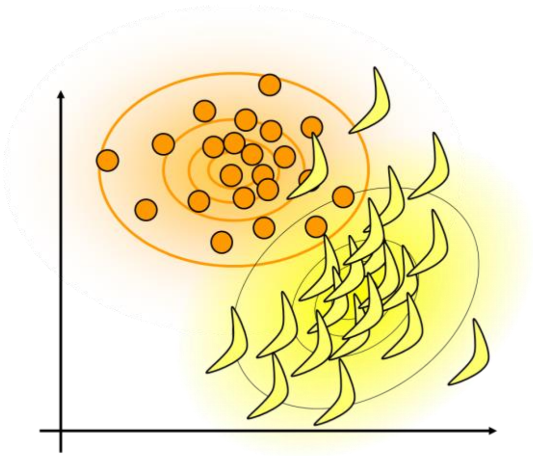
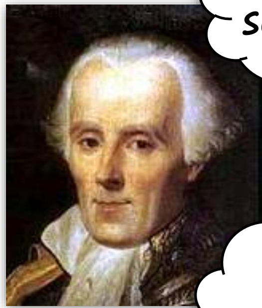
Modelling 2

STATISTICAL DATA MODELLING





Chapter 4

Statistics and Machine Learning

Michael Wand · Institut für Informatik, JGU Mainz · michael.wand@uni-mainz.de

Recap: Previous Video

Probability Theory

- Mathematical Axioms
  - Basis for all modeling of uncertainty
- Frequentist Interpretation / Application
  - Repeatable experiments
- Bayesian Interpretation / Application
  - General believes
  - Might be subjective

Hertzman's Principle #1

Laplace (1814)

> "*Probability theory is nothing more than common sense reduced to calculation*"

[image: Wikipedia]

Pierre-Simon Laplace
(1749–1827)

from Hertzman's course notes: [http://www.dgp.toronto.edu/~hertzman/ibl2004/bayes2004.pdf]

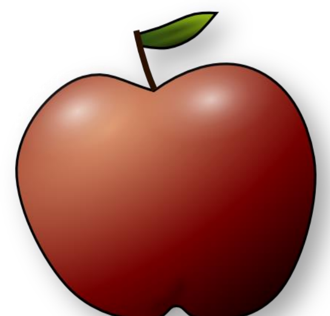
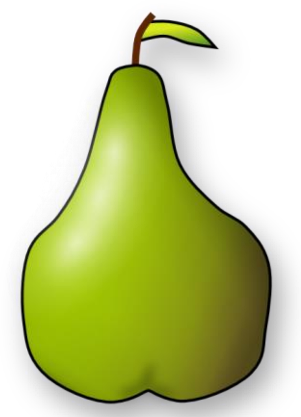
Video #04

Statistics & Machine Learning

- **Machine Learning Basics**
- **Bayesian Inference for ML**
- **Learning & Inference**

Machine Learning
& Bayesian Statistics



Machine Learning & Statistics

What is machine learning?

- Derive solution from examples (data)
  - "Data driven" computer science
  - Given a task and examples
- Statistical ML: Use statistical techniques
  - "Real world" data such as photos, sound, etc., rather than curated data bases
  - Algorithmic induction

Machine learning

Typical Tasks

- **Regression**

  learn function $f: X \to Y$

- **Classification**

  special case – B is a set of categories

- **Density reconstruction**

  learn probability distribution $p(\mathbf{x}), \quad \mathbf{x} \in X$

Machine learning

Typical Tasks

- **Compression / simplification / structure discovery**
  - Dimensionality reduction
  - Clustering
  - Latent (unobserved) variable discovery
  - …and the similar
- **Control**
  - Learn decision making
    - Steer some agent, or self-driving car
    - Play chess, GO, Robo-Soccer
  - Several actions, long term consequences
- There are probably more

Training Data

How / which data is provided?

- Supervised learning
  - Full "example solutions"
  - *Example:* Regression from pairs $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1..n}$
- Unsupervised learning
  - Unannotated data, infer solution from structure
  - *Example:* Density reconstruction from points $\{\mathbf{x}_i\}_{i=1..n}$
- Semi-supervised learning
  - Only some examples are "full solutions"
  - *Ex.:* Classification from $\{\mathbf{x}_i\}_{i=1..n}$ and $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1..m}$, usually $m \ll n$
- Reinforcement learning:
  - Qualitative feedback, only after a while

Statistical Approach

Meta-Algorithm

- Obtain training data
- Fit probabilistic model to the data
- Use probabilistic model to solve problem
  - Inferring solutions: Minimize risk of errors / loss

Statistical Approach

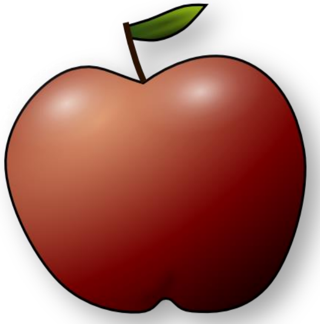
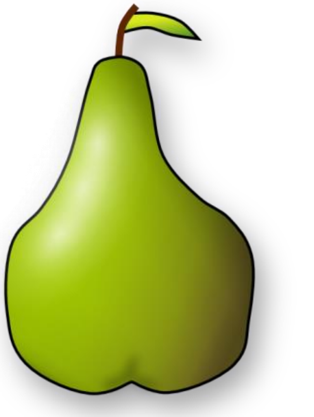
Goals

- Objective: Generalizability
  - Learned model should work on *non-training data*
  - of the same statistics as the training data
- Usual approach
  - Practical objective: “Fit model well to training data”
  - Control for “overfitting” (being “too specific”)

Machine Learning
& Bayesian Statistics

Example: **Classification**



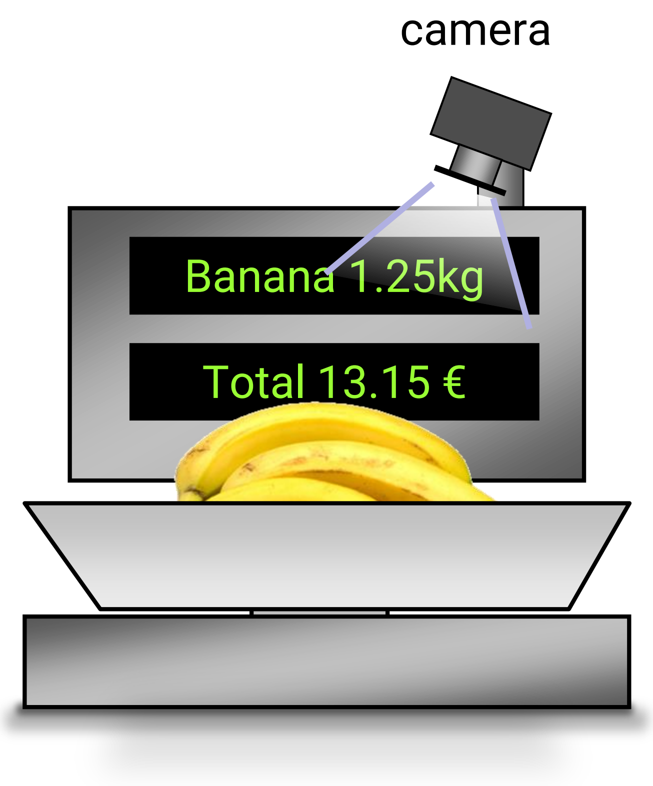
Example Application

Machine Learning Example

- Classification

Application Example

- Automatic scales at supermarket
- Detect type of fruit using a camera



Learning Probabilities

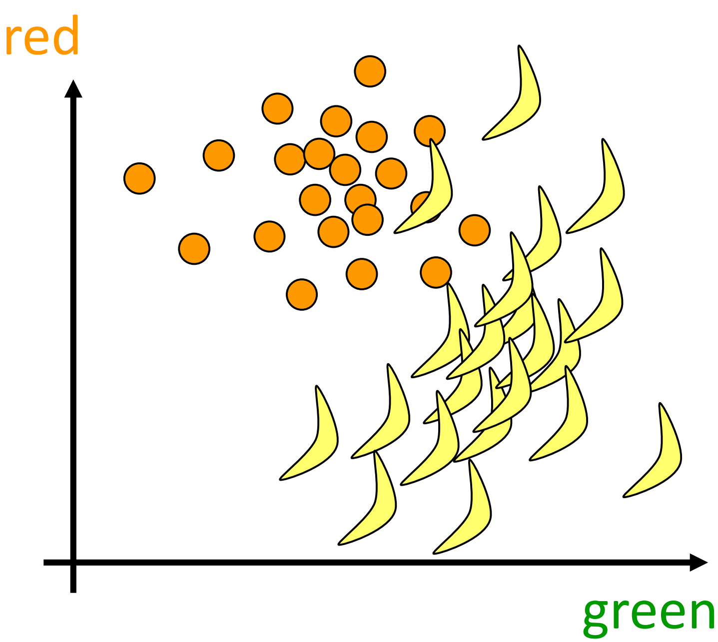
Toy Example

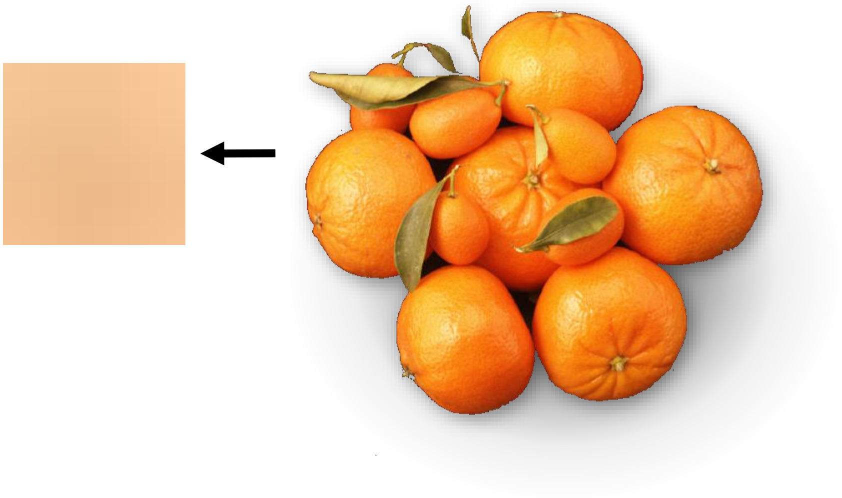
- Distinguish pictures of **oranges** and **bananas**
- 100 training pictures each
- Find rule to distinguish pictures

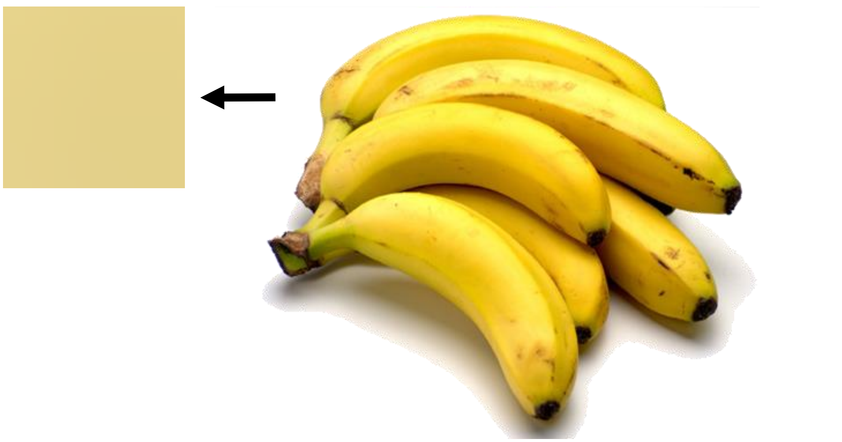
Learning Probabilities

Very simple approach

- Compute average color
- Learn distribution



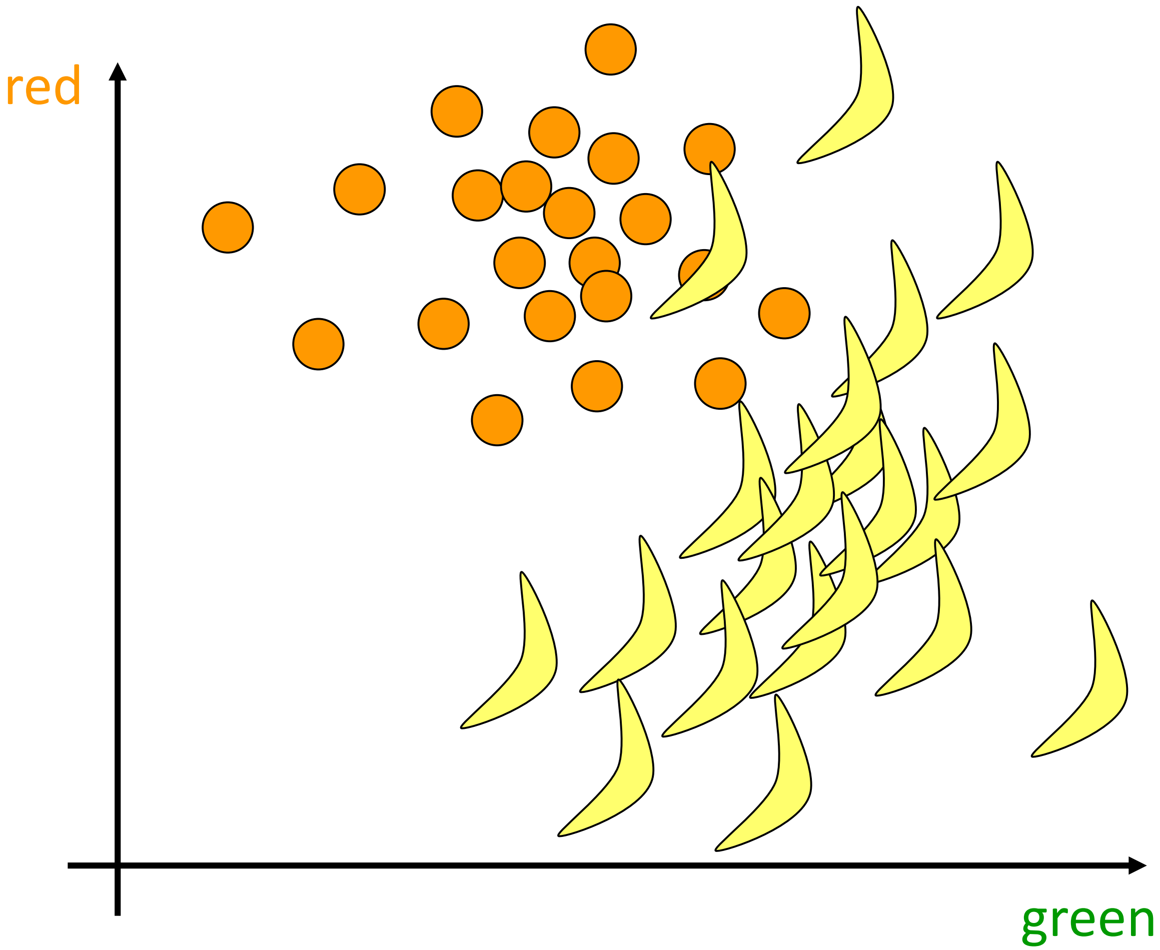




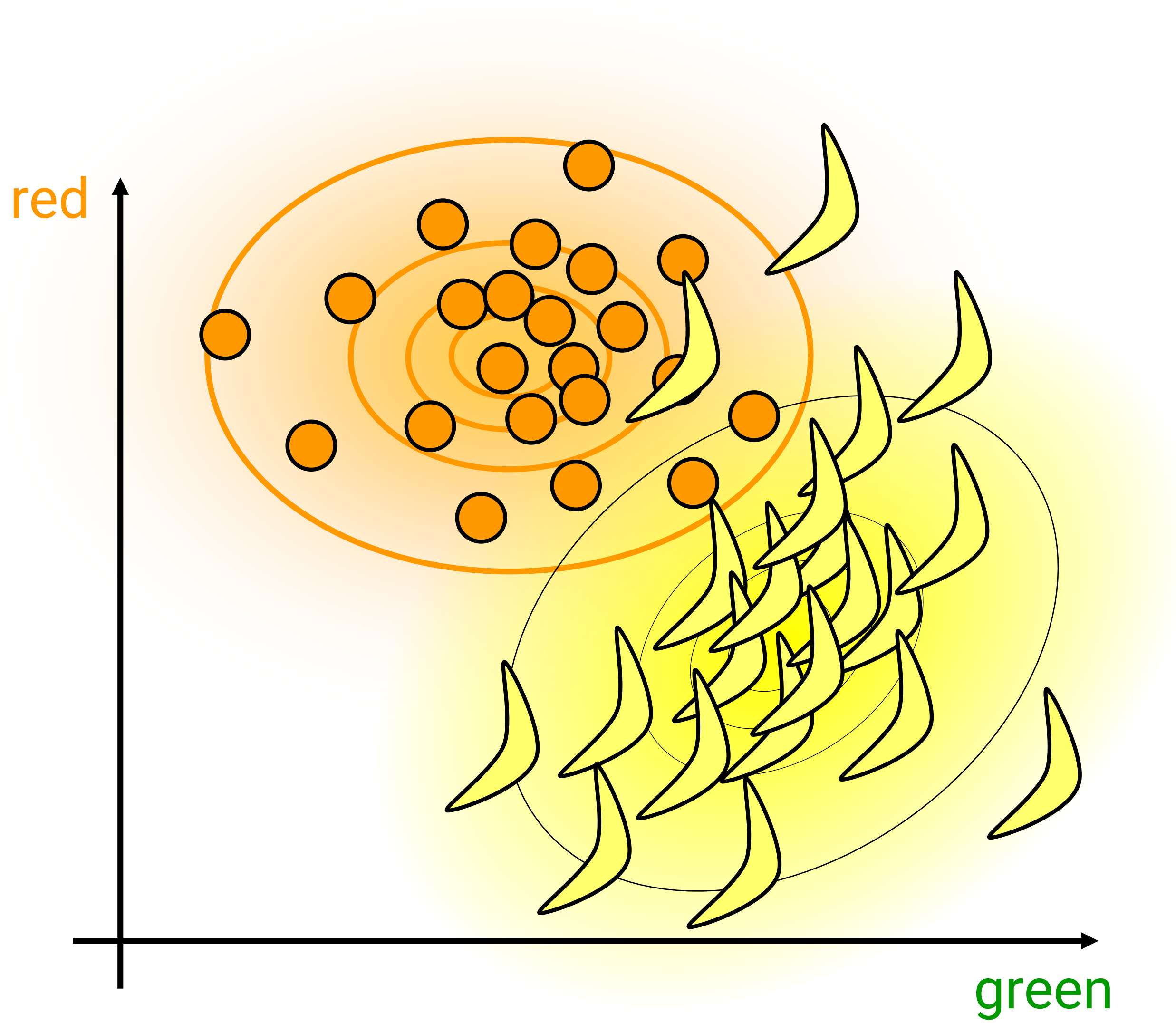
Machine Learning:

“Generative Models”

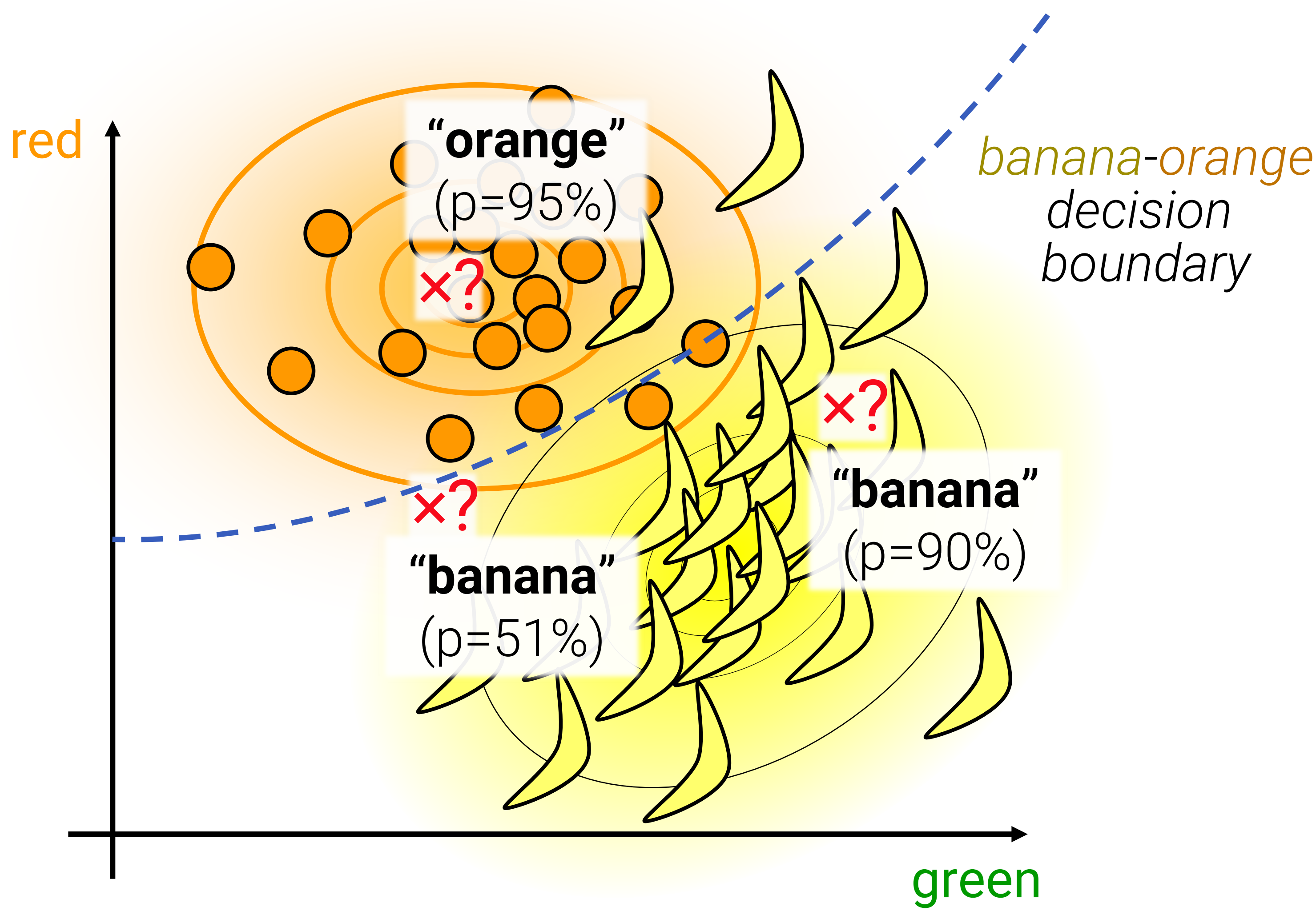
Learning Probabilities



Density Reconstruction



Bayesian Risk Minimization



Generative Learning

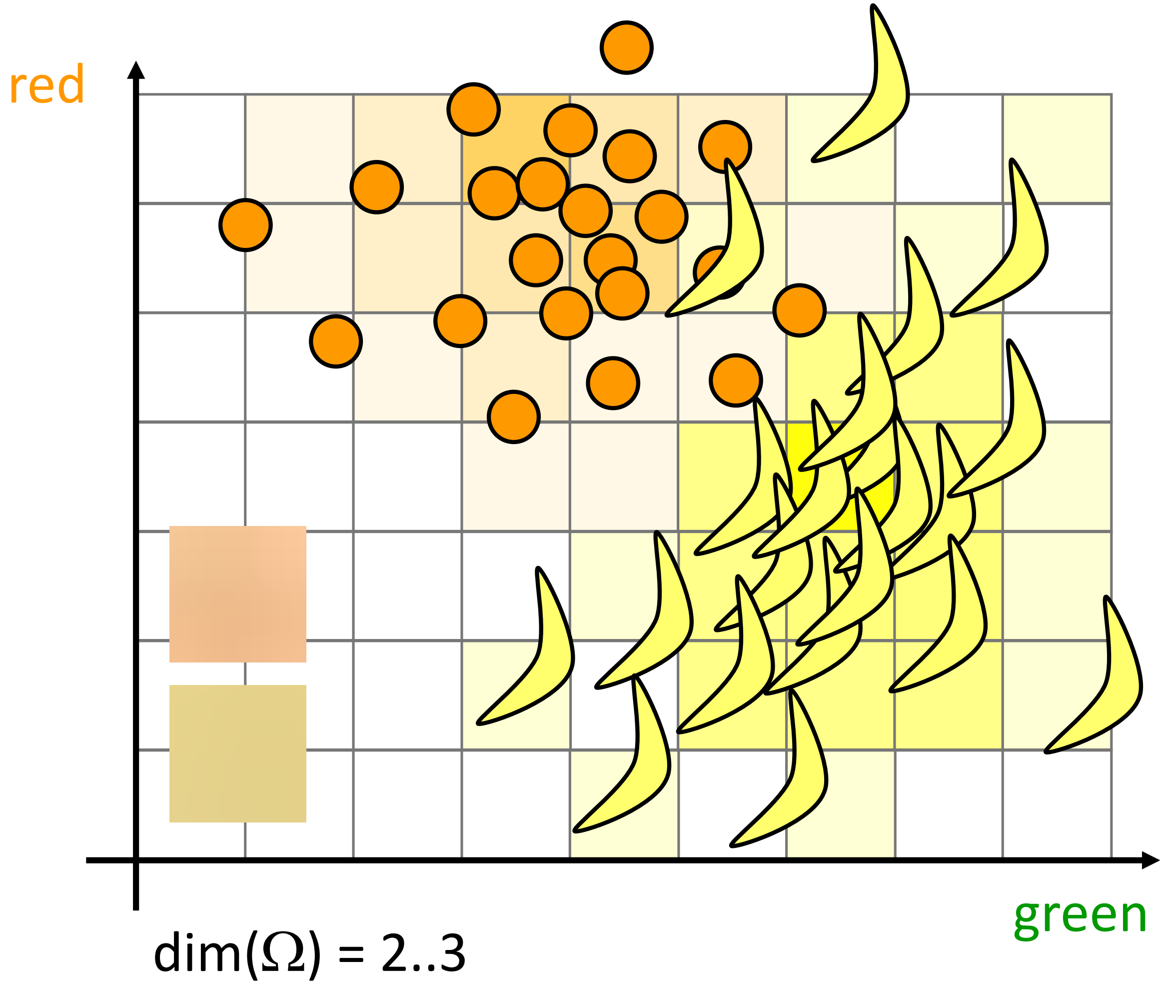
Very simple idea

- Collect data
- Estimate probability distribution
- Use learned probabilities for classification
- Always decide for the most likely case (largest probability)

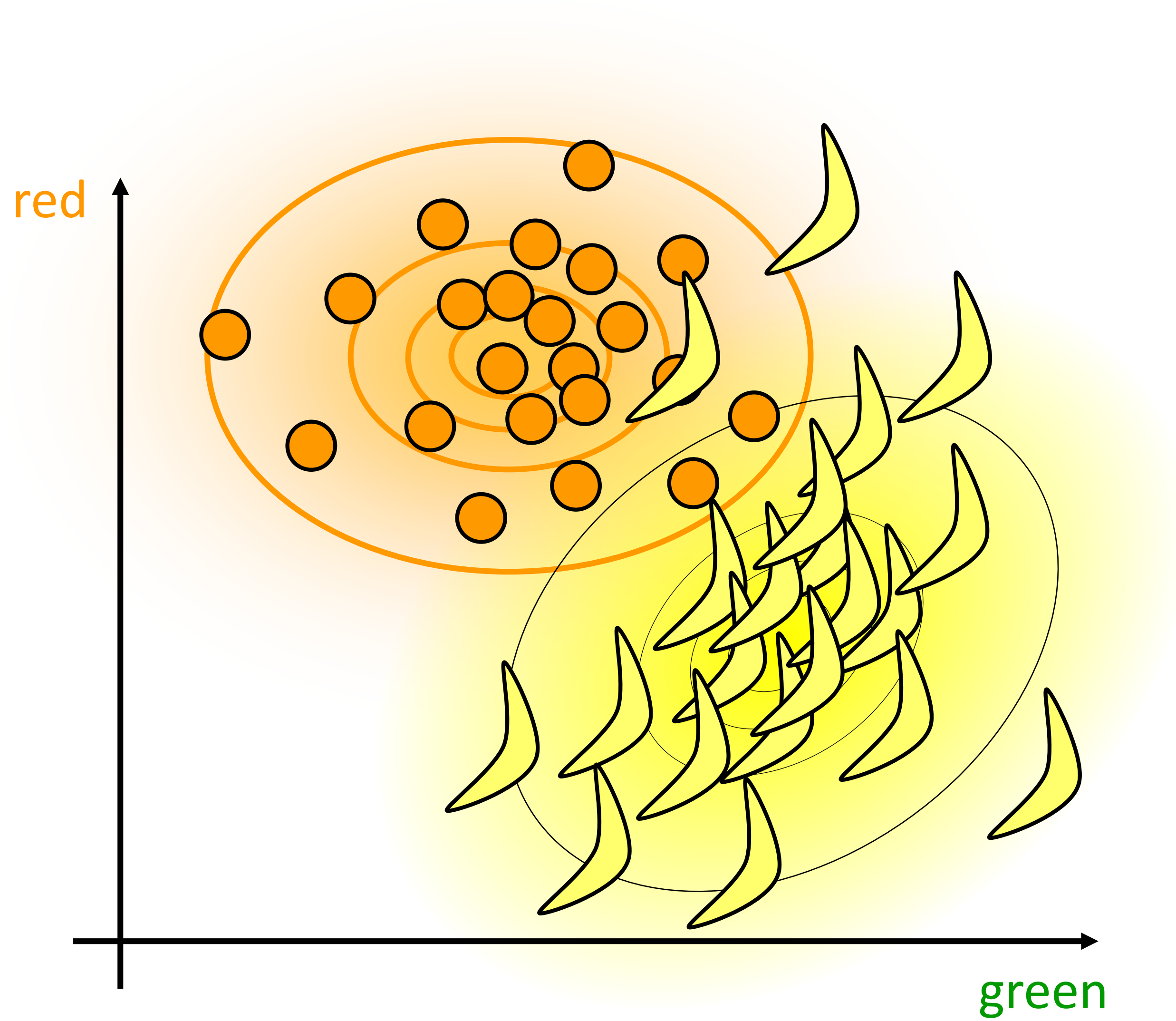
Easy to see

- If probability distributions is known exactly: decision is optimal (in expectation)
- “Minimal Bayesian risk classifier”

Simple Algorithm: Histograms



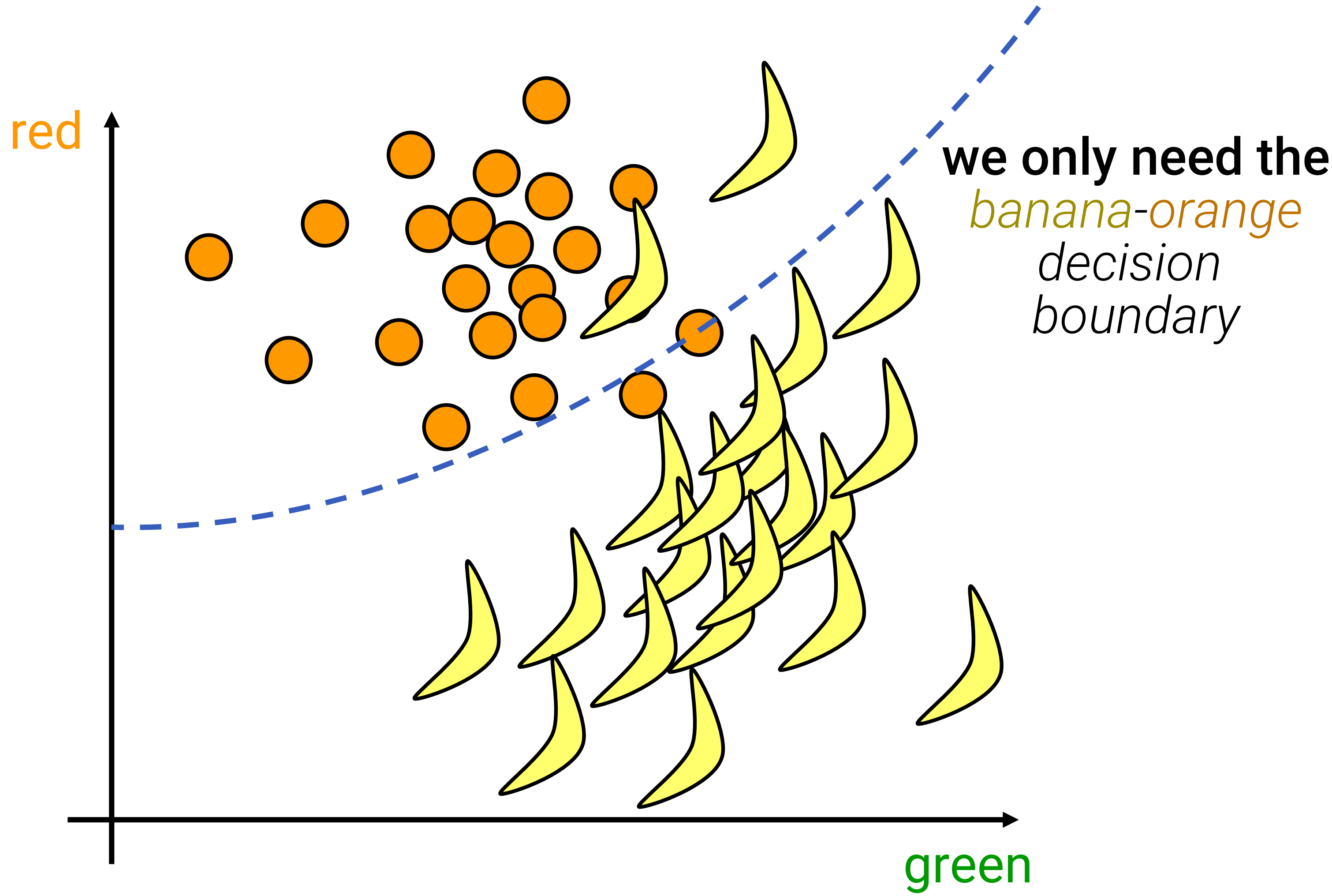
Simple Algorithm: Fit Gaussians



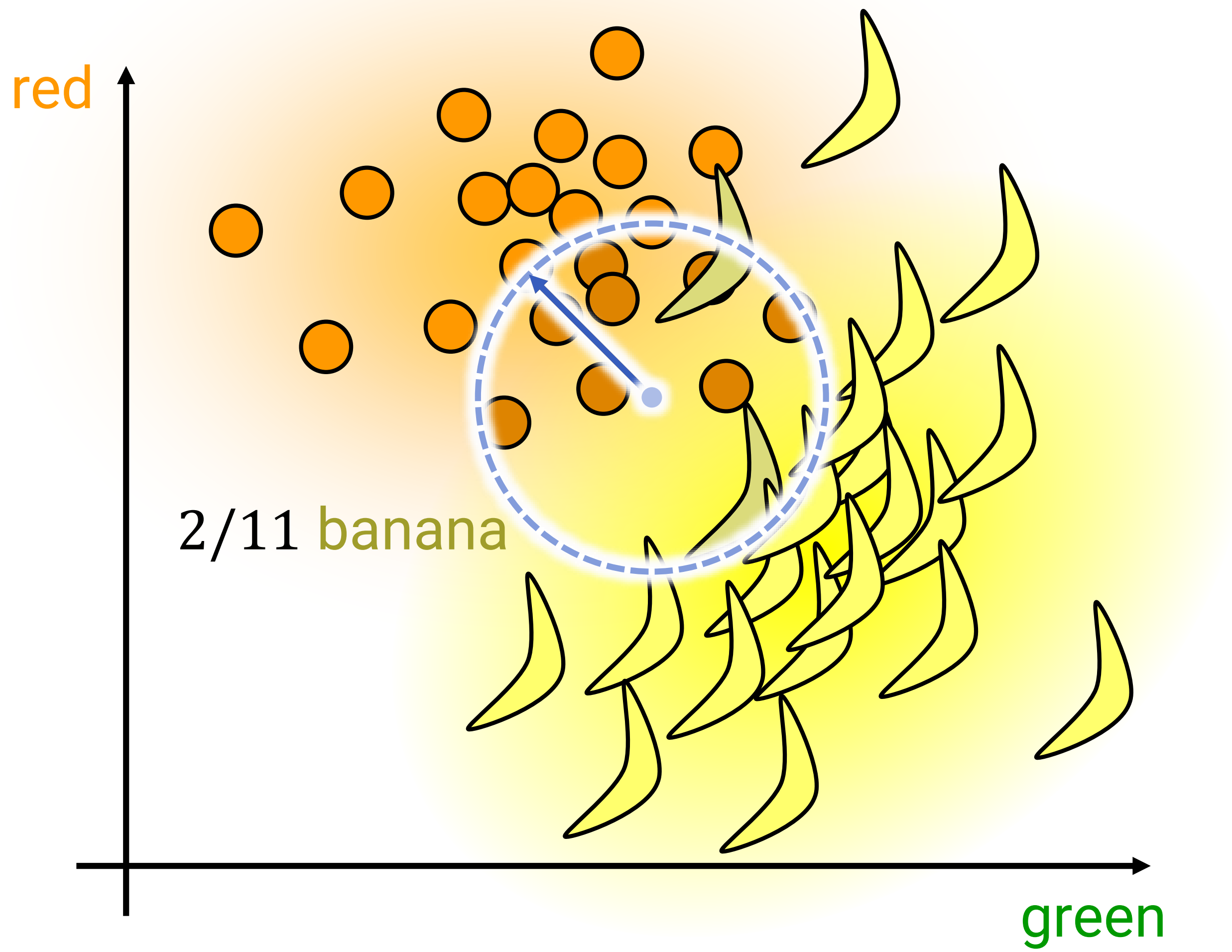
Machine Learning:

“Discriminative Models”

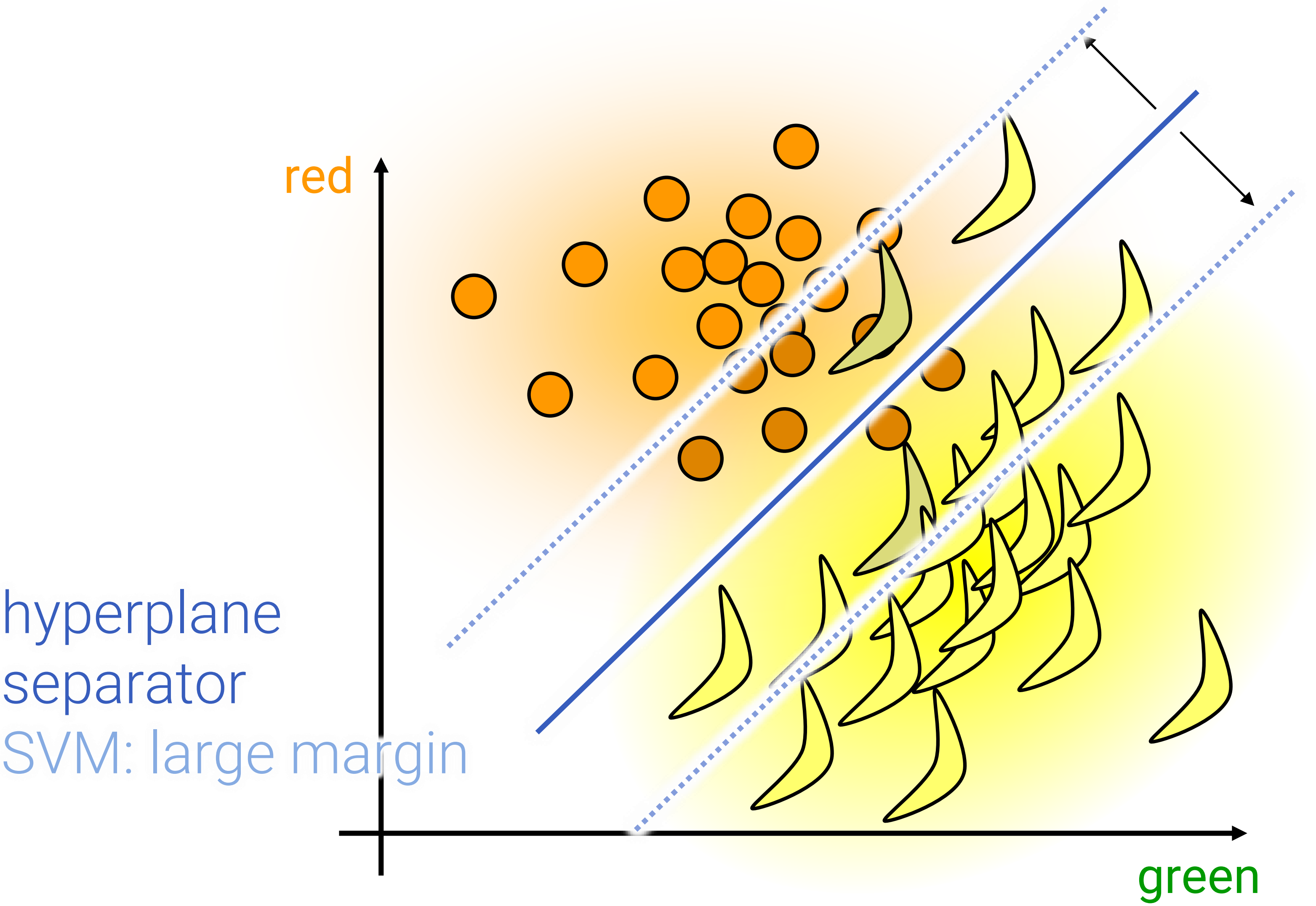
Idea: Why all the fuss?



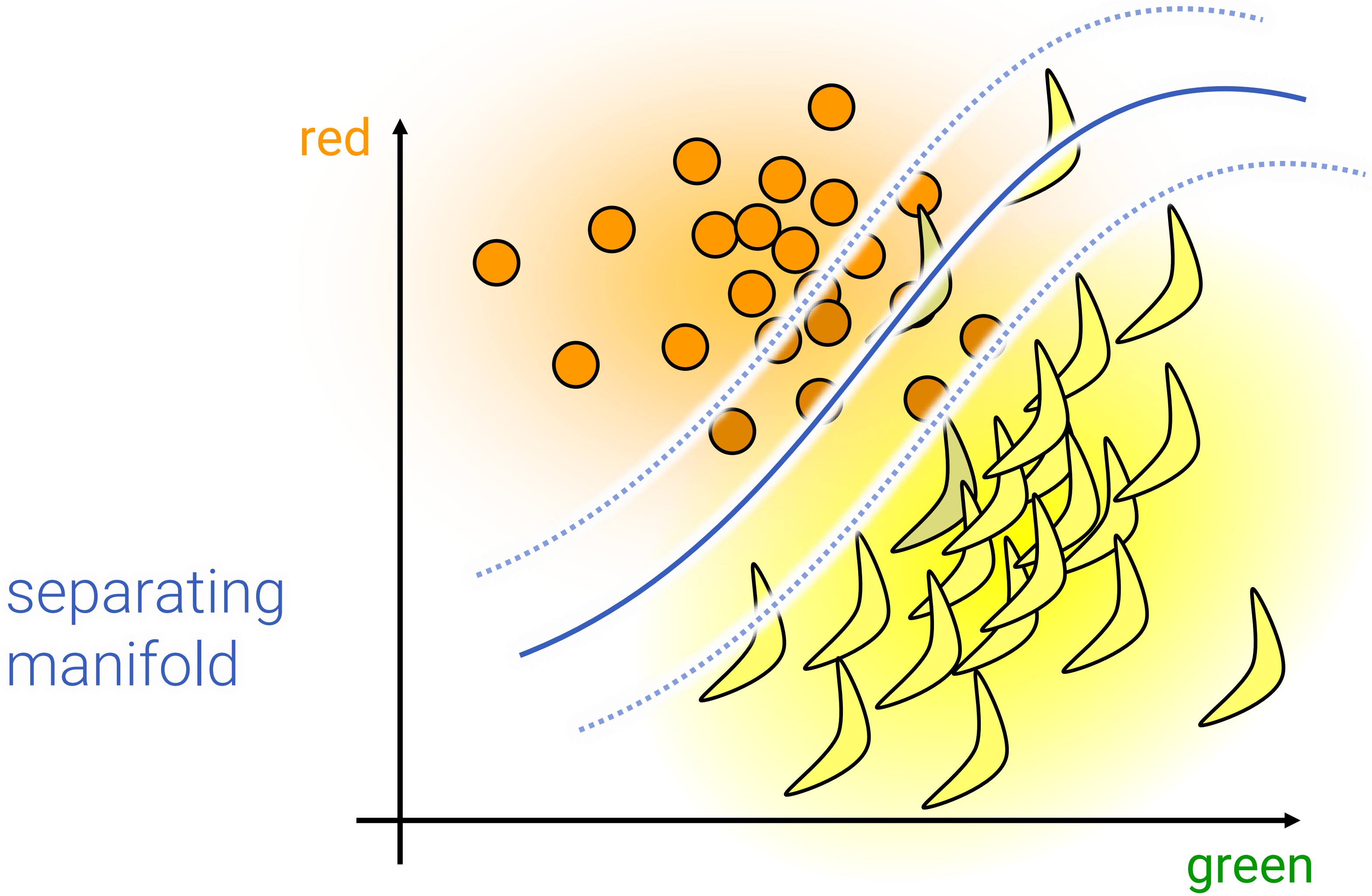
k-Nearest Neighbors



Linear Classifier (e.g. SVM)



General Classifiers



Generalization

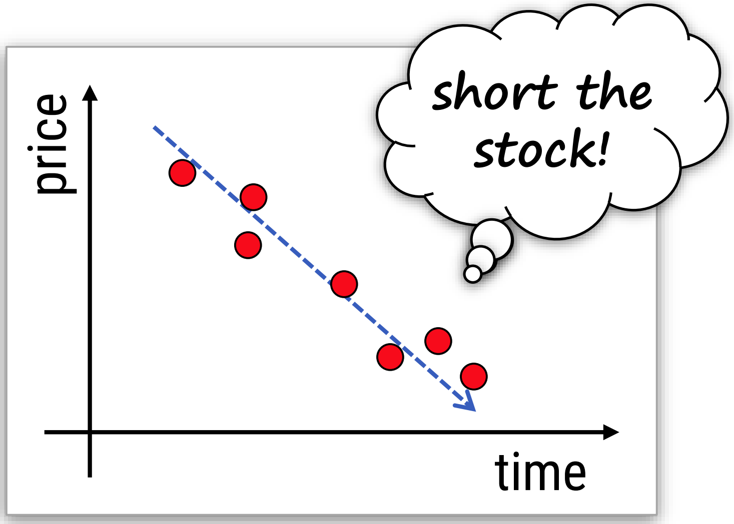
Unreliable Models

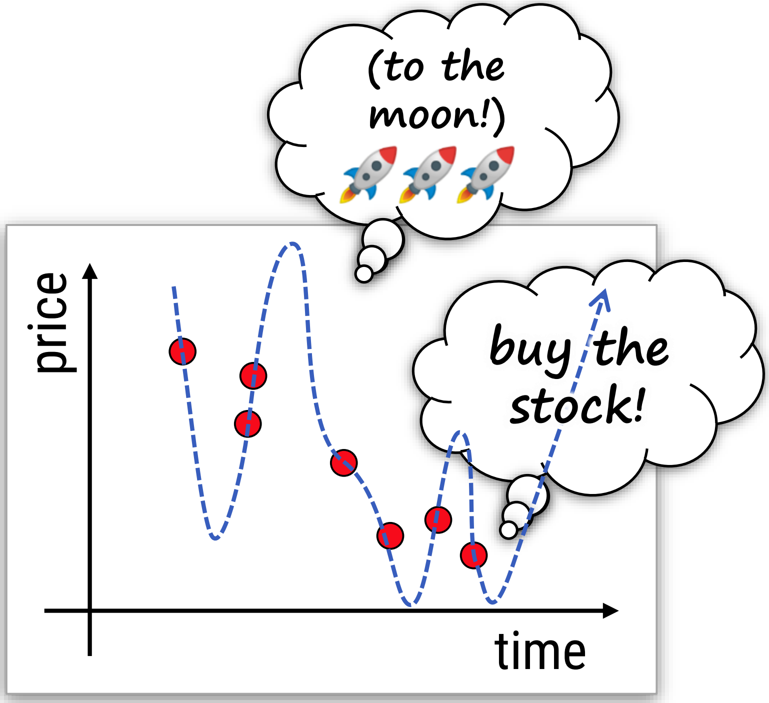
Previous example

- Betting on stock prices
- Polynomial fitting
- Seven observations

Degree k polynomial

- $k = 6$ fits any data
  - Unique model
  - But no predictive power
- $k = 5,4,3$ ...? fits any data
  - More or less reliable





We Care (Only) About Generalization

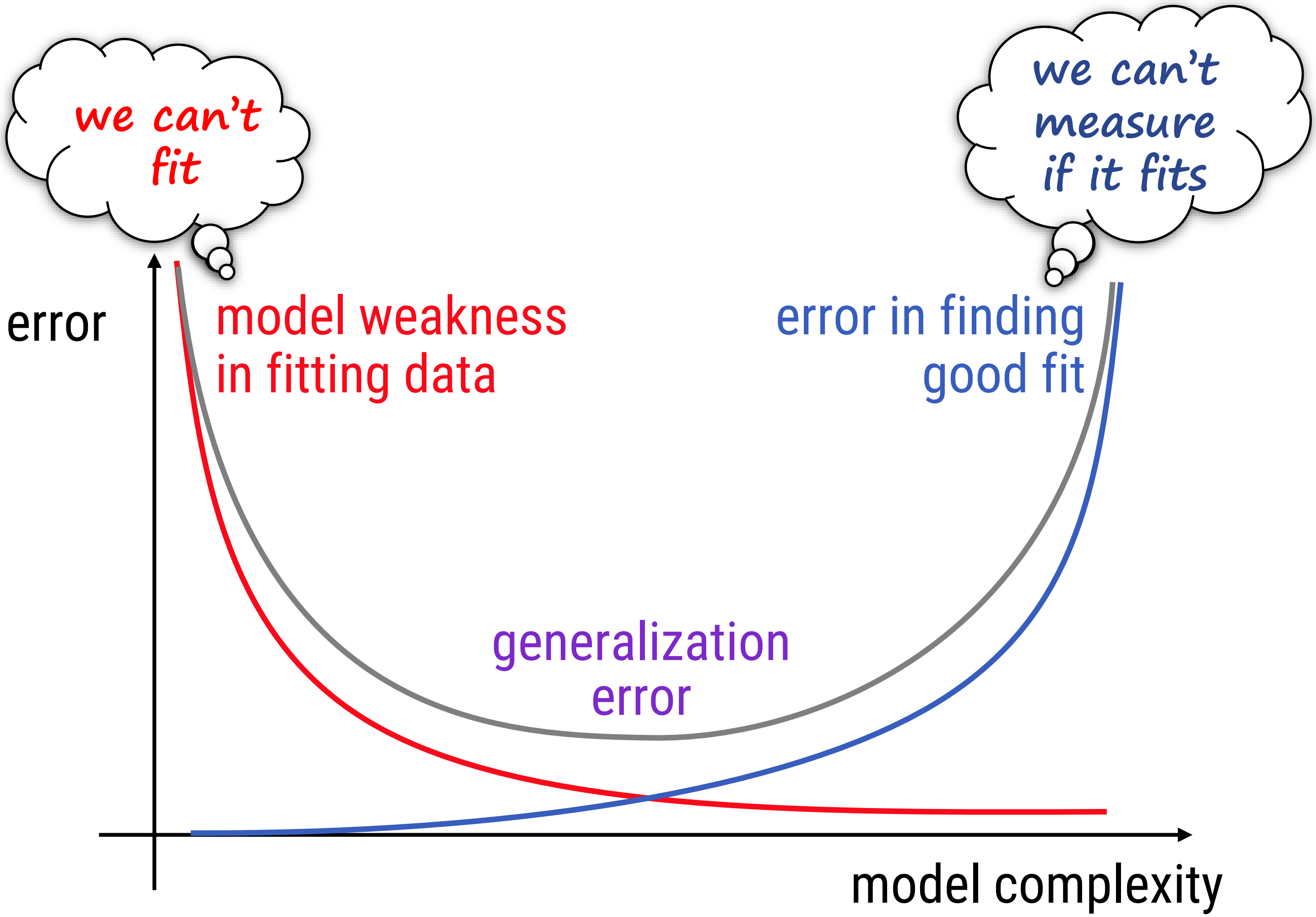
Performance on Training Data

- Might be misleading
- For example:
  - High degree polynomial fits perfectly
  - Very unlikely to fit in general

Problem

- How indicative is training performance for general performance (off-training data)?
  - Big error for complex models, small error for small models
  - We will make this quantitative soon

Bias Variance Trade-Off



Video #04a

Summary

Summary

Machine Learning

- Inductive reasoning: Learn solutions from examples
- Training vs. generalization: Beware of overfitting

Machine Learning & Statistics

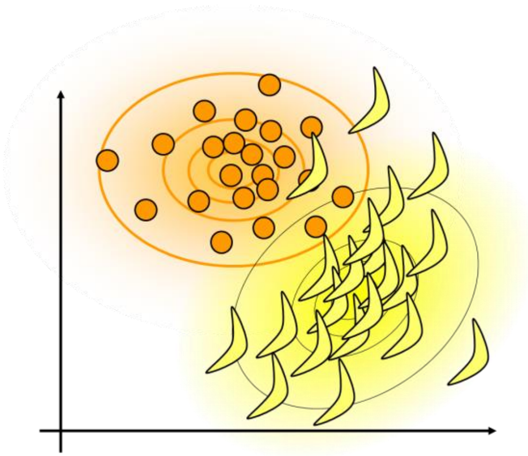
- Build suitable probabilistic model
- Determine probability distributions from examples

Two main approaches

- Generative: model statistics of everything
- Discriminative: Focus on task (classification)

Modelling 2

STATISTICAL DATA MODELLING



Chapter 4

Statistics and Machine Learning

Michael Wand · Institut für Informatik, JGU Mainz · michael.wand@uni-mainz.de

Video #04

Statistics & Machine Learning

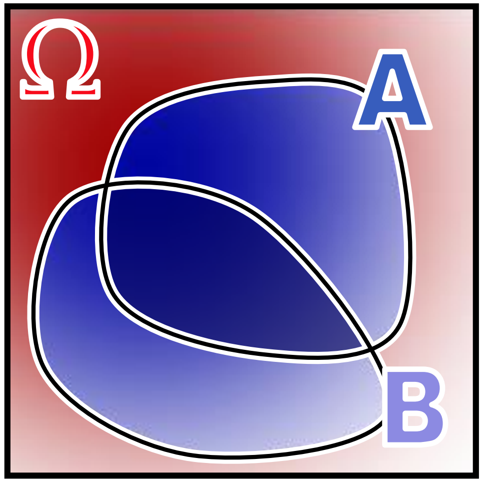
- Machine Learning Basics
- **Bayesian Inference for ML**
- Learning & Inference

Bayes' Rule

Derivation of Bayes' rule

Bayes' rule

$$\Pr(A \mid B) = \frac{\Pr(B \mid A) \cdot \Pr(A)}{\Pr(B)}$$



Derivation

- $\Pr(A \cap B) = \Pr(A \mid B) \cdot \Pr(B)$
  $\Pr(A \cap B) = \Pr(B \mid A) \cdot \Pr(A)$

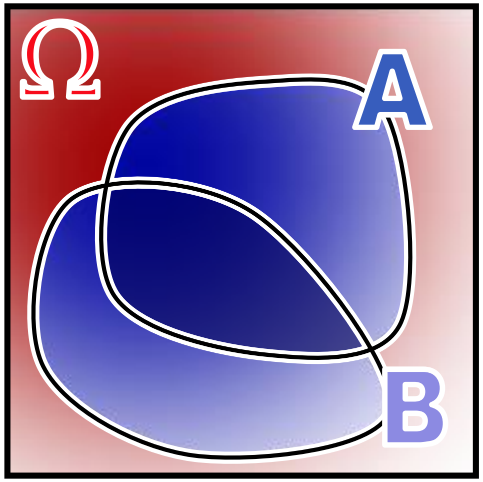
$\Rightarrow \Pr(A \mid B) \cdot \Pr(B) = \Pr(B \mid A) \cdot \Pr(A)$

Bayes for Densities

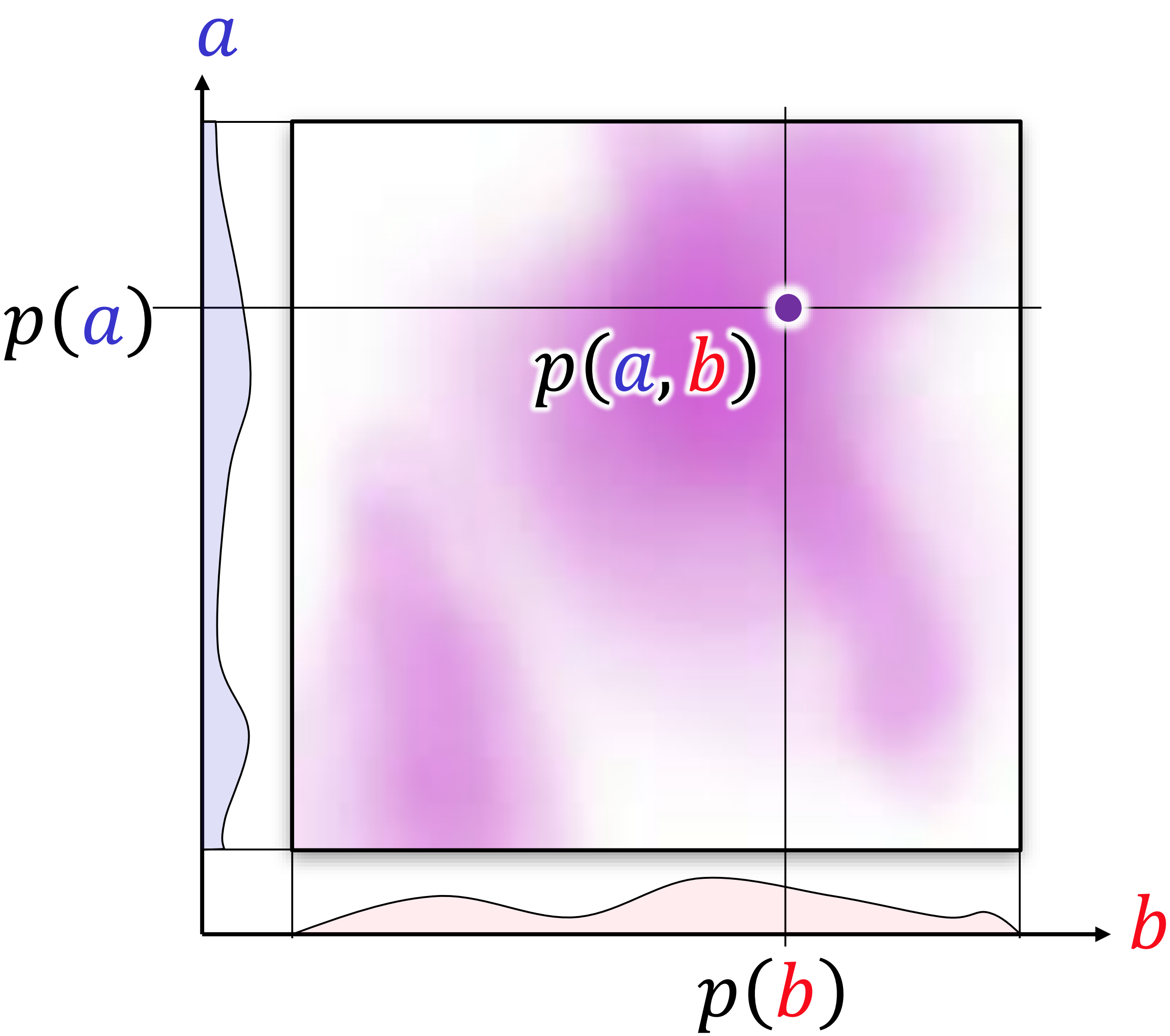
Bayes' rule for densities

$$p(x|y) = \frac{p(y|x) \cdot p(x)}{p(y)}$$

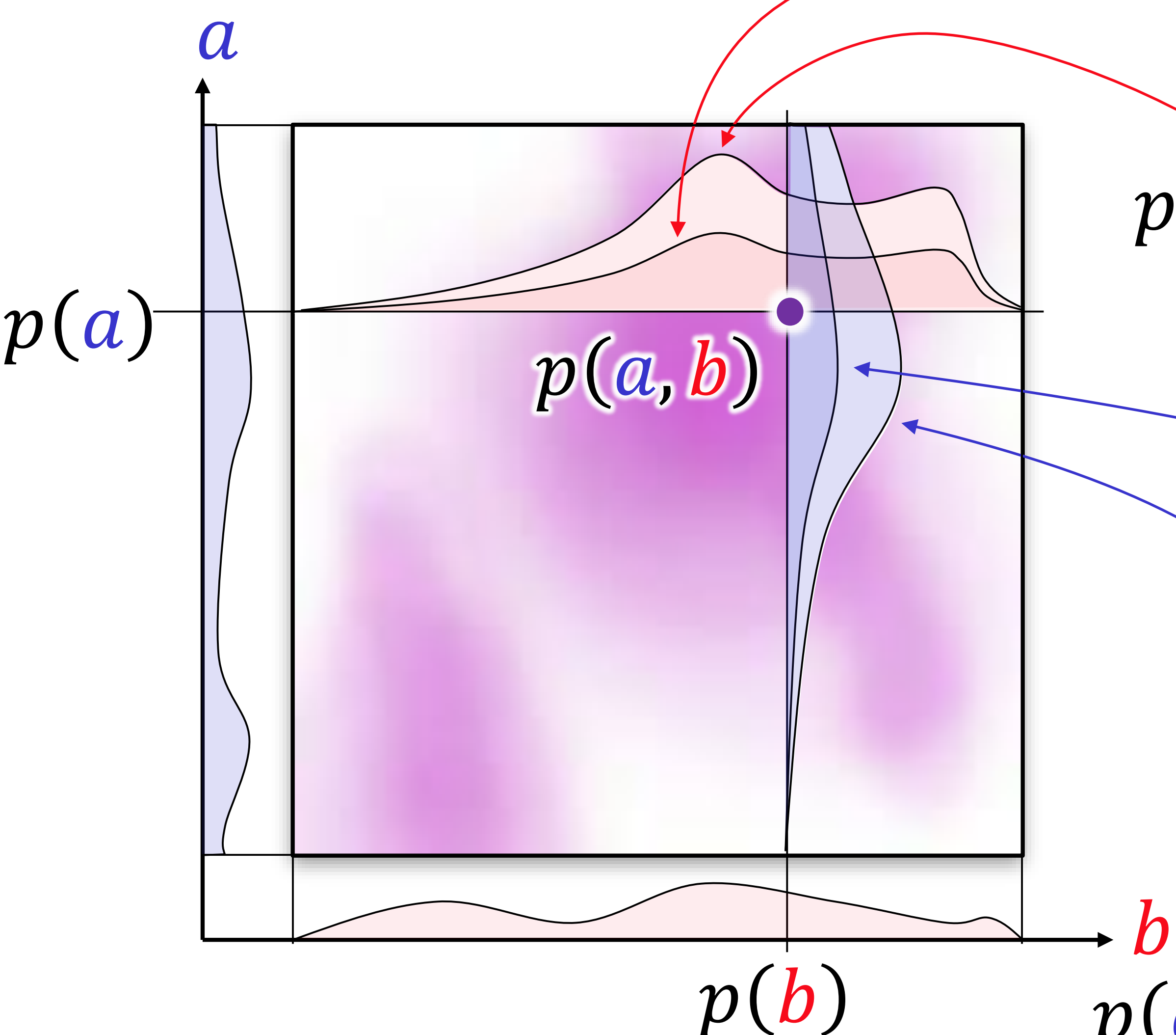
$$= \frac{p(y|x) \cdot p(x)}{\int_{x \in \Omega(X)} p(y|x)p(x)dy}$$



Bayes Rule for Densities: Visualization



Bayes Rule for Densities: Visualization



$$p(b|a) = \frac{p(a,b)}{p(a)}$$

$$p(a|b) = \frac{p(a,b)}{p(b)}$$

$$p(b) = \int_a p(a,b)\,da$$

$$p(a|b) = \frac{p(b|a)p(a)}{p(b)}$$

Bayesian Statistics for ML

A Practical How-To

Recommended Reading:
http://www.dgp.toronto.edu/~hertzman/ibl2004/notes.pdf

Bayesian Toolset

Rules

- Normalization

$$\int_{\Omega} p(\mathbf{x})d\mathbf{x} = 1\,, \qquad \int_{\Omega} p(\mathbf{x}|\mathbf{y})d\mathbf{x} = 1$$

- Marginalization

$$p(\mathbf{x}) = \int_{\Omega} p(\mathbf{x}, \mathbf{y})d\mathbf{y}$$

Bayesian Toolset

More rules…

- Product rule

$$p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x}|\mathbf{y}) \cdot p(\mathbf{y})$$

$$\begin{aligned} p(\mathbf{x}, \mathbf{y}, \mathbf{z}) &= p(\mathbf{x}|\mathbf{y}, \mathbf{z}) \cdot p(\mathbf{y}, \mathbf{z}) \\ &= p(\mathbf{x}|\mathbf{y}, \mathbf{z}) \cdot p(\mathbf{y}|\mathbf{z}) \cdot p(\mathbf{z}) \end{aligned}$$

- Product rule: condition on any (sub-) tuple(s)

$$\begin{aligned} p(\mathbf{x}, \mathbf{y}, \mathbf{z}) &= p(\mathbf{x}, \mathbf{y}|\mathbf{z}) \cdot p(\mathbf{z}) \\ &= p(\mathbf{x}|\mathbf{y}, \mathbf{z}) \cdot p(\mathbf{y}|\mathbf{z}) \cdot p(\mathbf{z}) \end{aligned}$$

Bayesian Toolset

Rules

- Marginalization (e.g. “nuisance” parameters)

$$p(\mathbf{x}) = \int_{\Omega(\varphi)} p(\mathbf{x}, \varphi) d\varphi$$

  - Integrate over everything you do not care about
  - If too costly: maximize with well-designed prior

- Direct observation

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{y})}$$

  - We have seen / we know $\mathbf{y}$
  - Divide joint pd $p(\mathbf{x}, \mathbf{y})$ by $p(\mathbf{y})$ to obtain conditional pd

Bayesian Toolset

When to use what?

- Marginalization

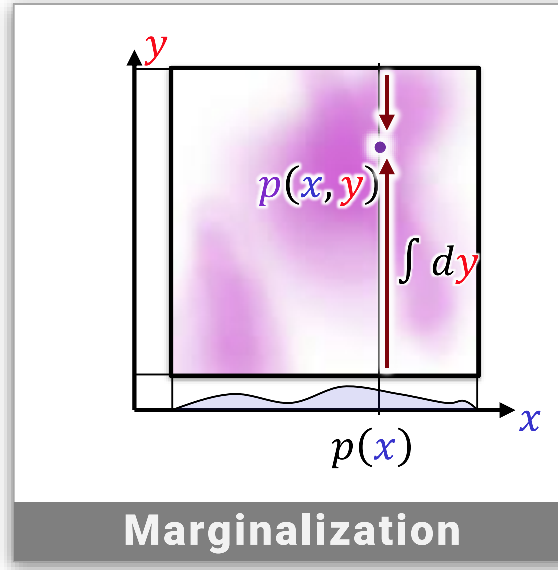
$$p(\mathbf{x}) = \int_{\Omega(\mathbf{y})} p(\mathbf{x}, \mathbf{y}) d\mathbf{y}$$

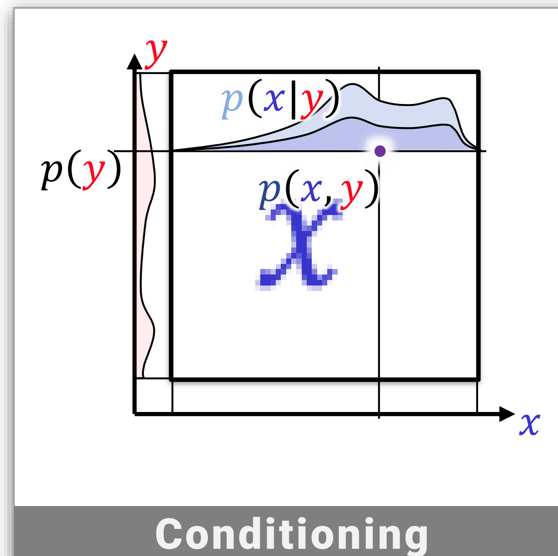
  - $\mathbf{y}$ could be anything
  - Want likelihood for $\mathbf{x}$ (overall, any $\mathbf{y}$)

- Conditioning

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{y})}$$

  - We have seen / we know $\mathbf{y}$!
  - $\mathbf{y}$ is fixed, we want to update (renormalize) distribution





Bayesian Toolset

Bayes' Rule

$$p(\mathbf{x}|\mathbf{y}) = \frac{\overbrace{p(\mathbf{y}|\mathbf{x}) \cdot p(\mathbf{x})}^{p(\mathbf{x},\mathbf{y})}}{p(\mathbf{y})}$$

- "Inverse" problem
  - We know conditional & marginal probabilities
  - We want to know the inverse conditional
  - Determine $p(\mathbf{x}|\mathbf{y})$ from $p(\mathbf{y}|\mathbf{x})$, $p(\mathbf{x})$

Bayes vs. simple conditioning

- We do not have $p(\mathbf{x},\mathbf{y})$ directly
- But we can model / observe $p(\mathbf{y}|\mathbf{x})$, $p(\mathbf{x})$

Example

Measurement device

- State of measured object: $\mathbf{X}$
- Measured data: $\mathbf{D}$

What is $\mathbf{X}$ given data $\mathbf{D}$?
$p(\mathbf{X}|\mathbf{D})$

We can model how device works

- "Likelihood" $p(\mathbf{D}|\mathbf{X})$

We have a rough idea how $\mathbf{X}$ looks like

- "Prior" $p(\mathbf{X})$

With this, we can compute inverse $p(\mathbf{X}|\mathbf{D})$

Hertzman's Principles

Laplace (1814)

"*Probability theory is nothing more than common sense reduced to calculation*"

[image: Wikipedia]

Pierre-Simon Laplace
(1749–1827)

Further principles

- Build complete model
- Infer knowledge (given observations)
- Bayes' Rule to infer from observation
- Marginalize to remove unknown parameters

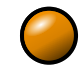
from Hertzman's course notes: [http://www.dgp.toronto.edu/~hertzman/ibl2004/bayes2004.pdf]

Likelihoods & Priors

Merging Information

Bayesian Models

Scenario

- Customer picks banana  (X = 0) or orange  (X = 1)
- Object X creates image D

Modeling

- Given image D (observed), what was X (latent)?

$$P(X|D) = \frac{P(D|X)P(X)}{P(D)}$$

$$P(X|D) \sim P(D|X)P(X)$$

Relation

Easy to confuse

- $p(x|y)$ and $p(x, y)$ with y fixed

Difference

- $p(x|y) = \frac{p(x,y)}{p(y)} = \frac{p(x,y)}{\int_{\Omega(x)} p(x,y)dx}$
- Conditional probability is normalized
  - Integrates to one
- Careful for varying y!
  - $p(x|y) \not\propto p(x, y)$ (not proportional in 2D!)
  - Normalization varies with y!

Bayes Rule for ML

Variables: Explanation X, data D, model θ

Learning θ given training pairs (D, X)

$$P_\theta(X|D) = \frac{P_\theta(D|X)P_\theta(X)}{P_\theta(D)}$$

Inferring X from data D given model θ

$$P_\theta(X|D) = \frac{P_\theta(D|X)P_\theta(X)}{P_\theta(D)}$$

◂ independent of X

$$P_\theta(X|D) \sim P_\theta(D|X)P(X)$$

Bayesian Models

Statistical Model

$$\underbrace{P_\theta(X|D)}_{\text{posterior}} = \frac{\overbrace{P_\theta(D|X)}^{\text{likelihood}}\overbrace{P_\theta(X)}^{\text{prior}}}{\underbrace{P_\theta(D)}_{\text{evidence}}}$$

Bayesian Models

Our Classifier

$$\underbrace{P_\theta(X|D)}_{\text{posterior}} = \frac{\overbrace{P_\theta(D|X)}^{\text{fruit} \to \text{img}}\ \overbrace{P_\theta(X)}^{\text{freq. of fruits}}}{\underbrace{P_\theta(D)}_{\text{fruit image probability}}}$$

Bayesian Models

Generative Model

learn — fruit → img

learn — freq. of fruits

$$P_\theta(X|D) = \frac{P_\theta(D|X)\,P_\theta(X)}{P_\theta(D)}$$

posterior — compute

fruit image probability — **compute / ignore**
(compute from
learned likelihood)

Properties

- Comprehensive model:
  Full description of how data is created
- Might be complex (how to create images of fruit?)

Bayesian Models

Discriminative Model

$$P_\theta(X|D) = \frac{P_\theta(D|X)\,P_\theta(X)}{P_\theta(D)}$$

posterior

fruit → img

freq. of fruits

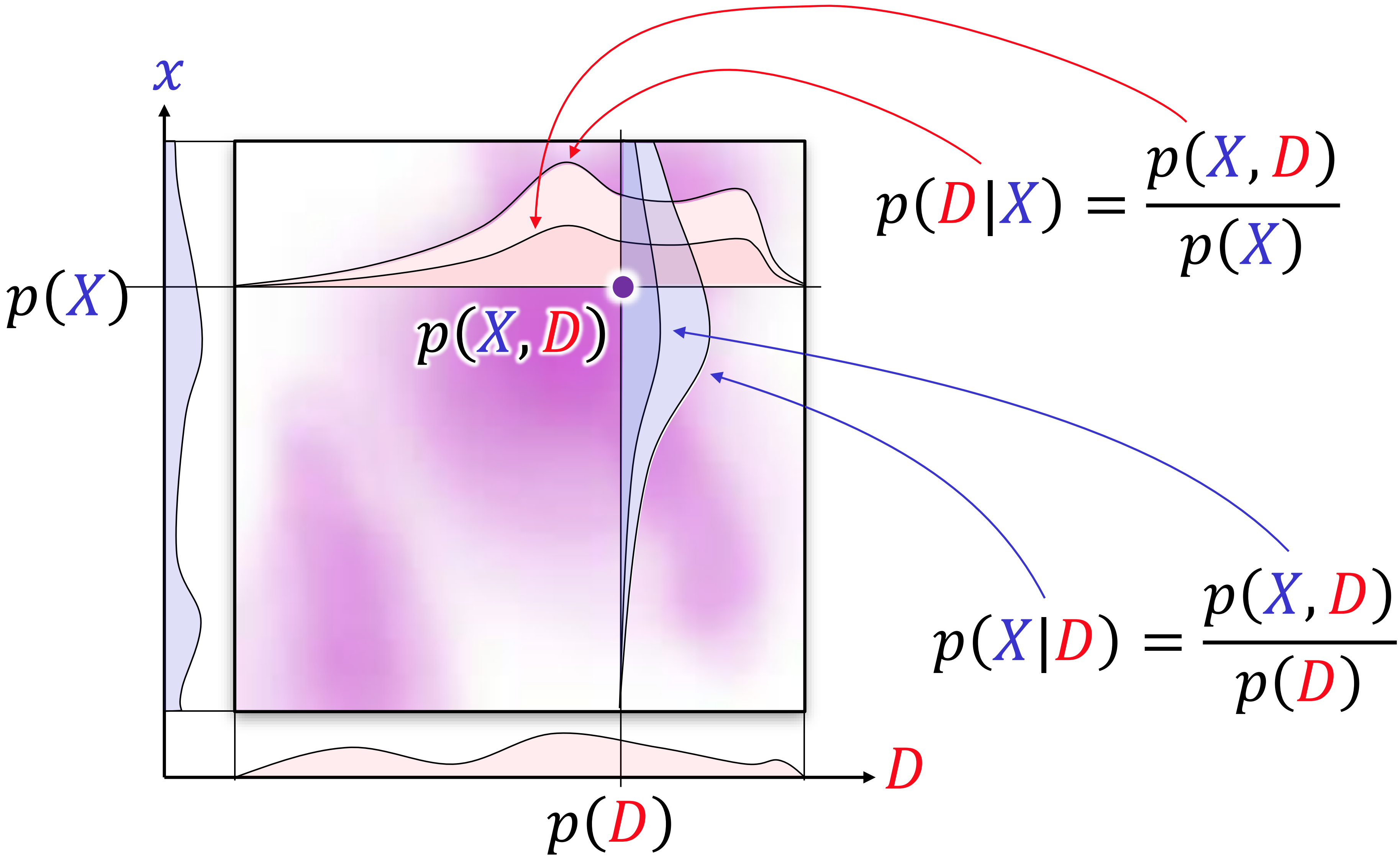
fruit image probability

learn directly

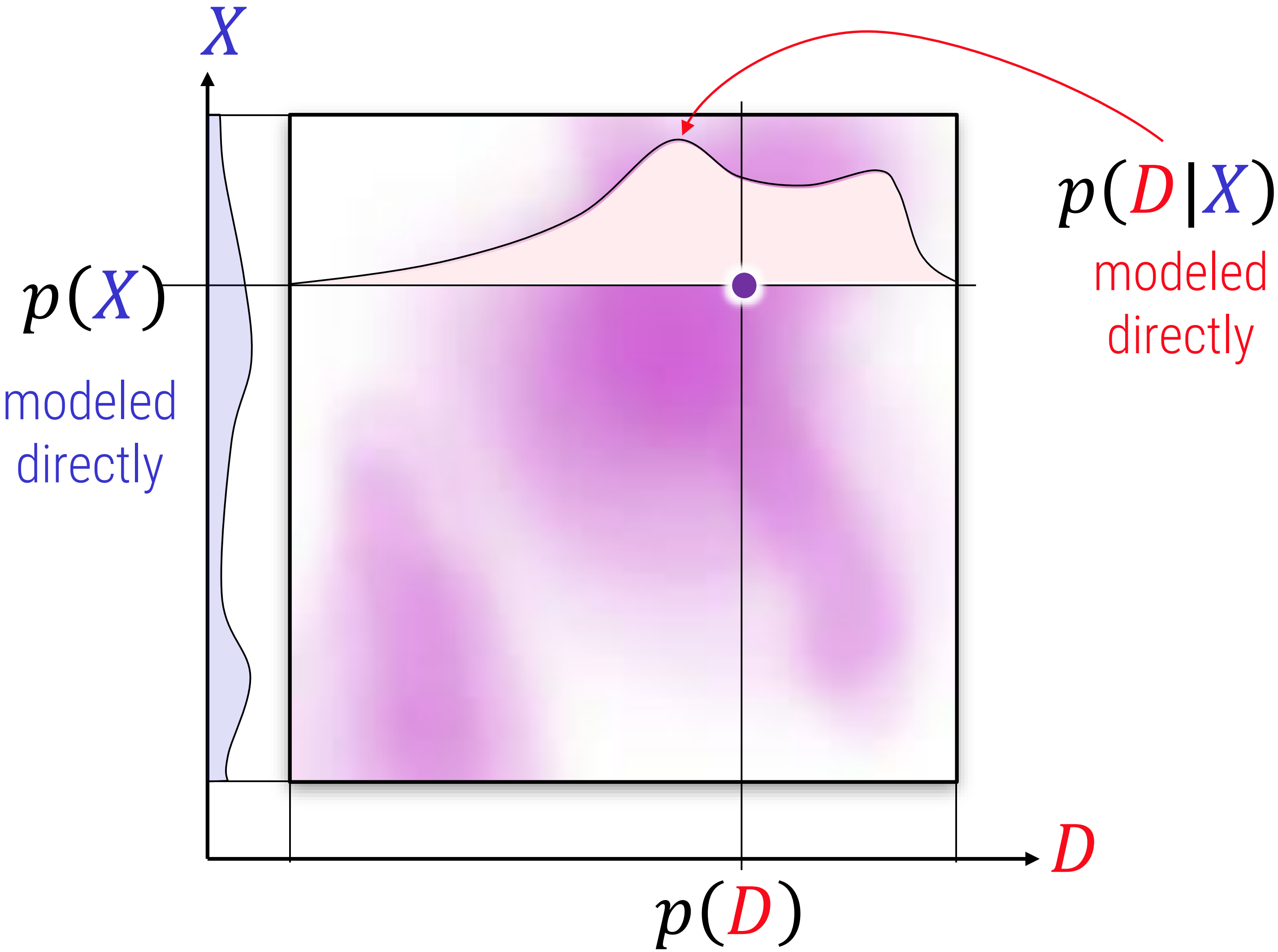
Often easier to learn

- Learn mapping from phenomenon to explanation
  - Less “powerful”: needs less data
- Not trying to explain the whole phenomenon
  - Can use reduced representation / features

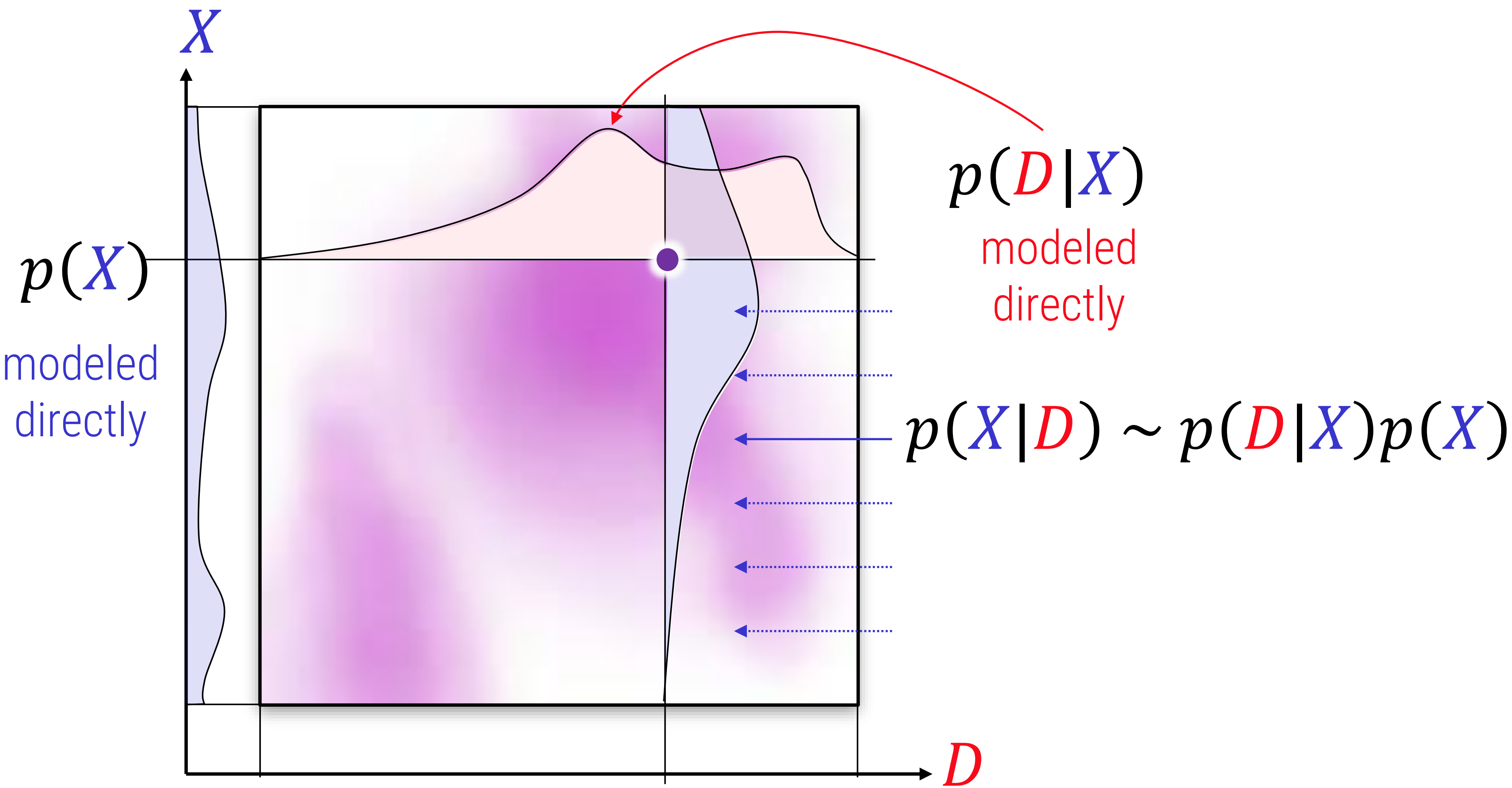
Generative Models



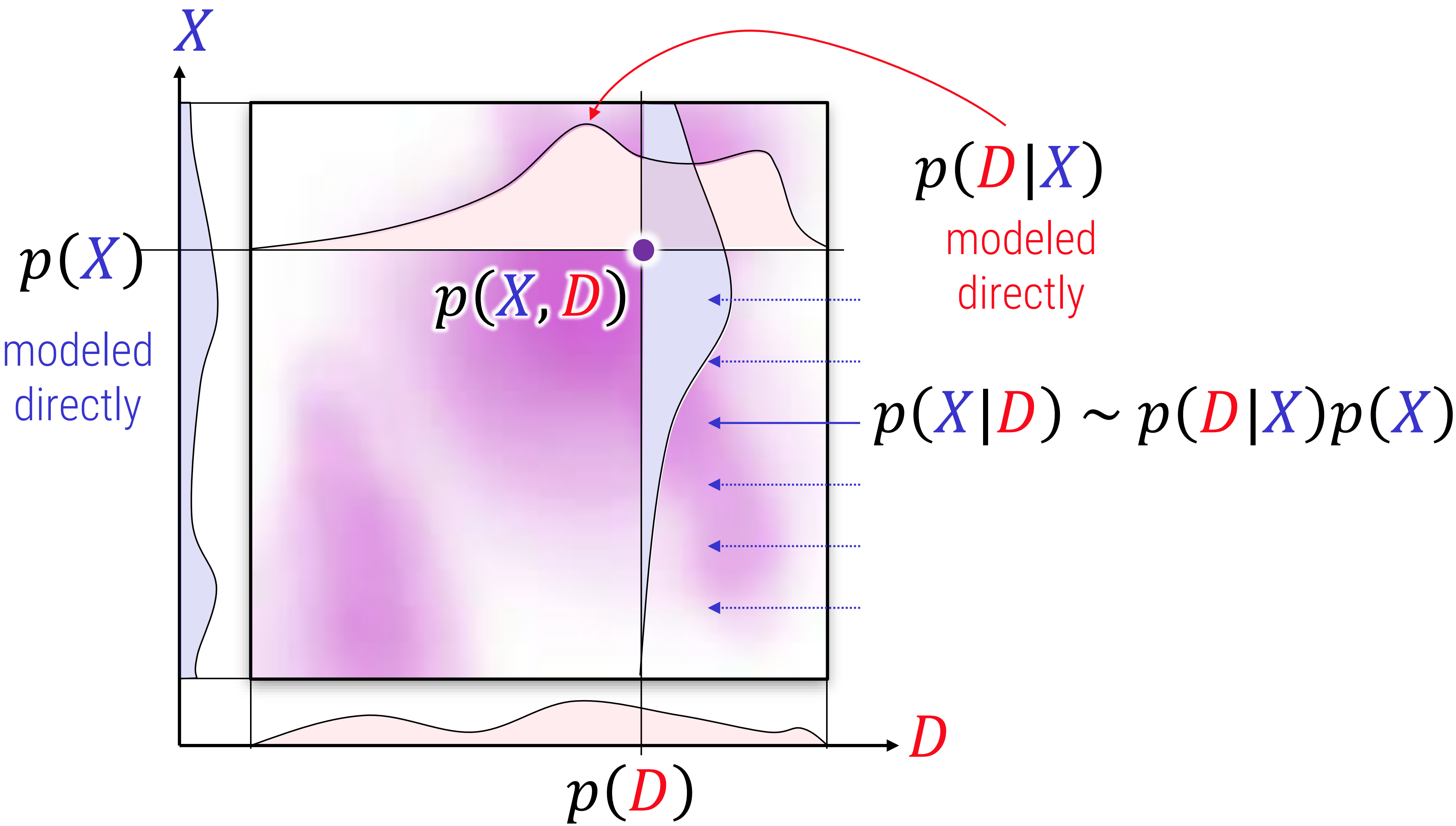
Generative Models



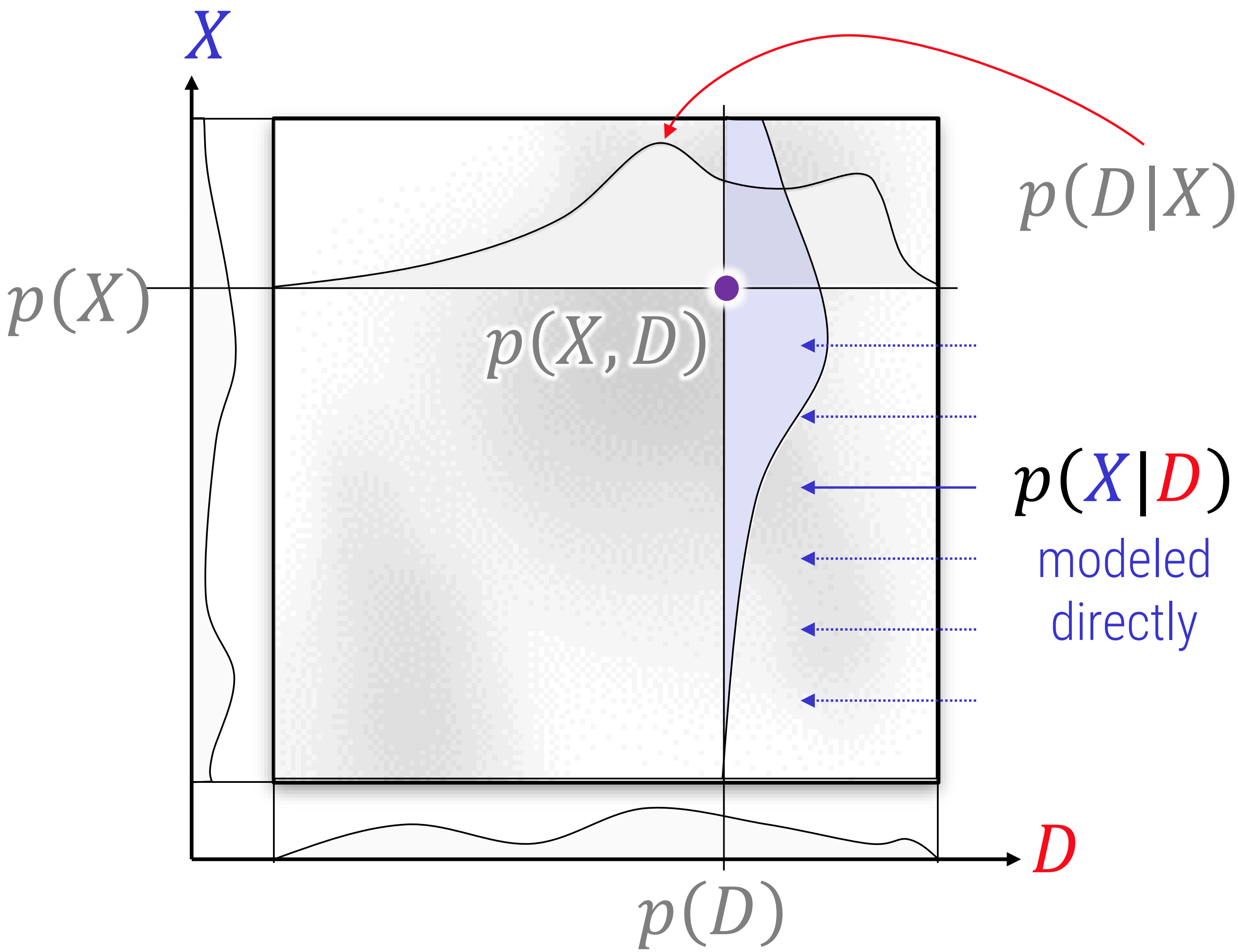
Generative Models



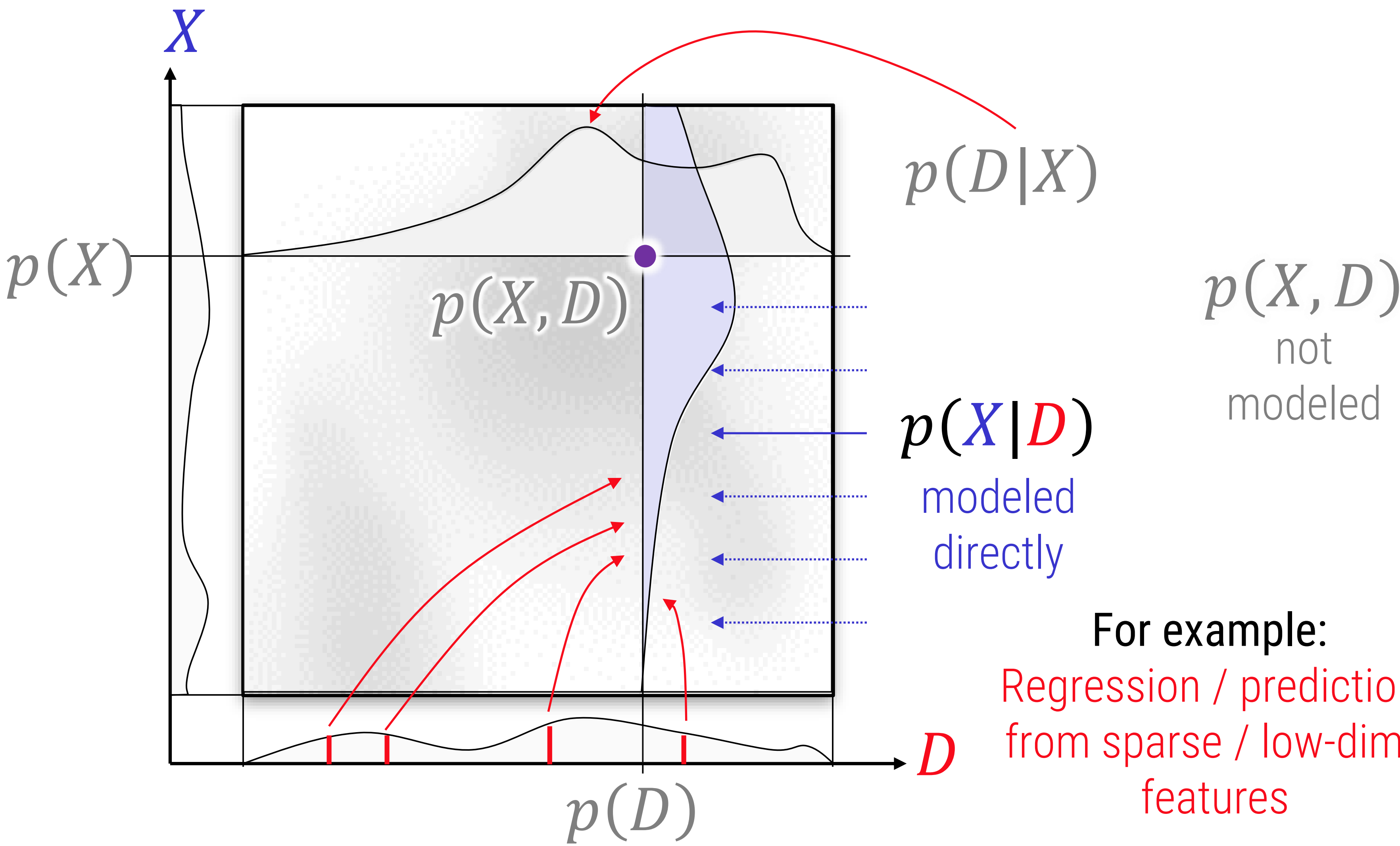
Generative Models



Discriminative Model

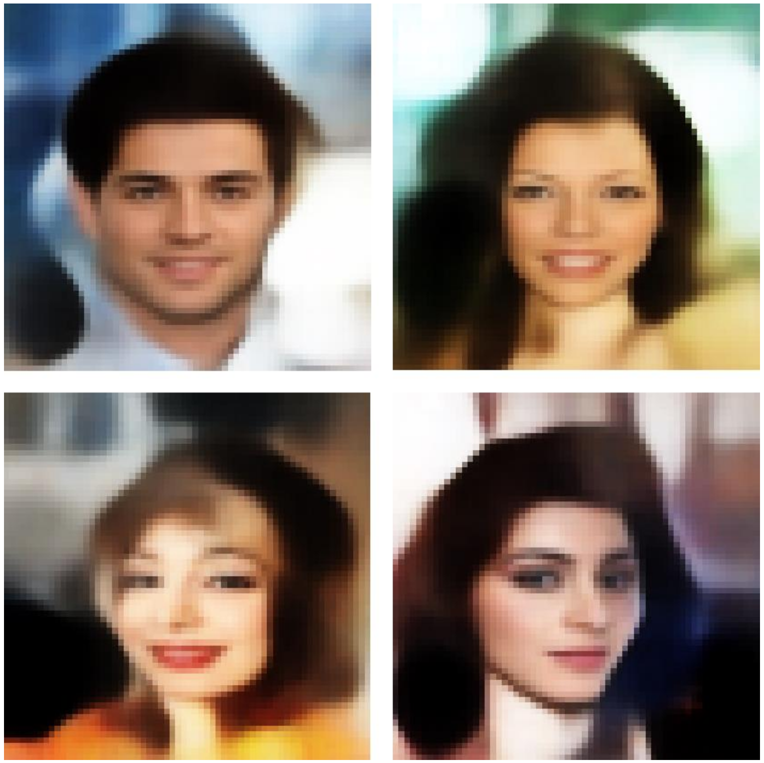


Discriminative Model

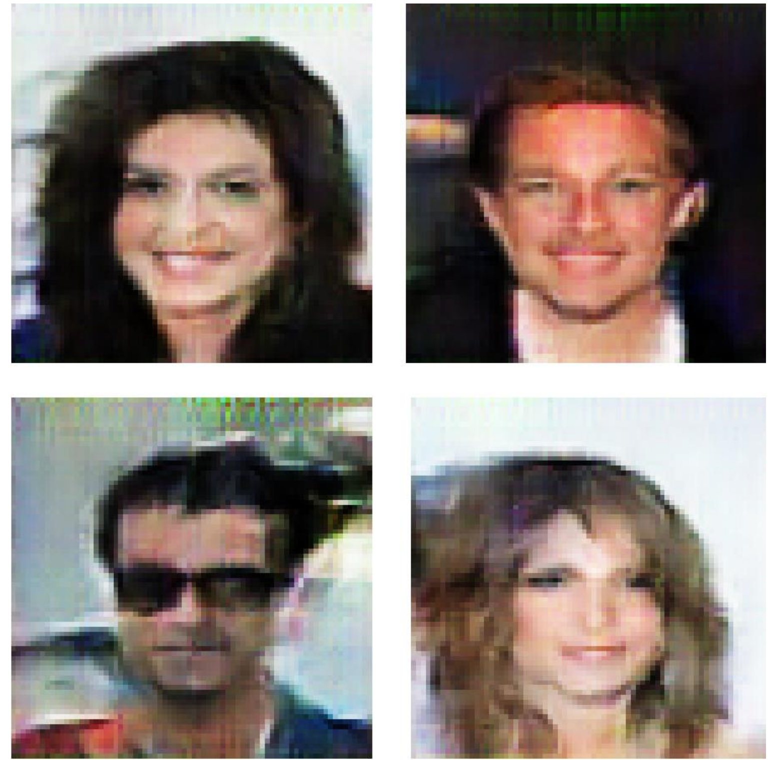


For example:
Regression / prediction from sparse / low-dim. features

Example: Generative Models



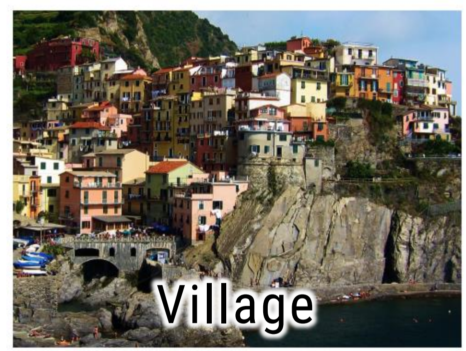
Autoencoder
(PCA in latent space)



WGAN-GP
(generative adversarial network)

[results courtesy of D. Schwarz, D. Klaus, A. Rübe]

Discriminative Models



[not an actual classification result, just photos]

Video #04b

Summary

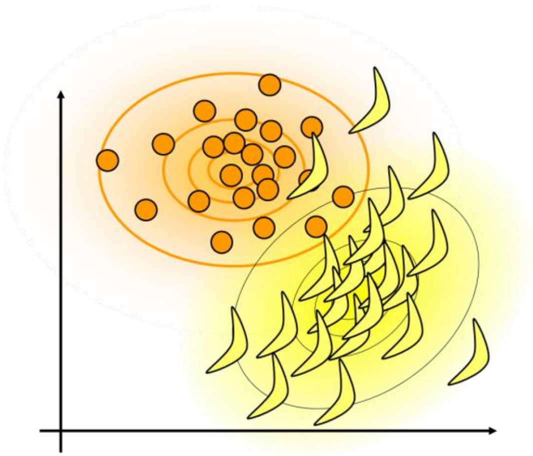
Summary

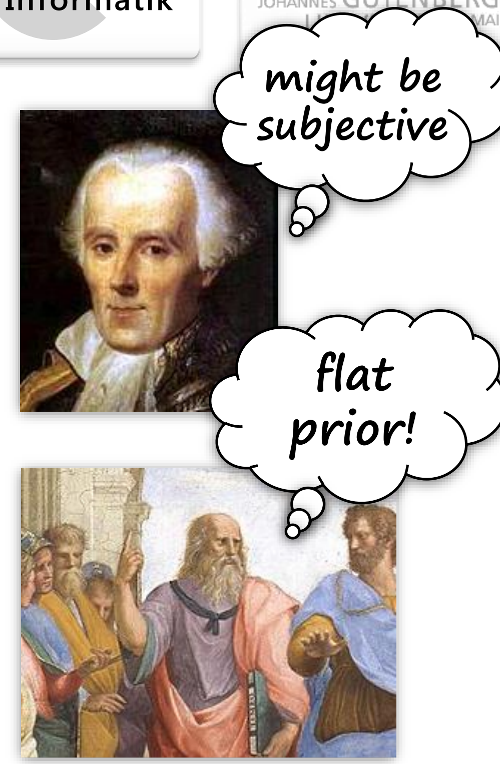
Bayesian Toolset

- **Conditioning:** We know something
- **Marginalization:** We disregard something
  - "Bayesian inference":
    Got a question, marginalize over everything not asked for
- **Chain rule:** Joint density from conditional & marginal
  - Build $p(x, y)$ from $p(x|y)$, $p(x)$
  - Stepwise modeling
- **Bayes rule:** Flip conditional
  - Build $p(y, x)$ from $p(x|y)$, $p(y)$
  - Interpret measurement/observation

Modelling 2

STATISTICAL DATA MODELLING





Chapter 4
Statistics and Machine Learning

Michael Wand · Institut für Informatik, JGU Mainz · michael.wand@uni-mainz.de

Video #04

Statistics & Machine Learning

- **Machine Learning Basics**
- **Bayesian Inference for ML**
- **Learning & Inference**

Let’s say we have a model already…

Inference

Inference

Model

$$P_\theta(X|D) = \frac{P_\theta(D|X)P_\theta(X)}{P_\theta(D)}$$

Situation

- We know the model parameters θ (e.g. classifier par.)
  - Fixed during inference
  - Determined during learning
- We have observed data D (e.g. photo of fruit)
- We want to infer X (e.g. class of fruit)

Three Variants

Model

$$P_\theta(X|D) = \frac{P_\theta(D|X)P_\theta(X)}{P_\theta(D)}$$

Inference Schemes

- Maximum Likelihood (simplest)
- Maximum-a-posteriori (with prior)
- Bayesian inference (most fancy, but often intractable)

Maximum Likelihood Estimation

Maximum Likelihood

Fixed Parameters θ

$$P_\theta(X|D) = \frac{P_\theta(D|X)P_\theta(X)}{P_\theta(D)}$$

$$\sim P_\theta(D|X)\cancel{P_\theta(X)}$$

$$= P_\theta(D|X)$$

$$\hat{X} = \underset{X \in \Omega(X)}{\arg\max}\, \underbrace{P_\theta(D|X)}_{\text{data term (likelihood) only}}$$

ML-Estimation (MLE)

- Only data likelihood, maximize for best X
  - Ignore prior, or uniform (pseudo-) prior
  - Model must be from restrictive family

Maximum-A-Posteriori (MAP) Estimation

Maximum-A-Posteriori (MAP)

Fixed model parameters θ

$$P_\theta(X|D) = \frac{P_\theta(D|X)P_\theta(X)}{P_\theta(D)} \sim P_\theta(D|X)P_\theta(X) \quad \Bigg\} \quad \hat{X} = \arg\max_{X\in\Omega(X)} \underbrace{P_\theta(D|X)P_\theta(X)}_{\text{posterior distribution (unnormalized)}}$$

MAP-Estimation

- Maximize for best X
  - Prior $P_\theta(X)$ non-trivial: X can be from overly flexible family
- Prior will fill in missing information
  - Can solve ill-posed problems, weak data term $P_\theta(D|X)$

Inference

Numerical trick for MAP/MLE

- Obtain X by maximizing

$$P(X|D) \sim P(D|X)P(X)$$

- Neg-log likelihoods: $E(\cdot) = -\ln P(\cdot)$

$$E(X|D) \sim E(D|X) + E(X)$$

← notation $E(\cdot)$ used in Mod-1 (variational modeling)

Useful for i.i.d. data

$$P(D|X) = \prod_{i=1}^{n} P(\mathbf{d}_i|X) \quad \rightarrow \quad E(D|X) = \sum_{i=1}^{n} E(\mathbf{d}_i|X)$$

“Bayesian Inference”

Bayesian Inference

Marginalization: Solution is the mean

$$\bar{X} = \mathbb{E}_{X \sim P_\theta(X|D)}[X]$$

$$= \int_{\mathbf{x} \in \Omega(X)} \mathbf{x} \cdot \frac{P_\theta(D|\mathbf{x}) P_\theta(\mathbf{x})}{P_\theta(D)} d\mathbf{x}$$

Determine $X = \bar{X}$ by marginalization

- Average all solutions (can be expensive)
  - Weight by posterior
  - Same as estimation for simple posteriors (e.g., Gaussian)
- Requires "proper" normalization; no neg-log tricks

ML & MAP Learning

(ML/MAP Parameter Estimation)

Maximum Likelihood

Maximum likelihood parameter estimation

$$P_\theta(X, D) = P_\theta(D|X) P_\theta(X)$$

$$\hat{\theta} = \arg\max_{\theta \in \Omega(\theta)} P_\theta(D|X) P_\theta(X)$$

properly normalized, $\int = 1$

Idea

- Maximize likelihood of observed data under model
  - Attention: Need properly normalized densities!
    - Normalization usually depends on θ
    - Thus, cannot be neglected
    - Often serious computational problem
- Optional prior on X, no prior on θ

Maximum A Posteriori

Maximum a posteriori parameter estimation

$$P(\theta|(X,D)) = \frac{P((X,D)|\theta)P(\theta)}{P((X,D))}$$

$$\hat{\theta} = \arg\max_{\theta \in \Omega(\theta)} P((X,D)|\theta)P(\theta)$$

properly normalized, $\int = 1$

often $P(X,D|\theta) = P(D|X,\theta)P(X|\theta)$ is used

Idea

- Add a prior on θ
- Use Bayes' rule to determine posterior on θ
- Again, $P((X,D)|\theta)$ must be normalized correctly
  - Scale factor usually depends on θ

Learning via Bayesian Inference

Bayesian Inference

Bayesians just ask simple*) questions:

$$\bar{\theta} = \mathbb{E}[\theta \cdot P(\theta|D)]$$

If you wanted Bayesian estimates:

$$\hat{\theta} = \arg\max_{\theta} \left(P(\theta|D)\right)$$

- This is not the same as „simple“ MAP
  - We will see later why / how this works differently

*) Computational costs might skyrocket. Terms and limitations apply.

Bayesian Inference

Bayesians just ask simple*) questions:

$$P(\theta|D) = \underbrace{P(D|\theta)}_{\text{"marginal likelihood" (given } \theta)}\underbrace{P(\theta)}_{\text{prior on } \theta}/P(D)$$

*) Computational costs might skyrocket. Terms and limitations apply.

Bayesian Inference

Bayesians just ask simple*) questions:

$$P(\theta|D) \sim P(\theta) \int_{\Omega(X)} P(D, X|\theta)\, dX$$

marginalization
of latent variable X

Marginal Likelihood

Bayesian Inference

Bayesians just ask simple*) questions:

$$P(\theta|D) \sim P(\theta) \int_{\Omega(X)} P(D, X|\theta)\, dX$$

chain rule

$$\sim P(\theta) \int_{\Omega(X)} \underbrace{P(D|X,\theta)}_{\text{likelihood}} \underbrace{P(X|\theta)}_{\text{prior on } X}\, dX$$

prior on θ

Marginal Likelihood

Pros & Cons

- Good: Improved generalization behavior (more later)
- Bad: Integral over inferred model X might be infeasible

Remark: Break in Notation

Break in Naming

- No longer looking for a probabilistic mapping

  $$D \to X, \text{ i.e. learn } \mathrm{P}(X|D)$$

- In the discussion of Bayesian learning, only D is data
  - We have computed

    $$P(\theta|D) \sim P(\theta) \int_{\Omega(X)} P(D, X|\theta)\, dX$$

  - So X is not fixed
- Only D is training data
  - Otherwise, substitute training data D by pairs (D, X)
  - Likelihood must measure the fit

So, again

Simple question:

$$P(\theta|(D,X)) \sim P(\theta)\int_{\Omega((D,X))} P((D,X),(D',X')|\theta)\, d(D',X')$$

Marginal Likelihood

Bayesian Inference

Calculation (notation tedious):

$$P(\theta|(D,X)) \sim P(\theta) \int_{\Omega((D,X))} P((D,X),(D',X')|\theta)\, d(D',X')$$

$$= P(\theta) \int_{\Omega(X)} P((D',X')|(D,X),\theta) P((D',X')|\theta)\, d(D',X')$$

Looks awful, but is actually simple

- Integrate over likelihood of predictions

Video #04c

Summary

Summary

Answers to questions

- Maximum Likelihood Estimation (MLE)
- Maximum A Priori (MAP) Estimation
- Bayesian inference

Two modes

- Inference (fixed model parameters θ)
- Training/learning (of θ)

Computational Hurdles

General Model

$$P_\theta(X|D) = \frac{P_\theta(D|X)P_\theta(X)}{P_\theta(D)}$$

MLE/MAP Inference (θ fixed)

- Can ignore denominator
- Can use unnormalized densities

MLE / MAP

Maximum search on log-density

MLE/MAP Learning (θ fixed)

- Denominator counts (usually depends on θ)
- Careful with normalization (dependence on θ)

Computational Hurdles

General Model

$$P_\theta(X|D) = \frac{P_\theta(D|X)P_\theta(X)}{P_\theta(D)}$$

Bayesian Inference

Integration

Bayesian Inference of X/θ

- Need high-dimensional integration
- Need to be careful to weight everything correctly
  - Normalization of numerator affects weight
- Log-space computations usually do not help
- Learning:
  Again – be careful with dependencies on θ