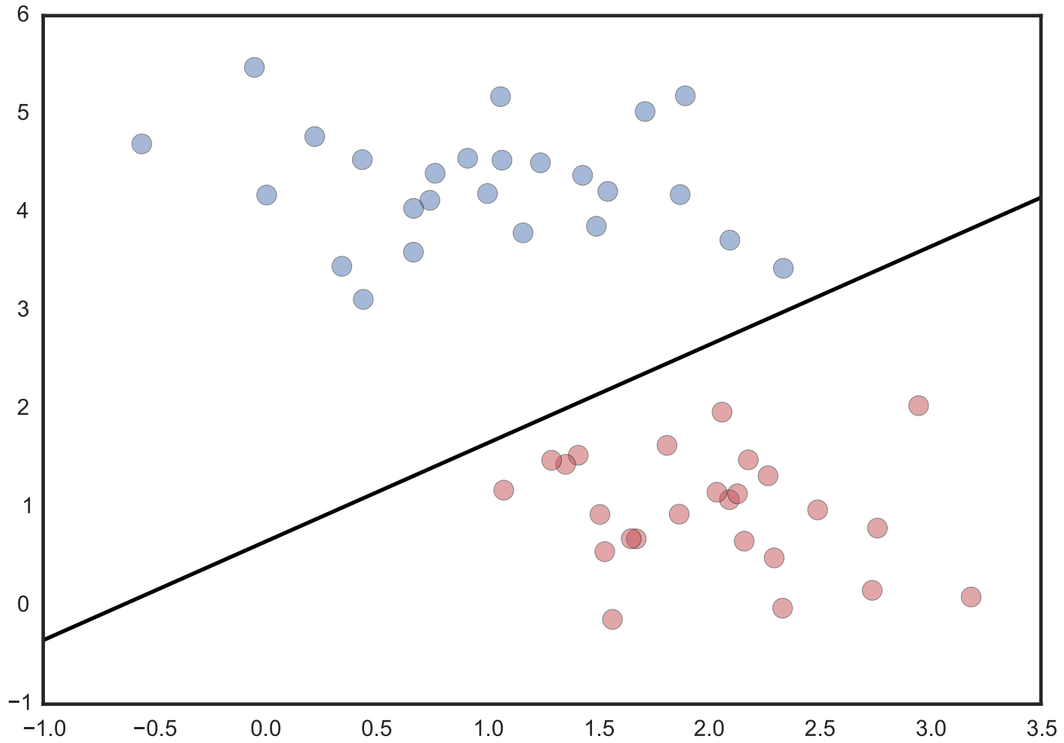
Generative Models

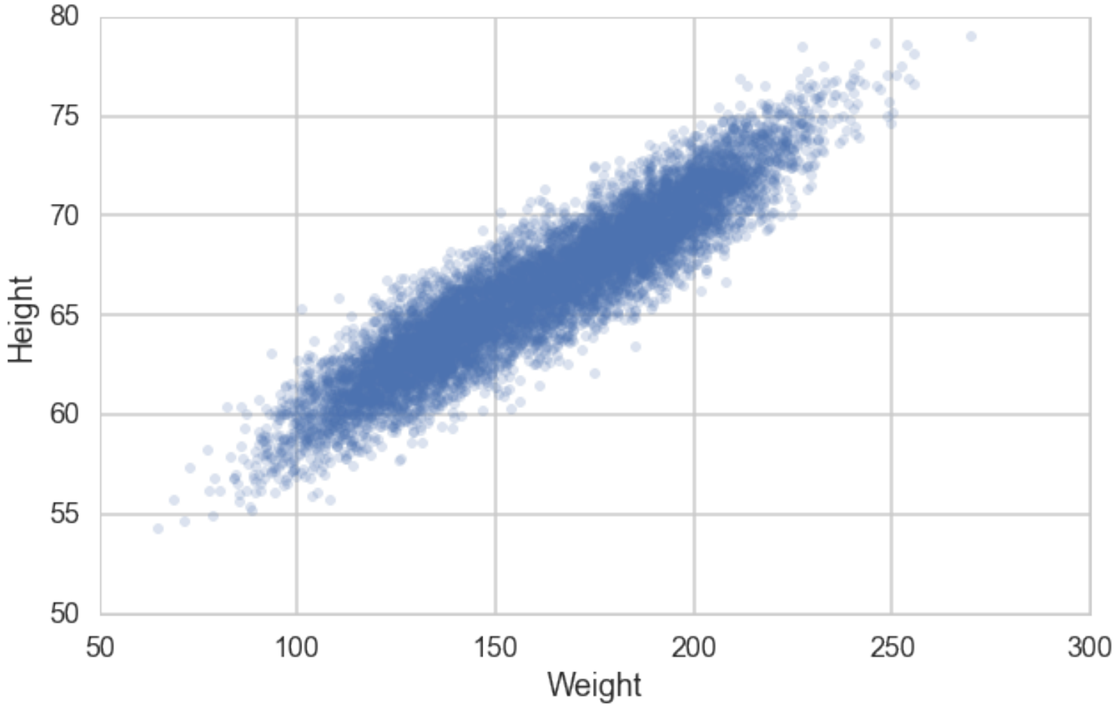
CLASSIFICATION

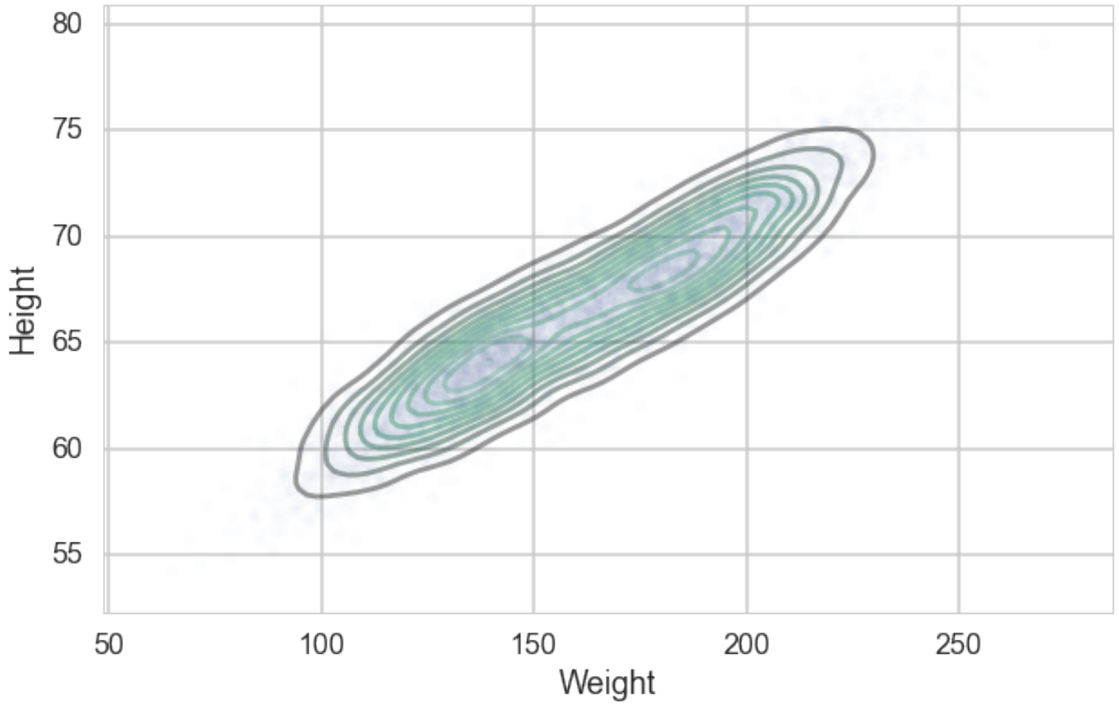


- will a customer churn?
- is this a check? For how much?
- a man or a woman?
- will this customer buy?
- do you have cancer?
- is this spam?
- whose picture is this?
- what is this text about?[j]

[j] image from code in http://bit.ly/1Azg29G

PROBABILISTIC CLASSIFICATION





In any machine learning problem we want to model $p(x, y)$.

We can choose to model as

$$p(x, y) = p(y \mid x)p(x) \text{ or } p(x \mid y)p(y)$$

In regression we modeled the former. In logistic regression, with $y = c$ (class c) we model the former as well. This is the probability of the class given the features x.

In "Generative models" we model the latter, the probability of the features fiven the class.

The conditional probabilities of $y = 1$ or $y = 0$ given a particular sample's features $\mathbf{x}$ are:

$$P(y = 1|\mathbf{x}) = h(\mathbf{w} \cdot \mathbf{x})$$
$$P(y = 0|\mathbf{x}) = 1 - h(\mathbf{w} \cdot \mathbf{x}).$$

These two can be written together as

$$P(y|\mathbf{x}, \mathbf{w}) = h(\mathbf{w} \cdot \mathbf{x})^y (1 - h(\mathbf{w} \cdot \mathbf{x}))^{(1-y)}$$

BERNOULLI!!

Multiplying over the samples we get:

$$P(y|\mathbf{x},\mathbf{w}) = P(\{y_i\}|\{\mathbf{x}_i\},\mathbf{w}) = \prod_{y_i \in \mathcal{D}} P(y_i|\mathbf{x}_i,\mathbf{w}) = \prod_{y_i \in \mathcal{D}} h(\mathbf{w}\cdot\mathbf{x}_i)^{y_i}\left(1-h(\mathbf{w}\cdot\mathbf{x}_i)\right)^{(1-y_i)}$$

Indeed its important to realize that a particular sample can be thought of as a draw from some "true" probability distribution.

maximum likelihood estimation maximises the **likelihood of the sample y**, or alternately the log-likelihood,

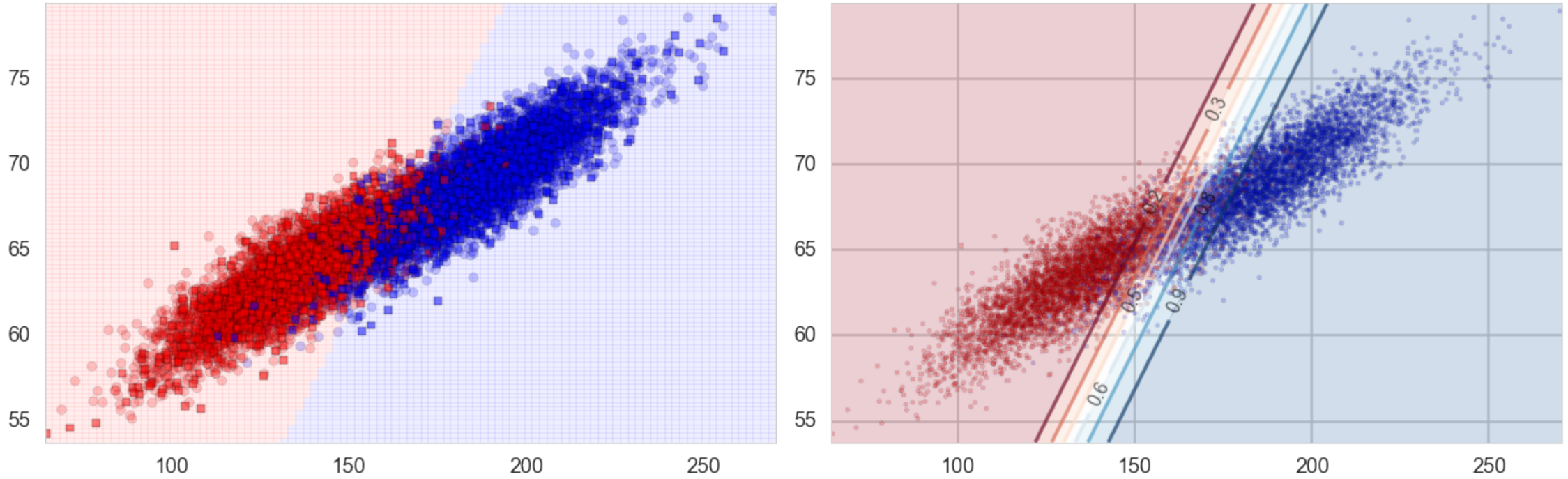
$$\mathcal{L} = P(y \mid \mathbf{x},\mathbf{w}). \text{ OR } \ell = log(P(y \mid \mathbf{x},\mathbf{w}))$$

Thus

$$
\begin{aligned}
\ell &= log\left(\prod_{y_i \in \mathcal{D}} h(\mathbf{w}\cdot\mathbf{x}_i)^{y_i}\left(1-h(\mathbf{w}\cdot\mathbf{x}_i)\right)^{(1-y_i)}\right) \\
&= \sum_{y_i \in \mathcal{D}} log\left(h(\mathbf{w}\cdot\mathbf{x}_i)^{y_i}\left(1-h(\mathbf{w}\cdot\mathbf{x}_i)\right)^{(1-y_i)}\right) \\
&= \sum_{y_i \in \mathcal{D}} log\, h(\mathbf{w}\cdot\mathbf{x}_i)^{y_i} + log\left(1-h(\mathbf{w}\cdot\mathbf{x}_i)\right)^{(1-y_i)} \\
&= \sum_{y_i \in \mathcal{D}} (y_i log(h(\mathbf{w}\cdot\mathbf{x})) + (1-y_i) log(1-h(\mathbf{w}\cdot\mathbf{x})))
\end{aligned}
$$

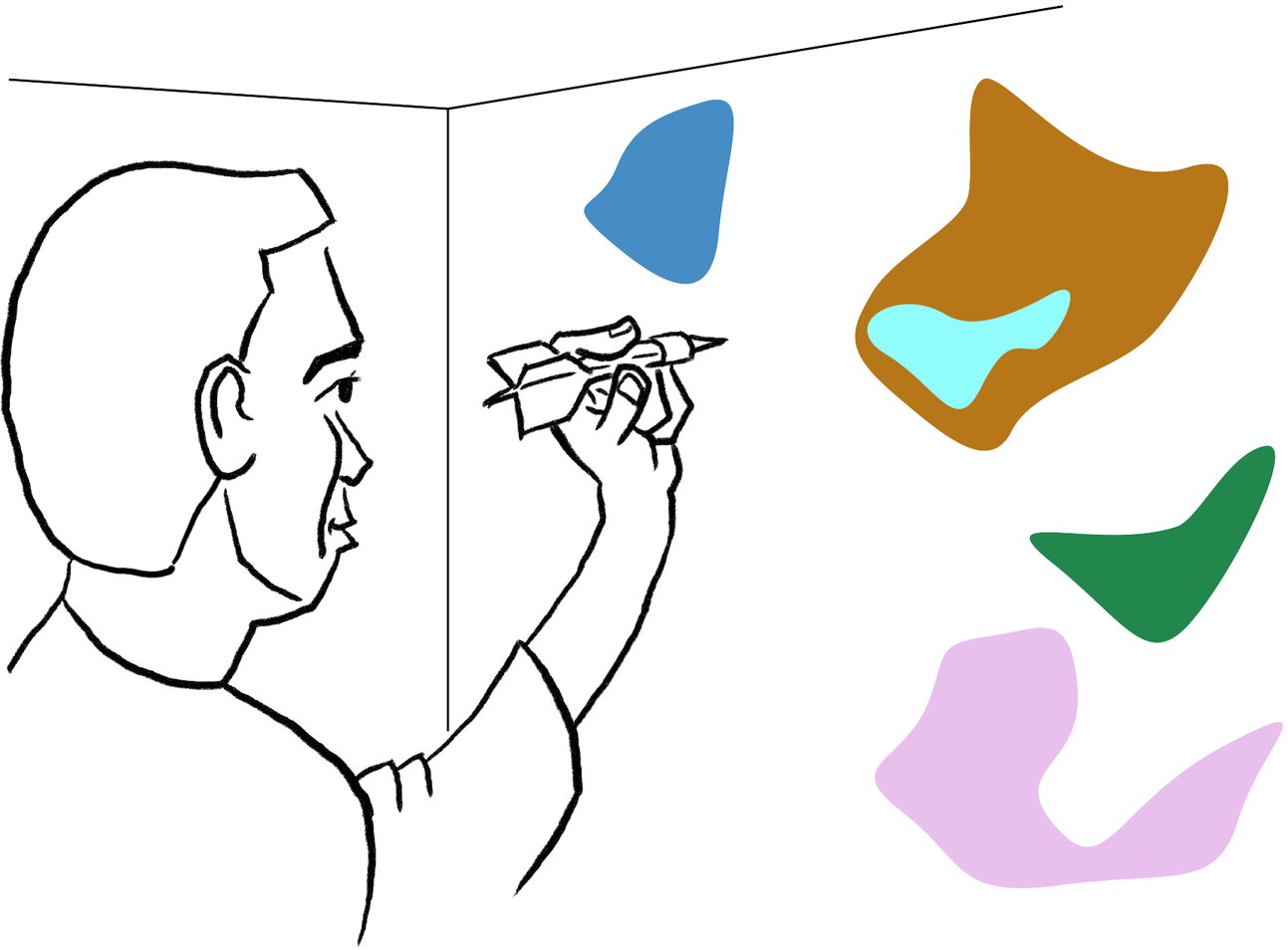
DISCRIMINATIVE CLASSIFIER

$$P(y|x) : P(male|height, weight)$$

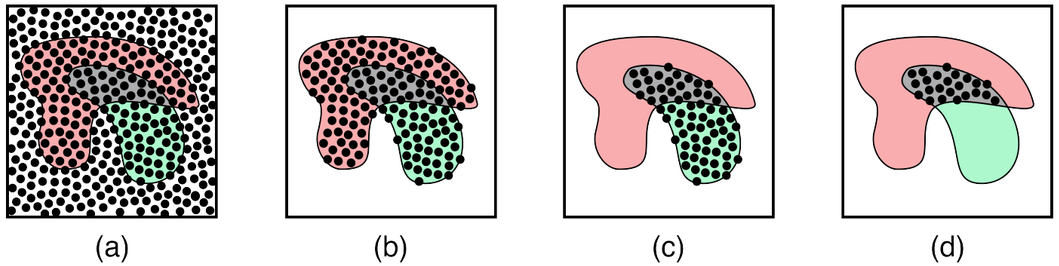


Discriminative Learning

- are these classifiers any good?
- they are discriminative and draw boundaries, but thats it
- they are cheaper to calculate but shed no insight
- would it not be better to have a classifier that captured the generative process



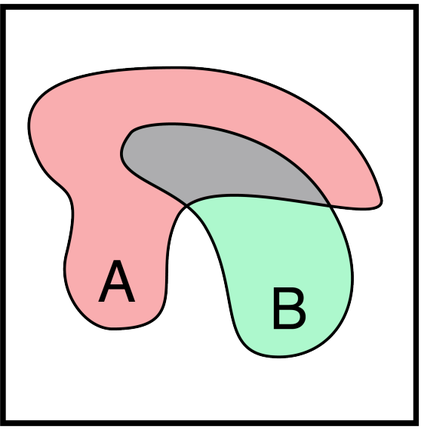
Throwing darts, uniformly



Throwing darts at the wall to find P(A|B). (a) Darts striking the wall. (b) All the darts in either A or B. (c) The darts only in B. (d) The darts that are in the overlap of A and B.

(pics like these from Andrew Glassner's book)

Conditional Probability



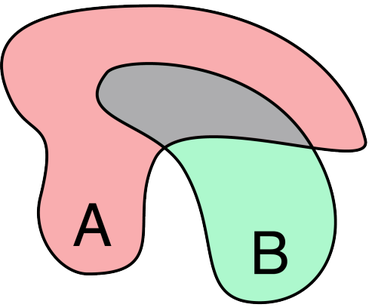
$$P(A|B) = \frac{\text{overlap}}{\text{B}}$$

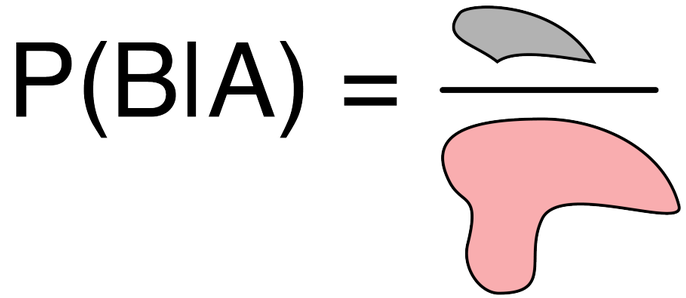
conditional probability tells us the chance that one thing will happen, given that another thing has already happened. In this case, we want to know the probability that our dart landed in blob A, given that we already know it landed in blob B.

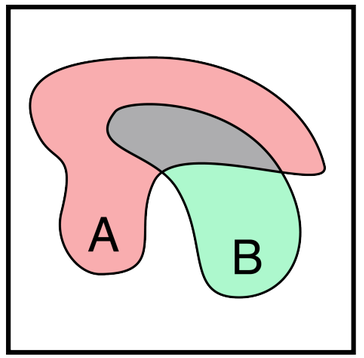
Other conditional and joint

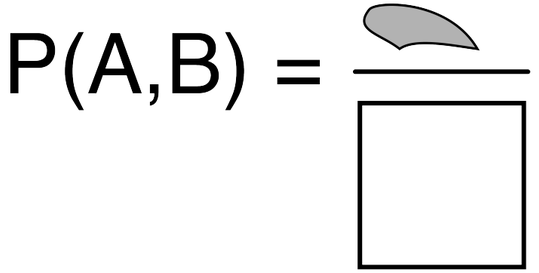
Left: the other conditional

Below: the joint probability $p(A, B)$, the chance that any randomly-thrown dart will land in both A and B at the same time.



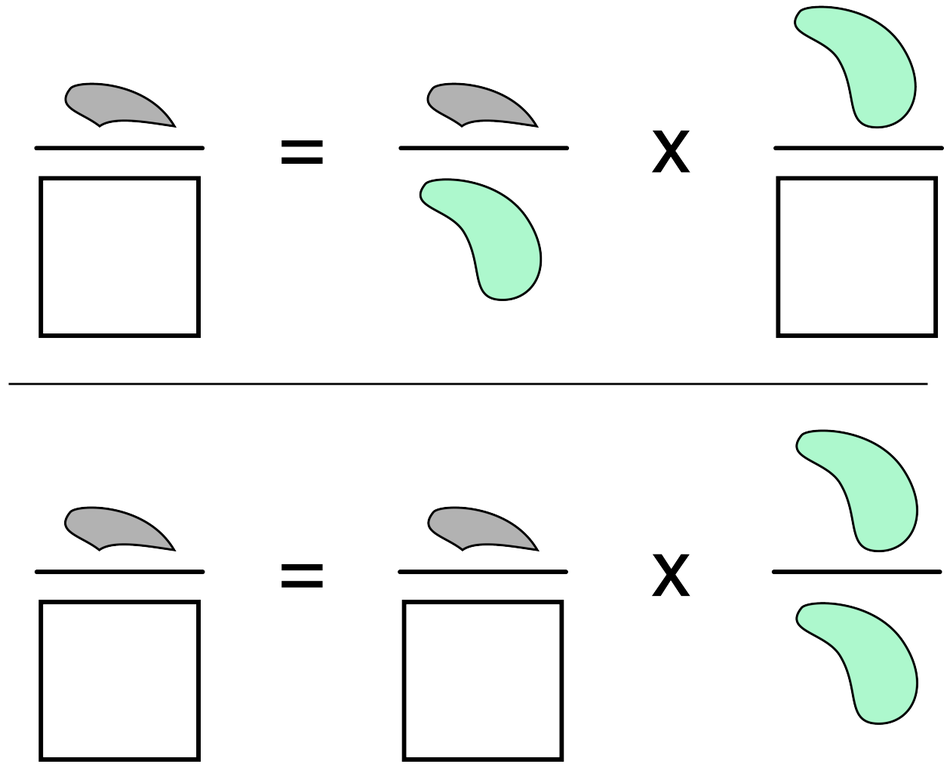




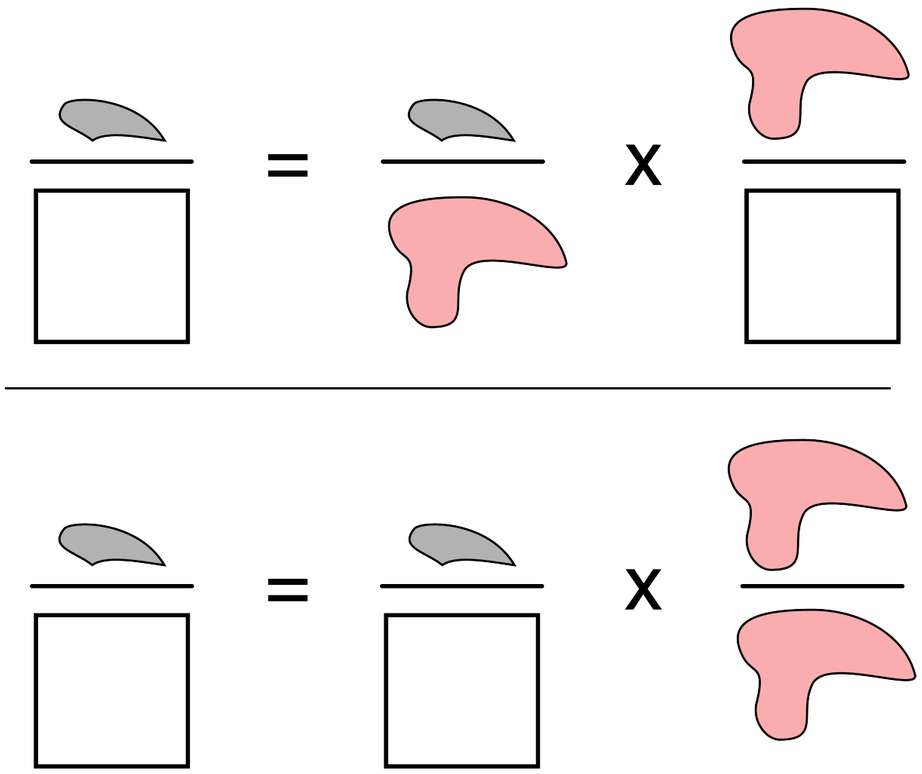


The joint probability can be written 2 ways

$$P(A,B) = P(A|B) \times P(B)$$



$$P(A,B) = P(B|A) \times P(A)$$



Bayes Theorem

Equating these gives us Bayes Theorem.

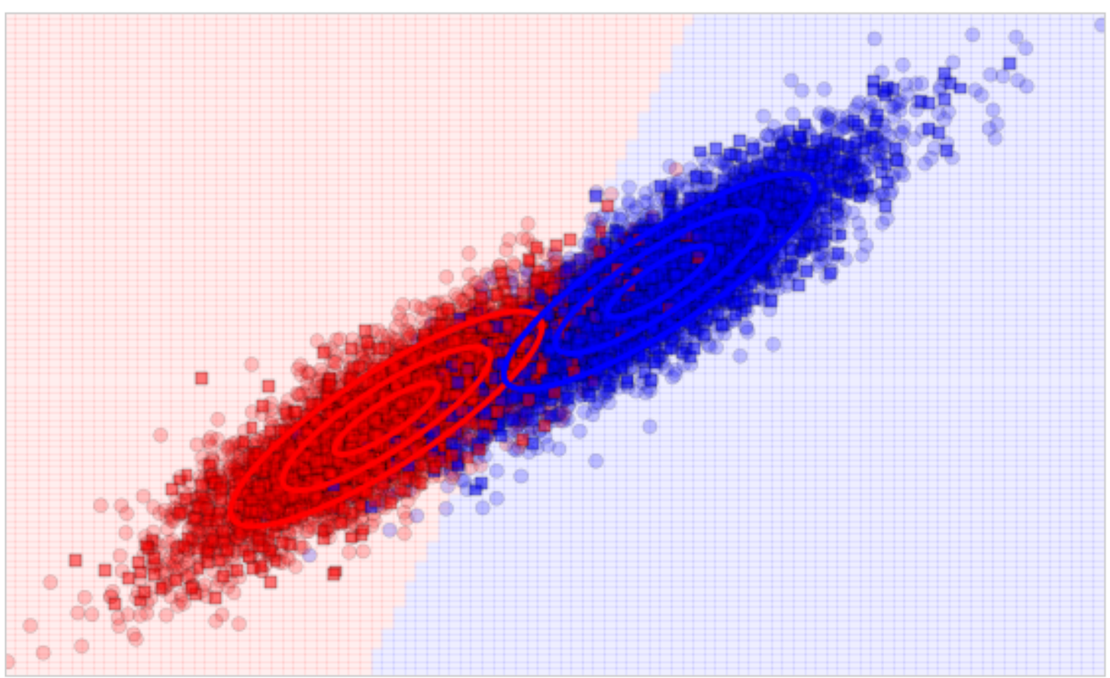
$$P(A \mid B)P(B) = P(B \mid A)P(A)$$

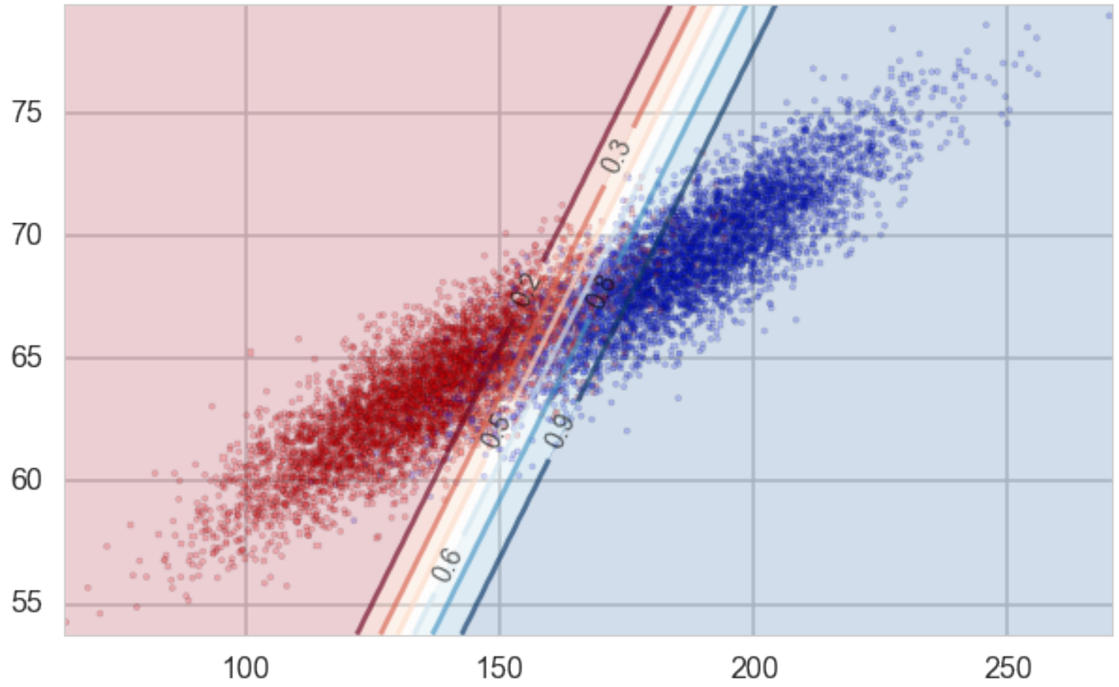
$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

the LHS probability $P(A \mid B)$is called the posterior, while P(A) is called the prior, and p(B) is called the evidence

GENERATIVE CLASSIFIER

$$P(y|x) \propto P(x|y)P(x) : P(height, weight|male) \times P(male)$$





Generative Classifier

For a feature vector x, we use Bayes rule to express the posterior of the class-conditional as:

$$p(c|x,\theta) = \frac{p(c|\theta)p(x|c,\theta)}{\sum_{c'} p(zc'|\theta)p(x|c',\theta)}$$

This is a **generative classifier**, since it specifies how to generate the data using the class-conditional density $p(x|c,\theta)$ and the class prior $p(c|\theta)$.

Representation Learning

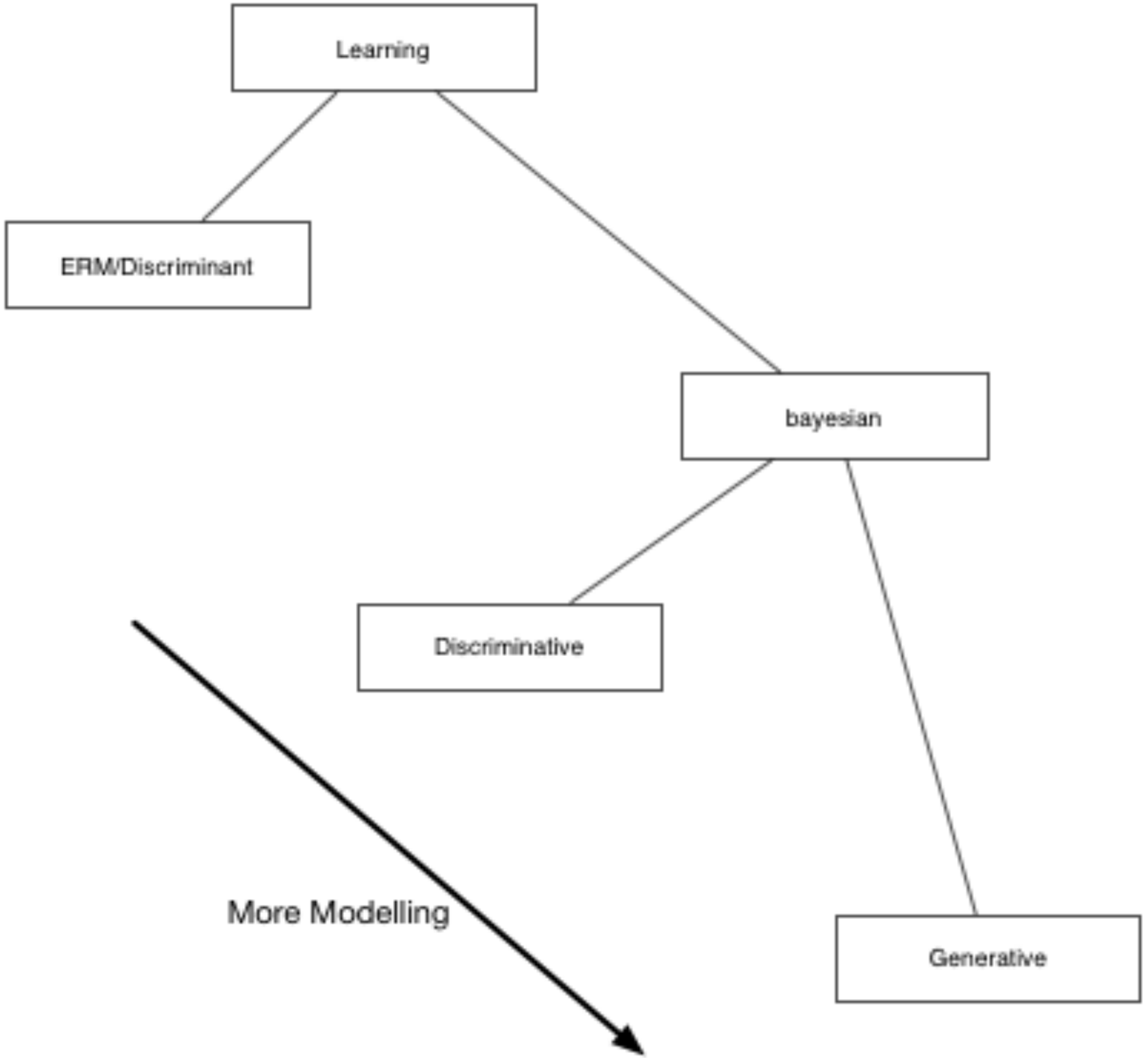
- the idea of generative learning is to capture an underlying representation (compressed) of the data
- in the previous slide it was 2 normal distributions
- generally more complex, but the idea if to fit a "generative" model whose parameters represent the process
- besides gpus and autodiff , this is the third pillar of the AI rennaissance: the choice of better representations: e.g. convolutions

Generative vs Discriminative classifiers

- LDA vs logistic respectively.
- LDA is generative as it models $p(x|c)$ while logistic models $p(c|x)$ directly. Here think of $\mathbf{z} = c$
- we do know c on the training set, so think of the unsupervised learning counterparts of these models where you dont know c

Generative vs Discriminative classifiers (contd)

- generative handles data asymmetry better
- sometimes generative models like LDA and Naive Bayes are easy to fit. Discriminative models require convex optimization via Gradient descent
- can add new classes to a generative classifier without retraining so better for online customer selection problems
- generative classifiers can handle missing data easily
- generative classifiers are better at handling unlabelled training data (semi-supervized learning)
- preprocessing data is easier with discriminative classifiers
- discriminative classifiers give generally better callibrated probabilities
- discriminative usually less expensive


Learning
ERM/Discriminant
bayesian
Discriminative
Generative
More Modelling