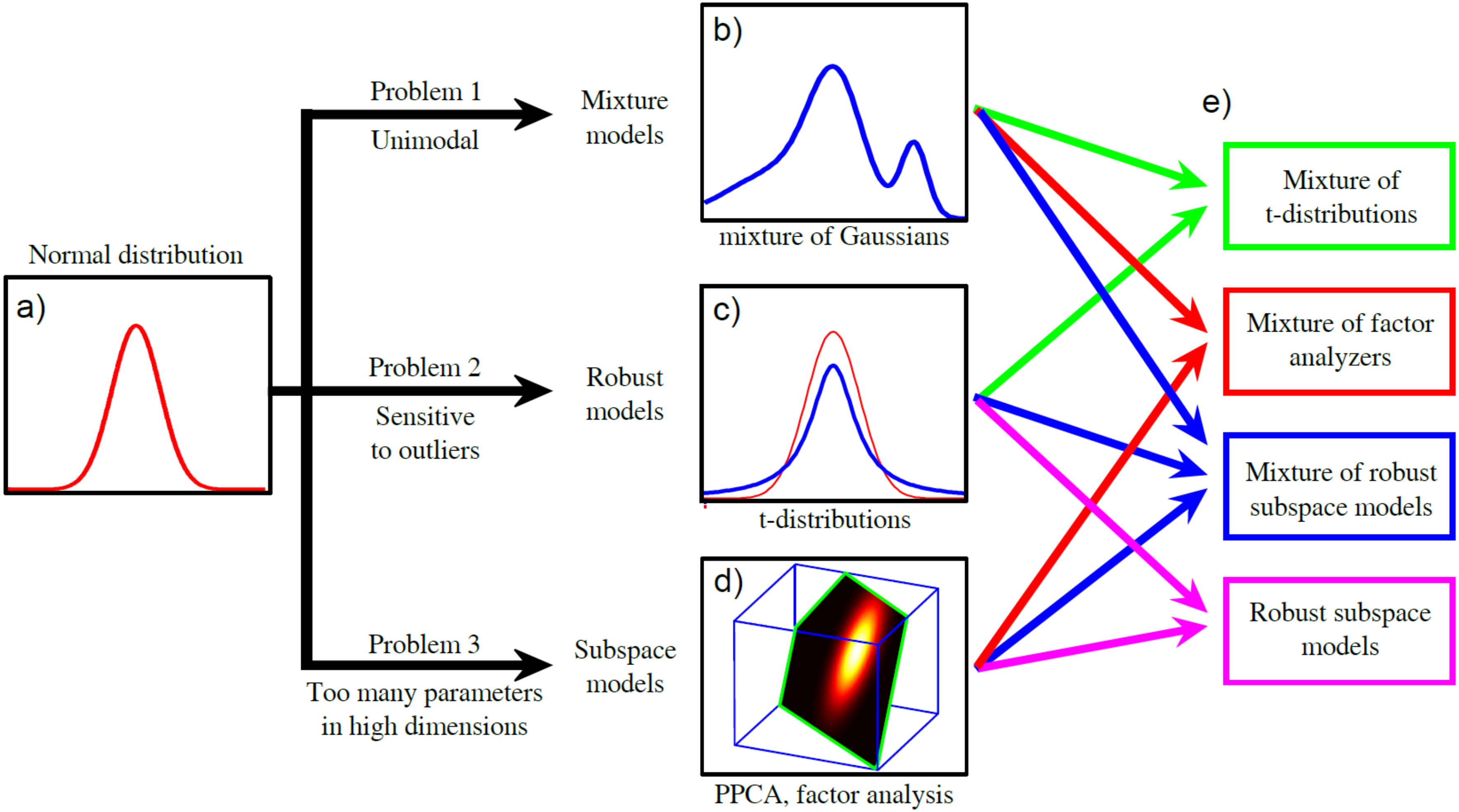
a)
μ
b)
Σ
c)
$\mu + 2\phi_1$
d)
$\mu - 2\phi_1$
e)
$\mu + 2\phi_2$
f)
$\mu - 2\phi_2$
g)
$\mu + 2\phi_3$
h)
$\mu - 2\phi_3$
i)
$\mu + 2\phi_4$
j)
$\mu - 2\phi_4$

Sampling from 10 parameter model

To generate:

- Choose factor loadings, $\mathbf{h}_i$ from standard normal distribution
- Multiply by factors, $\boldsymbol{\Phi}$
- Add mean, $\boldsymbol{\mu}$
- (should add random noise component ε_i w/ diagonal cov $\boldsymbol{\Sigma}$)

Probability Density Models



Conclusion

- To do computer vision we build a model relating the image data **x** to the world state that we wish to estimate **w**

- Three types of model
  - Model Pr(w|x) -- discriminative
  - Model Pr(w|x) – generative

Backup