Classification (overview)

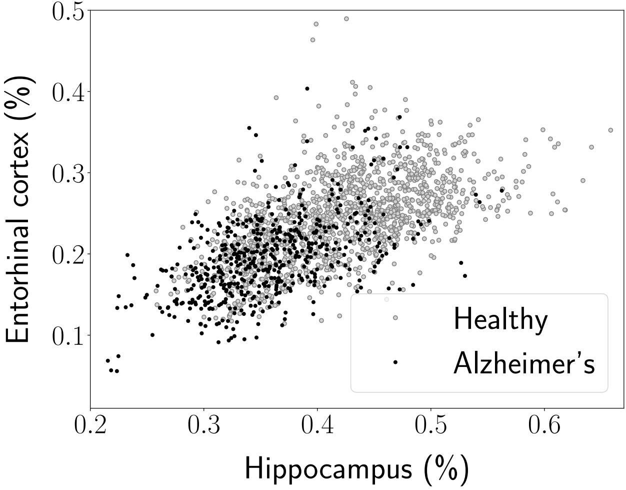
Probability and Statistics for Data Science

Carlos Fernandez-Granda

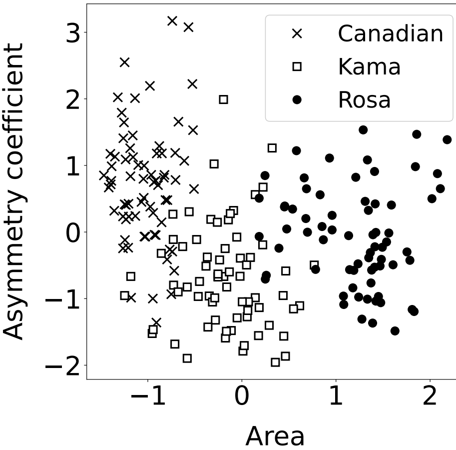
Classification

Goal: Assign class to data based on features

Alzheimer's diagnostics



Identification of wheat varieties



U.S. House of Representatives in 1984

Predict political affiliation (Republican or Democrat) from voting record on 16 issues

Probabilistic modeling

Probabilistic modeling

Class: Random variable $\tilde{y}$ with range Y

Probabilistic modeling

Class: Random variable $\tilde{y}$ with range Y

Features: Random vector $\tilde{x}$ with range $\mathcal{X}$

Probabilistic modeling

Class: Random variable $\tilde{y}$ with range Y

Features: Random vector $\tilde{x}$ with range $\mathcal{X}$

Maximum a posteriori (MAP) estimator of $\tilde{y}$ given $\tilde{x} = x \in \mathcal{X}$

$$\text{MAP}(x) := \arg\max_{y \in Y} p_{\tilde{y} \mid \tilde{x}}(y \mid x)$$

MAP estimation is optimal

MAP estimation is optimal

For any other estimator h

$$\mathrm{P}\left(\tilde{y} = h(\tilde{x})\right)$$

$$\leq$$

$$\mathrm{P}\left(\tilde{y} = \mathsf{MAP}(\tilde{x})\right)$$

MAP estimation is optimal

For any other estimator h

$$\mathrm{P}\left(\tilde{y} = h(\tilde{x})\right) = \sum_{x \in \mathcal{X}} \mathrm{P}\left(\tilde{x} = x, \tilde{y} = h\left(\tilde{x}\right)\right)$$

$$\leq$$

$$\mathrm{P}\left(\tilde{y} = \mathsf{MAP}(\tilde{x})\right)$$

MAP estimation is optimal

For any other estimator h

$$
\begin{aligned}
\mathrm{P}\left(\tilde{y} = h(\tilde{x})\right) &= \sum_{x \in \mathcal{X}} \mathrm{P}\left(\tilde{x} = x, \tilde{y} = h\left(\tilde{x}\right)\right) \\
&= \sum_{x \in \mathcal{X}} \mathrm{P}\left(\tilde{x} = x\right) \mathrm{P}\left(\tilde{y} = h\left(x\right) \mid \tilde{x} = x\right) \\
&\leq
\end{aligned}
$$

$$\mathrm{P}\left(\tilde{y} = \mathsf{MAP}(\tilde{x})\right)$$

MAP estimation is optimal

For any other estimator h

$$
\begin{aligned}
\mathrm{P}\left(\tilde{y} = h(\tilde{x})\right) &= \sum_{x \in \mathcal{X}} \mathrm{P}\left(\tilde{x} = x, \tilde{y} = h\left(\tilde{x}\right)\right) \\
&= \sum_{x \in \mathcal{X}} \mathrm{P}\left(\tilde{x} = x\right) \mathrm{P}\left(\tilde{y} = h\left(x\right) \mid \tilde{x} = x\right) \\
&= \sum_{x \in \mathcal{X}} p_{\tilde{x}}(x) p_{\tilde{y} \mid \tilde{x}}(h\left(x\right) \mid x) \\
&\leq
\end{aligned}
$$

$$\mathrm{P}\left(\tilde{y} = \mathrm{MAP}(\tilde{x})\right)$$

MAP estimation is optimal

For any other estimator h

$$\begin{aligned}
\mathrm{P}\left(\tilde{y} = h(\tilde{x})\right) &= \sum_{x \in \mathcal{X}} \mathrm{P}\left(\tilde{x} = x, \tilde{y} = h\left(\tilde{x}\right)\right) \\
&= \sum_{x \in \mathcal{X}} \mathrm{P}\left(\tilde{x} = x\right) \mathrm{P}\left(\tilde{y} = h\left(x\right) \mid \tilde{x} = x\right) \\
&= \sum_{x \in \mathcal{X}} p_{\tilde{x}}(x) p_{\tilde{y} \mid \tilde{x}}(h\left(x\right) \mid x) \\
&\leq \sum_{x \in \mathcal{X}} p_{\tilde{x}}(x) p_{\tilde{y} \mid \tilde{x}}(\mathrm{MAP}\left(x\right) \mid x)
\end{aligned}$$

$$\mathrm{P}\left(\tilde{y} = \mathrm{MAP}(\tilde{x})\right)$$

MAP estimation is optimal

For any other estimator h

$$
\begin{aligned}
\mathrm{P}\left(\tilde{y} = h(\tilde{x})\right) &= \sum_{x \in \mathcal{X}} \mathrm{P}\left(\tilde{x} = x, \tilde{y} = h\left(\tilde{x}\right)\right) \\
&= \sum_{x \in \mathcal{X}} \mathrm{P}\left(\tilde{x} = x\right) \mathrm{P}\left(\tilde{y} = h\left(x\right) \mid \tilde{x} = x\right) \\
&= \sum_{x \in \mathcal{X}} p_{\tilde{x}}(x) p_{\tilde{y} \mid \tilde{x}}(h\left(x\right) \mid x) \\
&\leq \sum_{x \in \mathcal{X}} p_{\tilde{x}}(x) p_{\tilde{y} \mid \tilde{x}}(\mathrm{MAP}\left(x\right) \mid x) \\
&= \sum_{x \in \mathcal{X}} \mathrm{P}\left(\tilde{x} = x\right) \mathrm{P}\left(\tilde{y} = \mathrm{MAP}\left(x\right) \mid \tilde{x} = x\right) \\
&= \mathrm{P}\left(\tilde{y} = \mathrm{MAP}(\tilde{x})\right)
\end{aligned}
$$

MAP estimation is optimal

For any other estimator h

$$\begin{aligned}
\mathrm{P}\left(\tilde{y} = h(\tilde{x})\right) &= \sum_{x \in \mathcal{X}} \mathrm{P}\left(\tilde{x} = x, \tilde{y} = h\left(\tilde{x}\right)\right) \\
&= \sum_{x \in \mathcal{X}} \mathrm{P}\left(\tilde{x} = x\right) \mathrm{P}\left(\tilde{y} = h\left(x\right) \mid \tilde{x} = x\right) \\
&= \sum_{x \in \mathcal{X}} p_{\tilde{x}}(x) p_{\tilde{y} \mid \tilde{x}}(h\left(x\right) \mid x) \\
&\leq \sum_{x \in \mathcal{X}} p_{\tilde{x}}(x) p_{\tilde{y} \mid \tilde{x}}(\mathsf{MAP}\left(x\right) \mid x) \\
&= \sum_{x \in \mathcal{X}} \mathrm{P}\left(\tilde{x} = x\right) \mathrm{P}\left(\tilde{y} = \mathsf{MAP}\left(x\right) \mid \tilde{x} = x\right) \\
&= \mathrm{P}\left(\tilde{y} = \mathsf{MAP}(\tilde{x})\right)
\end{aligned}$$

Are we done here?

Prediction of political affiliation

16 binary features (Yes/No votes)

Prediction of political affiliation

16 binary features (Yes/No votes)

Can we estimate $p_{\tilde{y} \mid \tilde{x}}(\cdot \mid x)$ for any x?

Prediction of political affiliation

16 binary features (Yes/No votes)

Can we estimate $p_{\tilde{y}\,|\,\tilde{x}}(\cdot\,|\,x)$ for any x?

Possible values of x?

Prediction of political affiliation

16 binary features (Yes/No votes)

Can we estimate $p_{\tilde{y} \mid \tilde{x}}(\cdot \mid x)$ for any x?

Possible values of x? $2^{16} = 65{,}536$

Prediction of political affiliation

16 binary features (Yes/No votes)

Can we estimate $p_{\tilde{y}\,|\,\tilde{x}}(\cdot\,|\,x)$ for any x?

Possible values of x? $2^{16} = 65{,}536$

We have 425 training examples...

Prediction of political affiliation

16 binary features (Yes/No votes)

Can we estimate $p_{\tilde{y}\,|\,\tilde{x}}(\cdot\,|\,x)$ for any x? **No**

Possible values of x? $2^{16} = 65{,}536$

We have 425 training examples...

Prediction of political affiliation

16 binary features (Yes/No votes)

Can we estimate $p_{\tilde{y}\,|\,\tilde{x}}(\cdot\,|\,x)$ for any x? **No**

Possible values of x? $2^{16} = 65{,}536$

We have 425 training examples...

We need assumptions to build tractable approximations

Plan

Generative Models

Discriminative Models

Evaluation

Generative Models

Discriminative Models

Evaluation

Generative classification models

Goal: Estimate conditional pmf $p_{\tilde{y} \mid \tilde{x}}$ of class given features

Generative classification models

Goal: Estimate conditional pmf $p_{\tilde{y} \mid \tilde{x}}$ of class given features

1. Use data to approximate

Generative classification models

Goal: Estimate conditional pmf $p_{\tilde{y} \mid \tilde{x}}$ of class given features

1. Use data to approximate
   - Pmf $p_{\tilde{y}}$ of class

Generative classification models

Goal: Estimate conditional pmf $p_{\tilde{y} \mid \tilde{x}}$ of class given features

1. Use data to approximate
   - Pmf $p_{\tilde{y}}$ of class
   - Conditional pmf $p_{\tilde{x} \mid \tilde{y}}$ or conditional pdf $f_{\tilde{x} \mid \tilde{y}}$ of features given class

Generative classification models

Goal: Estimate conditional pmf $p_{\tilde{y} \mid \tilde{x}}$ of class given features

1. Use data to approximate
   - Pmf $p_{\tilde{y}}$ of class
   - Conditional pmf $p_{\tilde{x} \mid \tilde{y}}$ or conditional pdf $f_{\tilde{x} \mid \tilde{y}}$ of features given class

2. Apply Bayes' rule

$$p_{\tilde{y} \mid \tilde{x}}(y \mid x) = \frac{p_{\tilde{y}}(y) p_{\tilde{x} \mid \tilde{y}}(x \mid y)}{p_{\tilde{x}}(x)} \quad \text{or} \quad \frac{p_{\tilde{y}}(y) f_{\tilde{x} \mid \tilde{y}}(x \mid y)}{f_{\tilde{x}}(x)}$$

Challenge

Approximating conditional (joint) pmf / pdf of features given class

Challenge

Approximating conditional (joint) pmf / pdf of features given class

- Naive Bayes:
  Features are **conditionally independent** given class

Challenge

Approximating conditional (joint) pmf / pdf of features given class

- Naive Bayes:
  Features are **conditionally independent** given class

- Gaussian discriminant analysis:
  Features are conditionally **Gaussian** given class

Prediction of political affiliation

Prediction of political affiliation

Class

$$\tilde{y} = \begin{cases} R & \text{Republican} \\ D & \text{Democrat} \end{cases}$$

Prediction of political affiliation

Class

$$\tilde{y} = \begin{cases} R & \text{Republican} \\ D & \text{Democrat} \end{cases}$$

Features ($1 \leq i \leq 16$)

$$\tilde{x}[i] = \begin{cases} 1 & \text{voted Yes on issue } i \\ 0 & \text{otherwise} \end{cases}$$

Naive Bayes

Naive Bayes

We assume votes are conditionally independent given affiliation

Naive Bayes

We assume votes are conditionally independent given affiliation

$$p_{\tilde{x} \mid \tilde{y}}(x \mid R) = \prod_{i=1}^{d} p_{\tilde{x}[i] \mid \tilde{y}}(x[i] \mid R)$$

Naive Bayes

We assume votes are conditionally independent given affiliation

$$p_{\tilde{x} \mid \tilde{y}}(x \mid R) = \prod_{i=1}^{d} p_{\tilde{x}[i] \mid \tilde{y}}(x[i] \mid R)$$

$$p_{\tilde{x} \mid \tilde{y}}(x \mid D) = \prod_{i=1}^{d} p_{\tilde{x}[i] \mid \tilde{y}}(x[i] \mid D)$$

Bayes rule

$$p_{\tilde{y} \mid \tilde{x}}(R \mid x)$$

$$=$$

Bayes rule

$$p_{\tilde{y} \mid \tilde{x}}(R \mid x)$$
$$= \frac{p_{\tilde{y},\tilde{x}}(R, x)}{p_{\tilde{x}}(x)}$$

Bayes rule

$$
\begin{aligned}
& p_{\tilde{y} \mid \tilde{x}}(R \mid x) \\
& = \frac{p_{\tilde{y},\tilde{x}}(R, x)}{p_{\tilde{x}}(x)} \\
& = \frac{p_{\tilde{y}}(R) p_{\tilde{x} \mid \tilde{y}}(x \mid R)}{p_{\tilde{y},\tilde{x}}(R, x) + p_{\tilde{y},\tilde{x}}(D, x)}
\end{aligned}
$$

Bayes rule

$$
\begin{aligned}
& p_{\tilde{y} \mid \tilde{x}}(R \mid x) \\
& = \frac{p_{\tilde{y},\tilde{x}}(R, x)}{p_{\tilde{x}}(x)} \\
& = \frac{p_{\tilde{y}}(R) p_{\tilde{x} \mid \tilde{y}}(x \mid R)}{p_{\tilde{y},\tilde{x}}(R, x) + p_{\tilde{y},\tilde{x}}(D, x)} \\
& = \frac{p_{\tilde{y}}(R) \prod_{i=1}^{d} p_{\tilde{x}[i] \mid \tilde{y}}(x[i] \mid R)}{p_{\tilde{y}}(R) \prod_{i=1}^{d} p_{\tilde{x}[i] \mid \tilde{y}}(x[i] \mid R) + p_{\tilde{y}}(D) \prod_{i=1}^{d} p_{\tilde{x}[i] \mid \tilde{y}}(x[i] \mid D)}
\end{aligned}
$$

Estimated probabilities

$p_{\tilde{y}}(R) = 0.381 \quad (p_{\tilde{y}}(D) = 0.619)$

Estimated probabilities

$p_{\tilde{y}}(R) = 0.381 \qquad (p_{\tilde{y}}(D) = 0.619)$

i	1	2	3	4	5	6	7	8
$p_{\tilde{x}[i] \mid \tilde{y}}(1 \mid R)$	0.19	0.50	0.14	0.99	0.95	0.90	0.24	0.15
$p_{\tilde{x}[i] \mid \tilde{y}}(1 \mid D)$	0.61	0.50	0.89	0.05	0.22	0.47	0.78	0.83

i	9	10	11	12	13	14	15	16
$p_{\tilde{x}[i] \mid \tilde{y}}(1 \mid R)$	0.11	0.55	0.14	0.87	0.86	0.98	0.09	0.66
$p_{\tilde{x}[i] \mid \tilde{y}}(1 \mid D)$	0.76	0.47	0.51	0.15	0.29	0.35	0.64	0.94

Applying the model

i	1	2	3	4	5	6	7	8
$p_{\tilde{x}[i] \mid \tilde{y}}(1 \mid R)$	0.19	0.50	0.14	0.99	0.95	0.90	0.24	0.15
$p_{\tilde{x}[i] \mid \tilde{y}}(1 \mid D)$	0.61	0.50	0.89	0.05	0.22	0.47	0.78	0.83
Example	N	–	**Y**	**N**	**N**	Y	**Y**	**Y**

i	9	10	11	12	13	14	15	16
$p_{\tilde{x}[i] \mid \tilde{y}}(1 \mid R)$	0.11	0.55	0.14	0.87	0.86	0.98	0.09	0.66
$p_{\tilde{x}[i] \mid \tilde{y}}(1 \mid D)$	0.76	0.47	0.51	0.15	0.29	0.35	0.64	0.94
Example	N	Y	N	**N**	**N**	**N**	**Y**	–

Applying the model

i	1	2	3	4	5	6	7	8
$p_{\tilde{x}[i]\,\vert\,\tilde{y}}(1\,\vert\,R)$	0.19	0.50	0.14	0.99	0.95	0.90	0.24	0.15
$p_{\tilde{x}[i]\,\vert\,\tilde{y}}(1\,\vert\,D)$	0.61	0.50	0.89	0.05	0.22	0.47	0.78	0.83
Example	N	–	**Y**	**N**	**N**	Y	**Y**	**Y**

i	9	10	11	12	13	14	15	16
$p_{\tilde{x}[i]\,\vert\,\tilde{y}}(1\,\vert\,R)$	0.11	0.55	0.14	0.87	0.86	0.98	0.09	0.66
$p_{\tilde{x}[i]\,\vert\,\tilde{y}}(1\,\vert\,D)$	0.76	0.47	0.51	0.15	0.29	0.35	0.64	0.94
Example	N	Y	N	**N**	**N**	**N**	**Y**	–

$$p_{\tilde{y}\,|\,\tilde{x}}(D\,|\,x)$$

$$= \frac{p_{\tilde{y}}(D) \prod_{i \in \{\mathbf{1},\mathbf{3},\ldots,\mathbf{15}\}} p_{\tilde{x}[i]\,|\,\tilde{y}}(x[i]\,|\,D)}{p_{\tilde{y}}(D) \prod_{i \in \{\mathbf{1},\mathbf{3},\ldots,\mathbf{15}\}} p_{\tilde{x}[i]\,|\,\tilde{y}}(x[i]\,|\,D) + p_{\tilde{y}}(R) \prod_{i \in \{\mathbf{1},\mathbf{3},\ldots,\mathbf{15}\}} p_{\tilde{x}[i]\,|\,\tilde{y}}(x[i]\,|\,R)}$$

Applying the model

i	1	2	3	4	5	6	7	8
$p_{\tilde{x}[i] \mid \tilde{y}}(1 \mid R)$	0.19	0.50	0.14	0.99	0.95	0.90	0.24	0.15
$p_{\tilde{x}[i] \mid \tilde{y}}(1 \mid D)$	0.61	0.50	0.89	0.05	0.22	0.47	0.78	0.83
Example	N	–	**Y**	**N**	**N**	Y	**Y**	**Y**

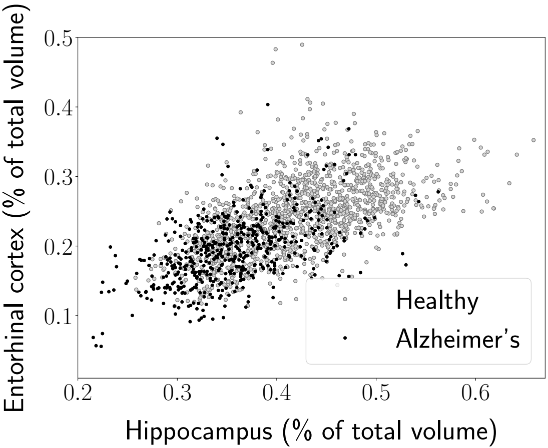
i	9	10	11	12	13	14	15	16
$p_{\tilde{x}[i] \mid \tilde{y}}(1 \mid R)$	0.11	0.55	0.14	0.87	0.86	0.98	0.09	0.66
$p_{\tilde{x}[i] \mid \tilde{y}}(1 \mid D)$	0.76	0.47	0.51	0.15	0.29	0.35	0.64	0.94
Example	N	Y	N	**N**	**N**	**N**	**Y**	–

$$p_{\tilde{y} \mid \tilde{x}}(D \mid x)$$

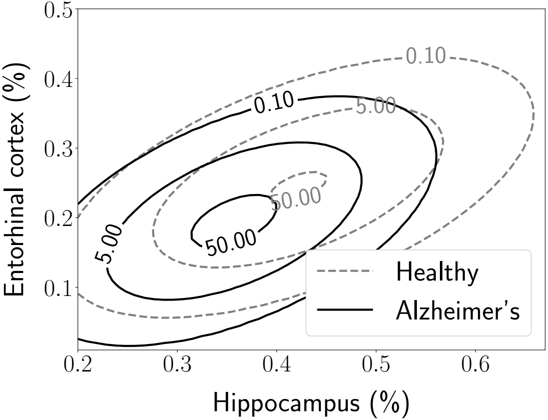
$$= \frac{p_{\tilde{y}}(D) \prod_{i \in \{1,3,\ldots,15\}} p_{\tilde{x}[i] \mid \tilde{y}}(x[i] \mid D)}{p_{\tilde{y}}(D) \prod_{i \in \{1,3,\ldots,15\}} p_{\tilde{x}[i] \mid \tilde{y}}(x[i] \mid D) + p_{\tilde{y}}(R) \prod_{i \in \{1,3,\ldots,15\}} p_{\tilde{x}[i] \mid \tilde{y}}(x[i] \mid R)}$$

$$= 1 - 1.4 10^{-8}$$

Gaussian discriminant analysis



Gaussian parametric model of features given class



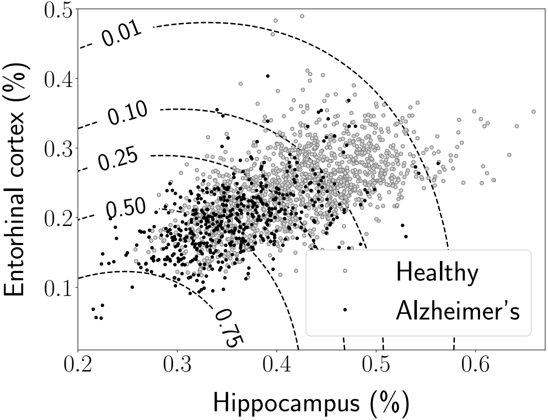
Bayes' rule

$$p_{\tilde{y}\,|\,\tilde{x}}(y\,|\,x) = \frac{p_{\tilde{y}}(y) f_{\tilde{x}\,|\,\tilde{y}}(x\,|\,y)}{\sum_{k\in\{1,2,\ldots,c\}} p_{\tilde{y}}(k) f_{\tilde{x}\,|\,\tilde{y}}(x\,|\,k)}$$

Bayes' rule

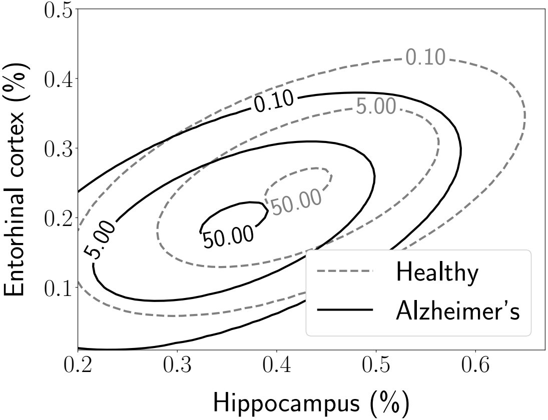
$$
\begin{aligned}
p_{\tilde{y}\,|\,\tilde{x}}(y\,|\,x) &= \frac{p_{\tilde{y}}(y) f_{\tilde{x}\,|\,\tilde{y}}(x\,|\,y)}{\sum_{k\in\{1,2,\ldots,c\}} p_{\tilde{y}}(k) f_{\tilde{x}\,|\,\tilde{y}}(x\,|\,k)} \\
&= \frac{\frac{p_{\tilde{y}}(y)}{\sqrt{(2\pi)^d \left|\Sigma_y\right|}} \exp\left(-\frac{1}{2}\left(x-\mu_y\right)^T \Sigma_y^{-1}\left(x-\mu_y\right)\right)}{\sum_{k\in\{1,2,\ldots,c\}} \frac{p_{\tilde{y}}(k)}{\sqrt{(2\pi)^d \left|\Sigma_k\right|}} \exp\left(-\frac{1}{2}\left(x-\mu_k\right)^T \Sigma_k^{-1}\left(x-\mu_k\right)\right)}
\end{aligned}
$$

$p_{\tilde{y} \mid \tilde{x}}$

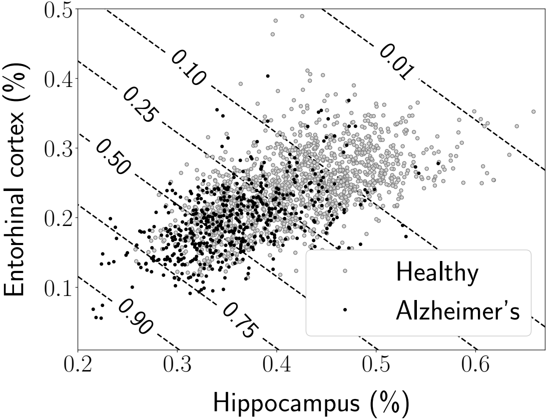


Linear discriminant analysis

We fit $\Sigma_a = \Sigma_b = \Sigma$



Linear discriminant analysis



Generative Models

Discriminative Models

Evaluation

Discriminative models

Goal: Directly approximate conditional pmf $p_{\tilde{y} \mid \tilde{x}}$ of class given features

Discriminative models

Goal: Directly approximate conditional pmf $p_{\tilde{y}\,|\,\tilde{x}}$ of class given features

Different assumptions lead to different models p_{Θ}

Discriminative models

Goal: Directly approximate conditional pmf $p_{\tilde{y} \mid \tilde{x}}$ of class given features

Different assumptions lead to different models p_{Θ}

- **Linear:** Logistic / softmax regression

Discriminative models

Goal: Directly approximate conditional pmf $p_{\tilde{y} \mid \tilde{x}}$ of class given features

Different assumptions lead to different models p_Θ

- **Linear:** Logistic / softmax regression
- **Nonlinear:** Neural networks / classification trees

Parameters Θ are learned from data

Parameters Θ are learned from data

Data: (x_1, y_1), (x_2, y_2), $\ldots$, (x_n, y_n)

Parameters Θ are learned from data

Data: (x_1, y_1), (x_2, y_2), $\ldots$, (x_n, y_n)

Model the ith feature and label as the random variables $\tilde{x}_i$ and $\tilde{y}_i$

Parameters Θ are learned from data

Data: (x_1, y_1), (x_2, y_2), $\ldots$, (x_n, y_n)

Model the ith feature and label as the random variables $\tilde{x}_i$ and $\tilde{y}_i$

Maximize the conditional likelihood of the labels given the features

Conditional likelihood and log-likelihood

$$\mathcal{L}_{XY}(\Theta) := \mathrm{P}\left(\tilde{y}_1 = y_1, ..., \tilde{y}_n = y_n \,|\, \tilde{x}_1 = x_1, ..., \tilde{x}_n = x_n\right)$$

Conditional likelihood and log-likelihood

Assumption 1:
Labels are conditionally independent given the features

$$\mathcal{L}_{XY}(\Theta) := \mathrm{P}\left(\tilde{y}_1 = y_1, ..., \tilde{y}_n = y_n \,|\, \tilde{x}_1 = x_1, ..., \tilde{x}_n = x_n\right)$$

Conditional likelihood and log-likelihood

Assumption 1:
Labels are conditionally independent given the features

Assumption 2:
$\tilde{y}_i$ is conditionally independent from $\{\tilde{x}_m\}_{m \neq i}$ given $\tilde{x}_i$

$$\mathcal{L}_{XY}(\Theta) := \mathrm{P}\left(\tilde{y}_1 = y_1, ..., \tilde{y}_n = y_n \,|\, \tilde{x}_1 = x_1, ..., \tilde{x}_n = x_n\right)$$

Conditional likelihood and log-likelihood

Assumption 1:
Labels are conditionally independent given the features

Assumption 2:
$\tilde{y}_i$ is conditionally independent from $\{\tilde{x}_m\}_{m \neq i}$ given $\tilde{x}_i$

$$\begin{aligned}\mathcal{L}_{XY}(\Theta) &:= \mathrm{P}\left(\tilde{y}_1 = y_1, \ldots, \tilde{y}_n = y_n \,|\, \tilde{x}_1 = x_1, \ldots, \tilde{x}_n = x_n\right) \\ &= \prod_{k=1}^{c} \prod_{\{i : y_i = k\}} p_\Theta\left(x_i\right)_k\end{aligned}$$

Conditional likelihood and log-likelihood

Assumption 1:
Labels are conditionally independent given the features

Assumption 2:
$\tilde{y}_i$ is conditionally independent from $\{\tilde{x}_m\}_{m \neq i}$ given $\tilde{x}_i$

$$
\begin{aligned}
\mathcal{L}_{XY}(\Theta) &:= \mathrm{P}\left(\tilde{y}_1 = y_1, \ldots, \tilde{y}_n = y_n \,\middle|\, \tilde{x}_1 = x_1, \ldots, \tilde{x}_n = x_n\right) \\
&= \prod_{k=1}^{c} \prod_{\{i : y_i = k\}} p_\Theta(x_i)_k \\
\log \mathcal{L}_{XY}(\Theta) &= \sum_{k=1}^{c} \sum_{\{i : y_i = k\}} \log p_\Theta(x_i)_k
\end{aligned}
$$

Conditional likelihood and log-likelihood

Assumption 1:
Labels are conditionally independent given the features

Assumption 2:
$\tilde{y}_i$ is conditionally independent from $\{\tilde{x}_m\}_{m \neq i}$ given $\tilde{x}_i$

$$
\begin{aligned}
\mathcal{L}_{XY}(\Theta) &:= \mathrm{P}\left(\tilde{y}_1 = y_1, \ldots, \tilde{y}_n = y_n \,|\, \tilde{x}_1 = x_1, \ldots, \tilde{x}_n = x_n\right) \\
&= \prod_{k=1}^{c} \prod_{\{i : y_i = k\}} p_\Theta\left(x_i\right)_k \\
\log \mathcal{L}_{XY}(\Theta) &= \sum_{k=1}^{c} \sum_{\{i : y_i = k\}} \log p_\Theta\left(x_i\right)_k
\end{aligned}
$$

Can be difficult to maximize!

Generalized linear models

Idea: Approximate $p_{\tilde{y} \mid \tilde{x}}$ as a linear function of the features

Generalized linear models

Idea: Approximate $p_{\tilde{y} \mid \tilde{x}}$ as a linear function of the features

Problem: Linear functions are *not valid probabilities*

Generalized linear models

Idea: Approximate $p_{\tilde{y} \mid \tilde{x}}$ as a linear function of the features

Problem: Linear functions are *not valid probabilities*

Solution: Map linear function to [0,1] using link function:

Generalized linear models

Idea: Approximate $p_{\tilde{y} \mid \tilde{x}}$ as a linear function of the features

Problem: Linear functions are *not valid probabilities*

Solution: Map linear function to [0,1] using link function:

- Logistic function for binary classification

Generalized linear models

Idea: Approximate $p_{\tilde{y} \mid \tilde{x}}$ as a linear function of the features

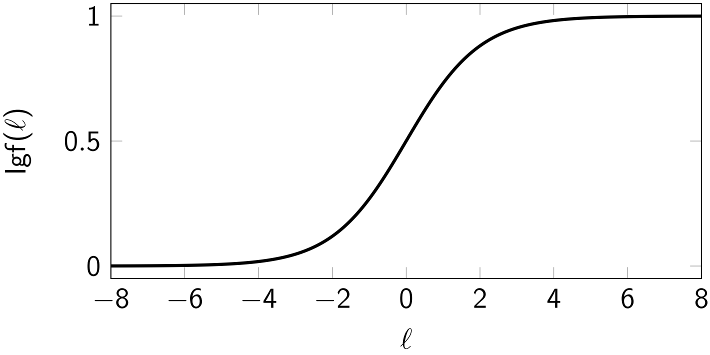
Problem: Linear functions are *not valid probabilities*

Solution: Map linear function to [0,1] using link function:

- Logistic function for binary classification
- Softmax function for multiclass classification

Logistic function

$$\mathsf{lgf}(\ell) := \frac{\exp(\ell)}{1+\exp(\ell)} = \frac{1}{1+\exp(-\ell)}$$



Logistic regression

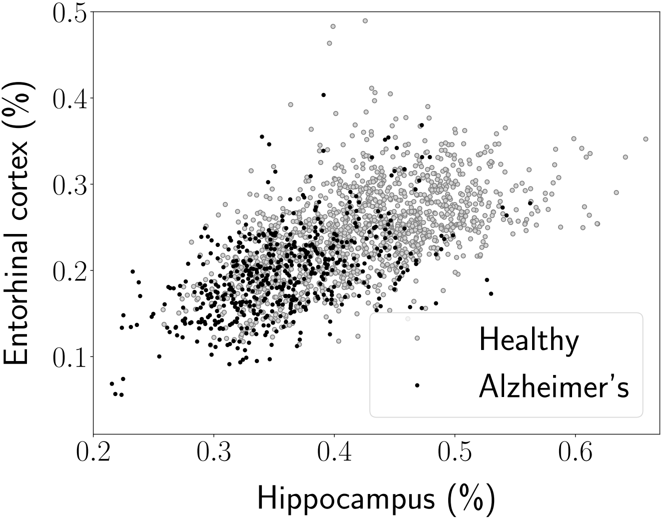
$$\mathrm{P}\left(\tilde{y} = 1 \,|\, \tilde{x} = x\right) \approx p_{\Theta}\left(x\right) := \mathrm{lgf}(\beta^T x + \alpha)$$

Logistic regression

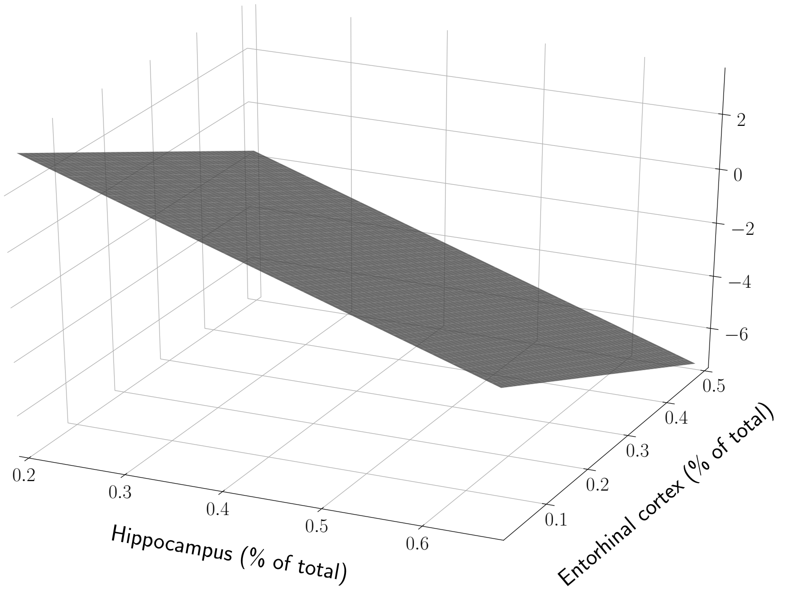
$$\mathrm{P}\left(\tilde{y}=1 \,|\, \tilde{x}=x\right) \approx p_{\Theta}\left(x\right) := \operatorname{lgf}(\beta^{T} x+\alpha)$$

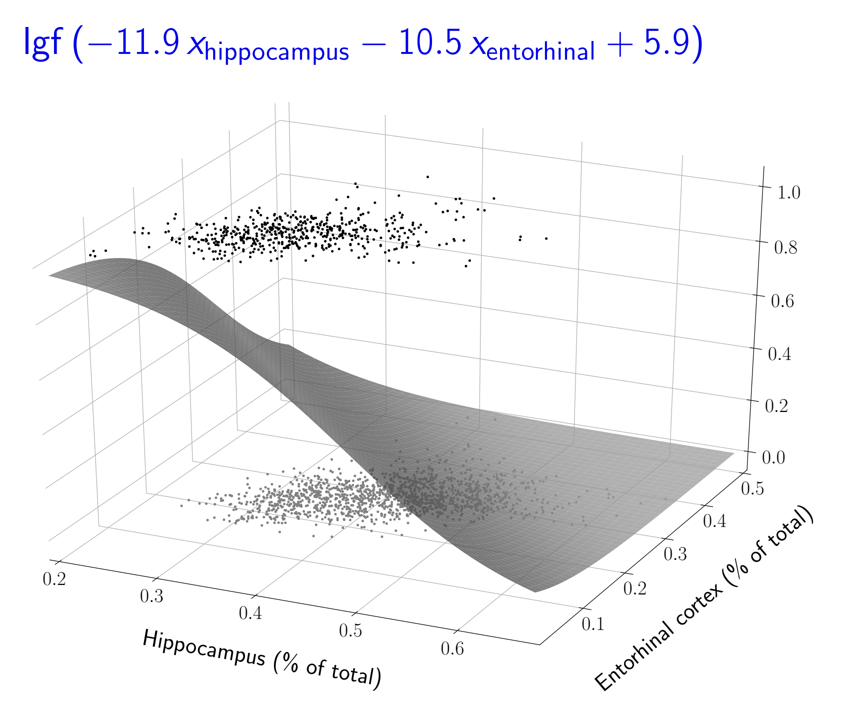
Concave log-likelihood maximized via iterative methods

Alzheimer's diagnosis



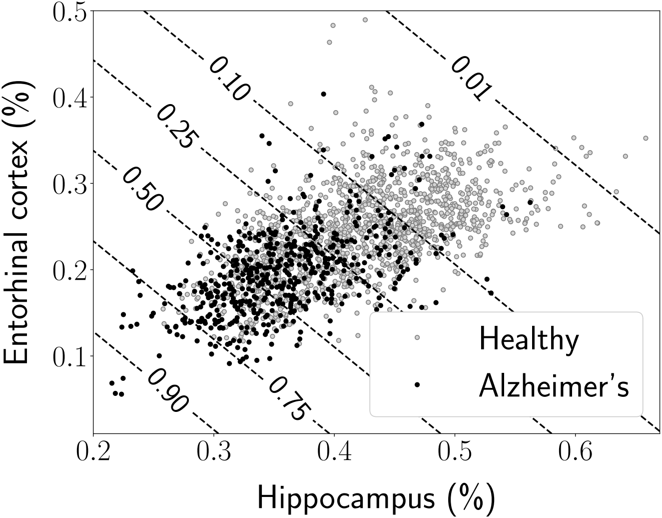
$$-11.9\,x_{\text{hippocampus}} - 10.5\,x_{\text{entorhinal}} + 5.9$$





$\mathrm{lgf}\,(-11.9\,x_{\text{hippocampus}} - 10.5\,x_{\text{entorhinal}} + 5.9)$
Hippocampus (% of total)
Entorhinal cortex (% of total)
0.2
0.3
0.4
0.5
0.6
0.1
0.2
0.3
0.4
0.5
0.0
0.2
0.4
0.6
0.8
1.0

$\operatorname{lgf}(-11.9\, x_{\text{hippocampus}} - 10.5\, x_{\text{entorhinal}} + 5.9)$



Softmax

$$\mathrm{P}\left(\tilde{y} = k \,|\, \tilde{x} = x\right) \approx p_{\Theta}\left(x\right)$$

Softmax

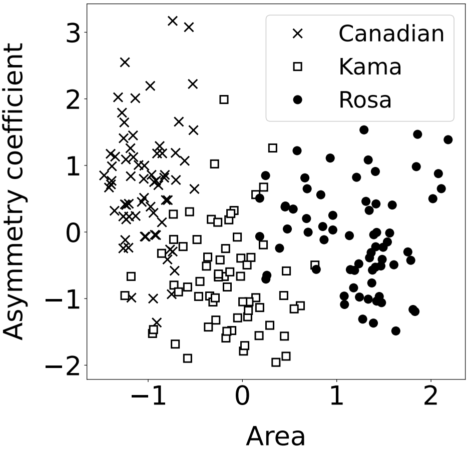
$$\mathrm{P}\left(\tilde{y} = k \mid \tilde{x} = x\right) \approx p_{\Theta}\left(x\right) := \frac{\exp\left({\beta_k}^T x + \alpha_k\right)}{\sum_{l=1}^{c} \exp\left({\beta_l}^T x + \alpha_l\right)} \quad 1 \leq k \leq c$$

Softmax

$$\mathrm{P}\left(\tilde{y} = k \,|\, \tilde{x} = x\right) \approx p_{\Theta}\left(x\right) := \frac{\exp\left(\beta_k{}^T x + \alpha_k\right)}{\sum_{l=1}^{c} \exp\left(\beta_l{}^T x + \alpha_l\right)} \quad 1 \leq k \leq c$$

Concave log-likelihood maximized via iterative methods

Wheat varieties



Softmax acts like a soft maximum

$x_{\text{area}} := -1$, $x_{\text{asym}} := 2$

$$\begin{bmatrix} \text{Canadian} \\ \text{Kama} \\ \text{Rosa} \end{bmatrix} = \begin{bmatrix} -\,7.7\,x_{\text{area}} + 0.9\,x_{\text{asym}} - 2.9 \\ 0.4\,x_{\text{area}} - 1.2\,x_{\text{asym}} + 2.7 \\ 7.3\,x_{\text{area}} + 0.4\,x_{\text{asym}} + 0.2 \end{bmatrix} = \begin{bmatrix} 6.6 \\ -\,0.1 \\ -\,6.3 \end{bmatrix}$$

Softmax acts like a soft maximum

$x_{\text{area}} := -1,\ x_{\text{asym}} := 2$

$$\begin{bmatrix} \text{Canadian} \\ \text{Kama} \\ \text{Rosa} \end{bmatrix} = \begin{bmatrix} -7.7\,x_{\text{area}} + 0.9\,x_{\text{asym}} - 2.9 \\ 0.4\,x_{\text{area}} - 1.2\,x_{\text{asym}} + 2.7 \\ 7.3\,x_{\text{area}} + 0.4\,x_{\text{asym}} + 0.2 \end{bmatrix} = \begin{bmatrix} 6.6 \\ -0.1 \\ -6.3 \end{bmatrix}$$

$$\exp(6.6) = 735 \quad \exp(-0.1) = 0.905 \quad \exp(-6.3) = 0.002$$

Softmax acts like a soft maximum

$x_{\text{area}} := -1,\ x_{\text{asym}} := 2$

$$\begin{bmatrix} \text{Canadian} \\ \text{Kama} \\ \text{Rosa} \end{bmatrix} = \begin{bmatrix} -7.7\,x_{\text{area}} + 0.9\,x_{\text{asym}} - 2.9 \\ 0.4\,x_{\text{area}} - 1.2\,x_{\text{asym}} + 2.7 \\ 7.3\,x_{\text{area}} + 0.4\,x_{\text{asym}} + 0.2 \end{bmatrix} = \begin{bmatrix} 6.6 \\ -0.1 \\ -6.3 \end{bmatrix}$$

$$\exp(6.6) = 735 \quad \exp(-0.1) = 0.905 \quad \exp(-6.3) = 0.002$$

Softmax:

$$\begin{bmatrix} \mathrm{P}\left(\text{Canadian} \mid x_{\text{area}}, x_{\text{asym}}\right) \\ \mathrm{P}\left(\text{Kama} \mid x_{\text{area}}, x_{\text{asym}}\right) \\ \mathrm{P}\left(\text{Rosa} \mid x_{\text{area}}, x_{\text{asym}}\right) \end{bmatrix} = \begin{bmatrix} \frac{735}{735+0.905+0.002} \\ \frac{0.905}{735+0.905+0.002} \\ \frac{0.002}{735+0.905+0.002} \end{bmatrix} = \begin{bmatrix} \\ \\ \\ \end{bmatrix}$$

Softmax acts like a soft maximum

$x_{\text{area}} := -1,\ x_{\text{asym}} := 2$

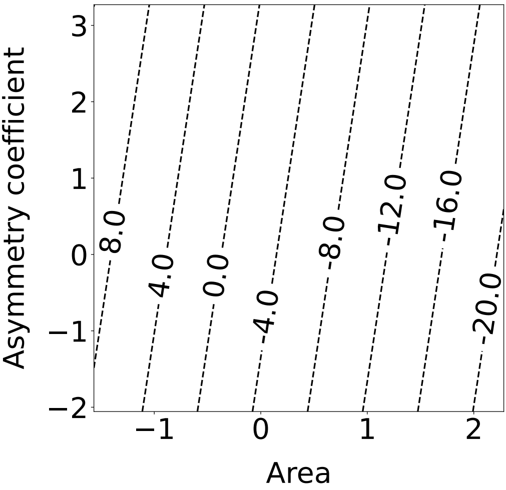
$$\begin{bmatrix} \text{Canadian} \\ \text{Kama} \\ \text{Rosa} \end{bmatrix} = \begin{bmatrix} -7.7\,x_{\text{area}} + 0.9\,x_{\text{asym}} - 2.9 \\ 0.4\,x_{\text{area}} - 1.2\,x_{\text{asym}} + 2.7 \\ 7.3\,x_{\text{area}} + 0.4\,x_{\text{asym}} + 0.2 \end{bmatrix} = \begin{bmatrix} 6.6 \\ -0.1 \\ -6.3 \end{bmatrix}$$

$$\exp(6.6) = 735 \quad \exp(-0.1) = 0.905 \quad \exp(-6.3) = 0.002$$

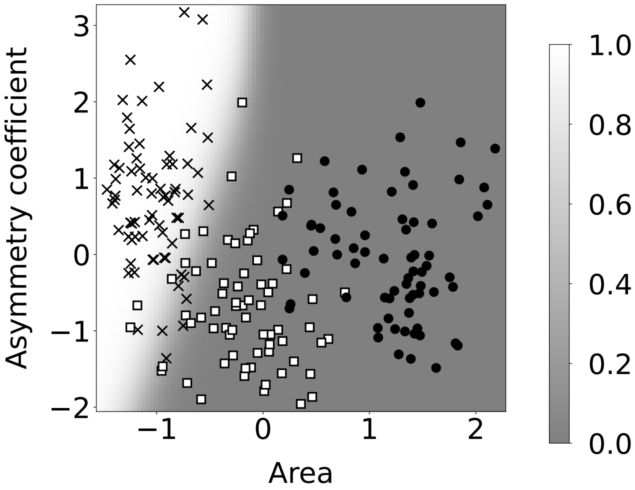
Softmax:

$$\begin{bmatrix} \mathrm{P}(\text{Canadian} \mid x_{\text{area}}, x_{\text{asym}}) \\ \mathrm{P}(\text{Kama} \mid x_{\text{area}}, x_{\text{asym}}) \\ \mathrm{P}(\text{Rosa} \mid x_{\text{area}}, x_{\text{asym}}) \end{bmatrix} = \begin{bmatrix} \frac{735}{735+0.905+0.002} \\ \frac{0.905}{735+0.905+0.002} \\ \frac{0.002}{735+0.905+0.002} \end{bmatrix} = \begin{bmatrix} 0.999 \\ 0.001 \\ 0.000 \end{bmatrix}$$

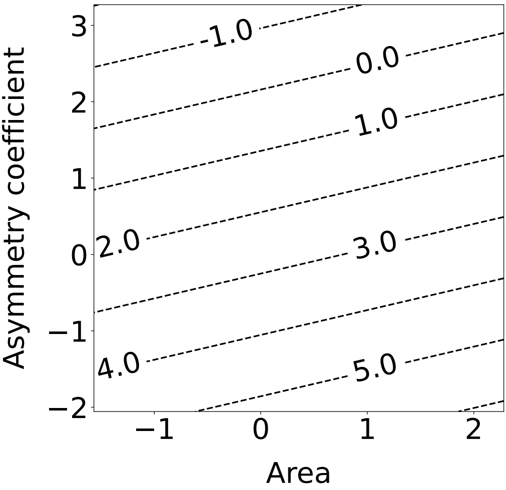
Canadian: $-7.7\, x_{\text{area}} + 0.9\, x_{\text{asym}} - 2.9$



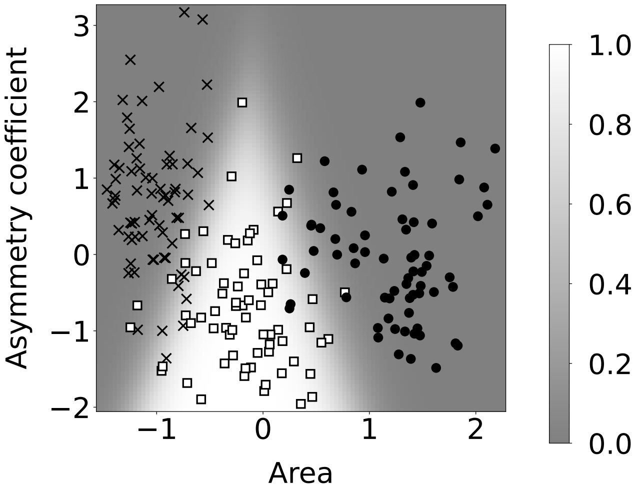
Canadian: Estimated probability



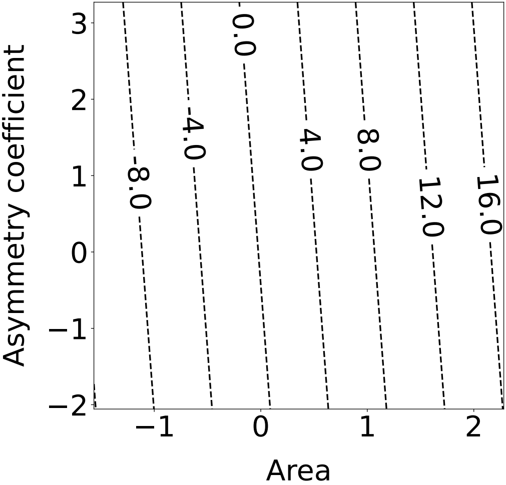
Kama: $0.4\,x_{\text{area}} - 1.2\,x_{\text{asym}} + 2.7$



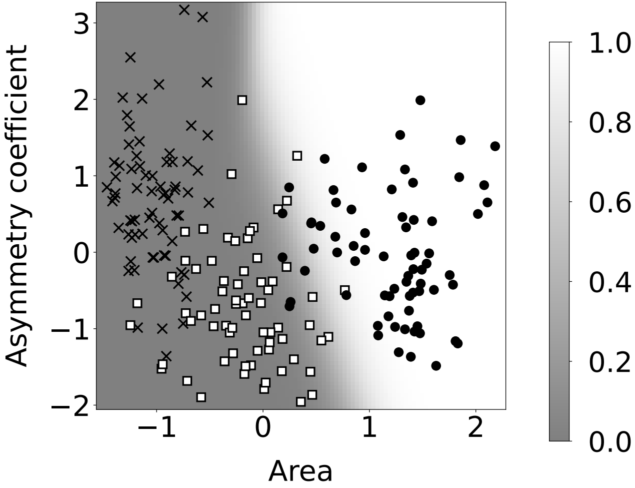
Kama: Estimated probability



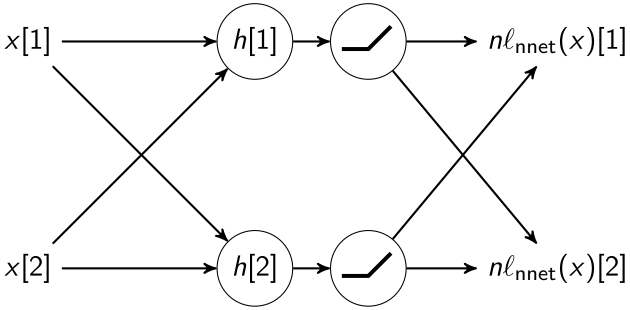
Rosa: $7.3\,x_{\text{area}} + 0.4\,x_{\text{asym}} + 0.2$



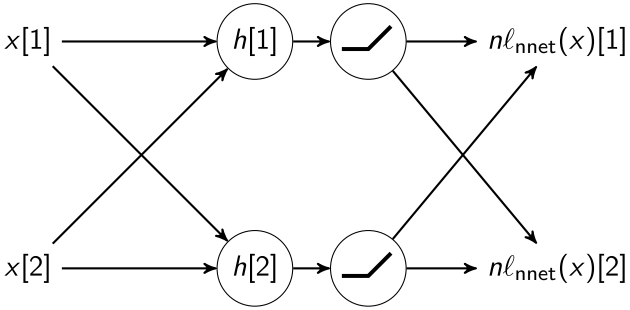
Rosa: Estimated probability



Neural network

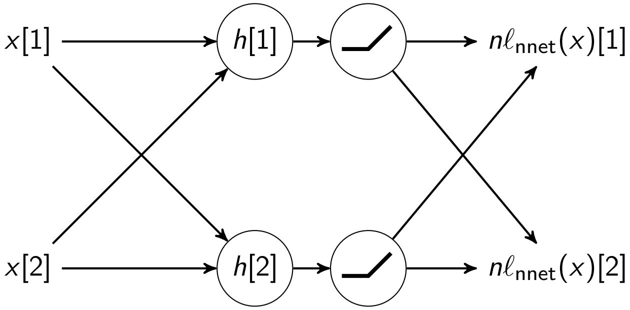


Neural network



$$\mathrm{P}\left(\tilde{y} = k \mid \tilde{x} = x\right) \approx p_{\Theta}\left(x\right)_k := \frac{\exp\left(n\ell_{\text{nnet}}\left(x\right)\left[k\right]\right)}{\sum_{l=1}^{c} \exp\left(n\ell_{\text{nnet}}\left(x\right)\left[l\right]\right)}$$

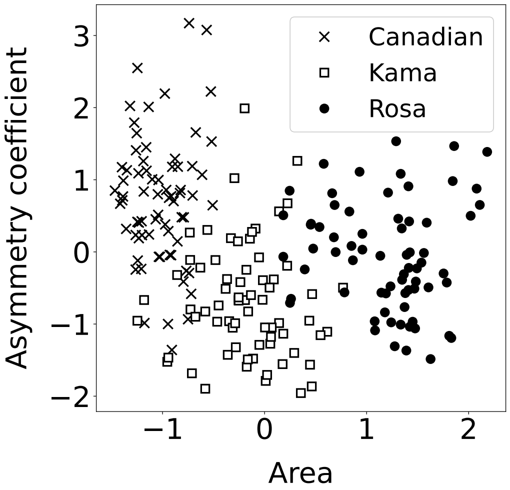
Neural network



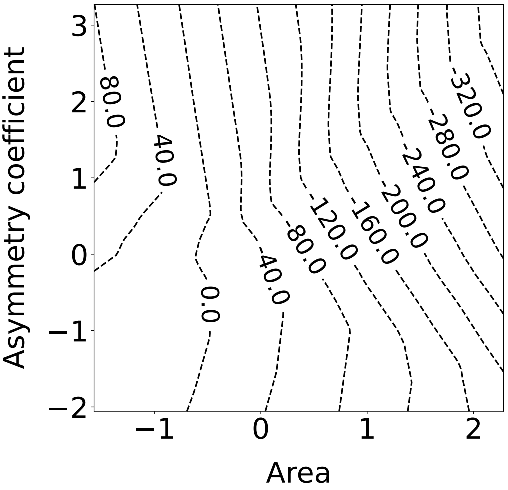
$$\mathrm{P}\left(\tilde{y} = k \mid \tilde{x} = x\right) \approx p_{\Theta}\left(x\right)_k := \frac{\exp\left(n\ell_{\text{nnet}}\left(x\right)\left[k\right]\right)}{\sum_{l=1}^{c} \exp\left(n\ell_{\text{nnet}}\left(x\right)\left[l\right]\right)}$$

Non-concave log likelihood maximized via stochastic gradient descent on batches

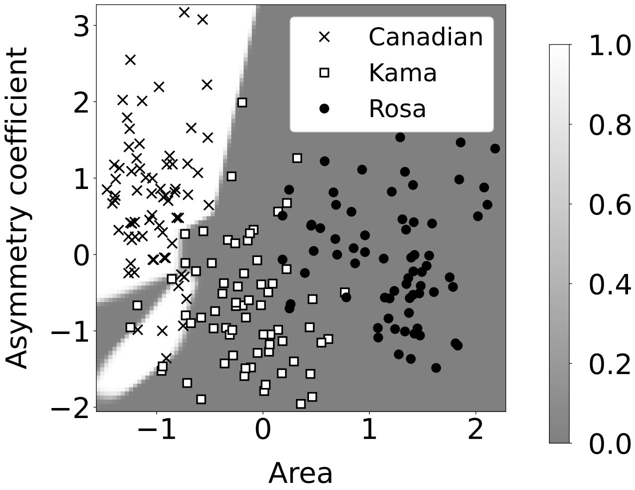
2-layer neural network with 100 hidden variables



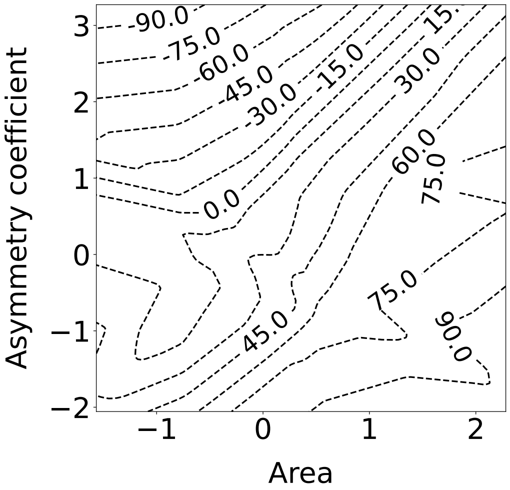
Nonlinear output: Canadian



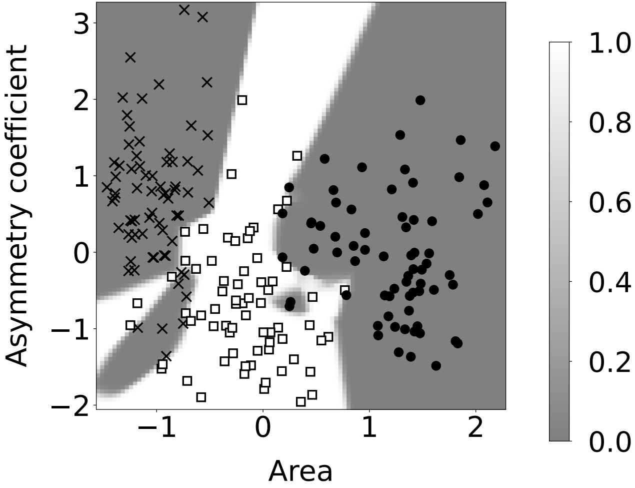
Estimated probability: Canadian



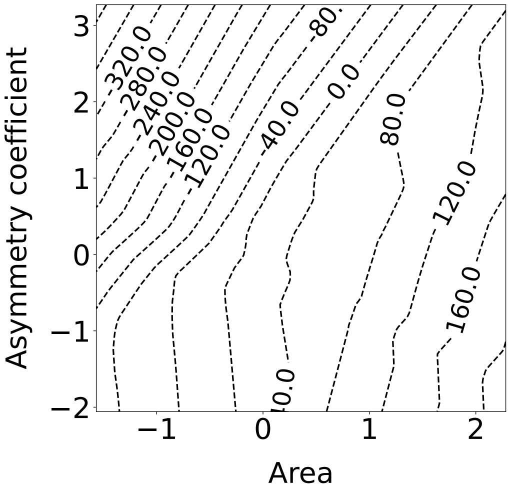
Nonlinear output: Kama



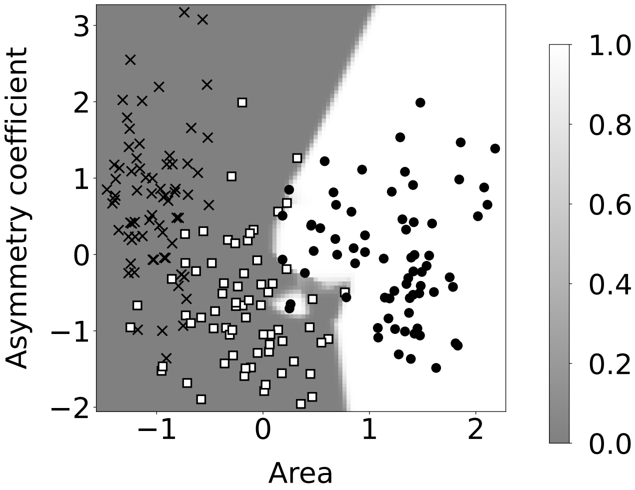
Estimated probability: Kama



Nonlinear output: Rosa



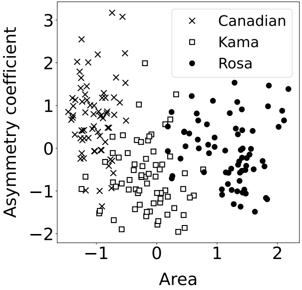
Estimated probability: Rosa

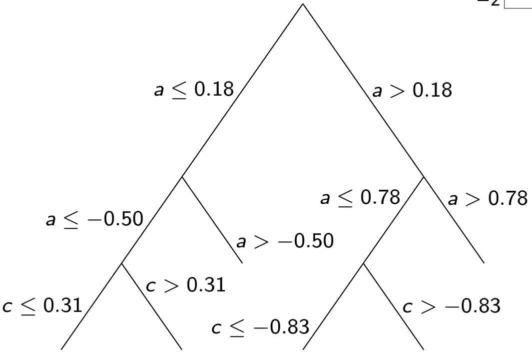


Classification tree

Area (a)

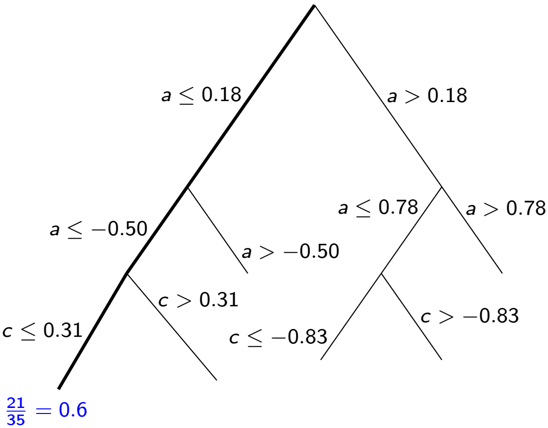
Asymmetric coefficient (c)





$\mathrm{P}\left(\tilde{y} = \text{Canadian} \,|\, \tilde{a} = -1, \tilde{c} = -1\right)$

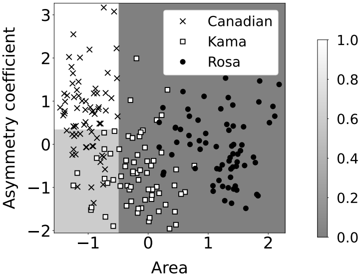
Probability estimates
assigned to leaves

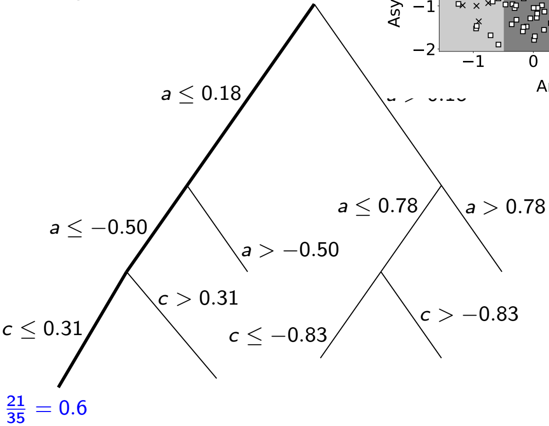


$\mathrm{P}\left(\tilde{y} = \text{Canadian} \mid \tilde{a} = -1, \tilde{c} = -1\right)$

Probability estimates assigned to leaves

Fraction within region in training set

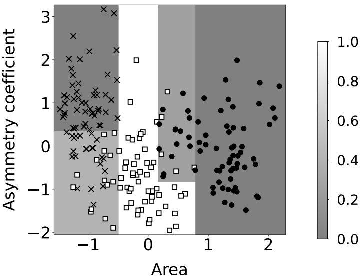


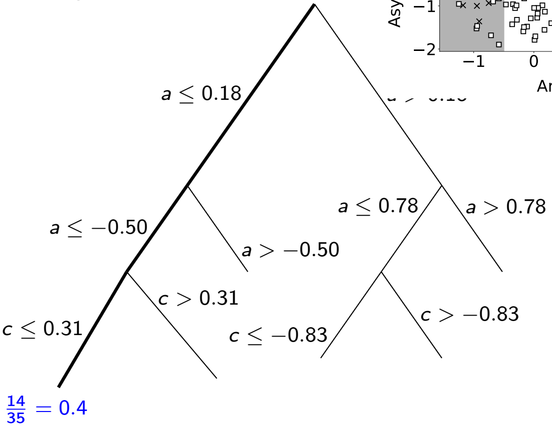


$\mathrm{P}\left(\tilde{y} = \text{Kama} \,\middle|\, \tilde{a} = -1, \tilde{c} = -1\right)$

Probability estimates assigned to leaves

Fraction within region in training set

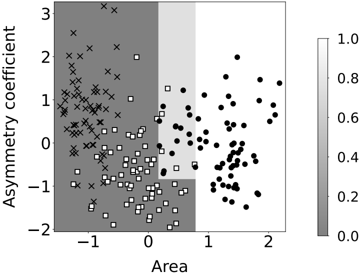


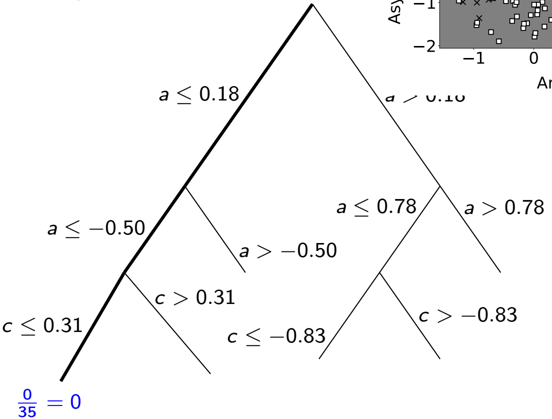


$\mathrm{P}\left(\tilde{y} = \text{Rosa} \,|\, \tilde{a} = -1, \tilde{c} = -1\right)$

Probability estimates assigned to leaves

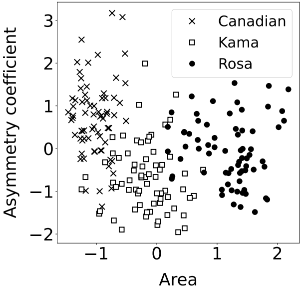
Fraction within region in training set

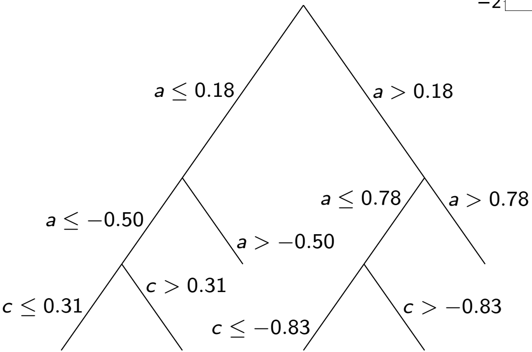




Classification tree

Tree built iteratively, adding bifurcation that most increases the log likelihood





Ensembles

Problem: Simple trees underfit / Complex trees overfit

Ensembles

Problem: Simple trees underfit / Complex trees overfit

Solution: Combine multiple simple trees

Ensembles

Problem: Simple trees underfit / Complex trees overfit

Solution: Combine multiple simple trees

Three main strategies to obtain individual trees:

Ensembles

Problem: Simple trees underfit / Complex trees overfit

Solution: Combine multiple simple trees

Three main strategies to obtain individual trees:

1. Bagging: Train on resampled datasets generated via *bootstrapping*

Ensembles

Problem: Simple trees underfit / Complex trees overfit

Solution: Combine multiple simple trees

Three main strategies to obtain individual trees:

1. Bagging: Train on resampled datasets generated via *bootstrapping*

2. Random forests: Train *randomized* trees on resampled datasets generated via bootstrapping

Ensembles

Problem: Simple trees underfit / Complex trees overfit

Solution: Combine multiple simple trees

Three main strategies to obtain individual trees:

1. Bagging: Train on resampled datasets generated via *bootstrapping*

2. Random forests: Train *randomized* trees on resampled datasets generated via bootstrapping

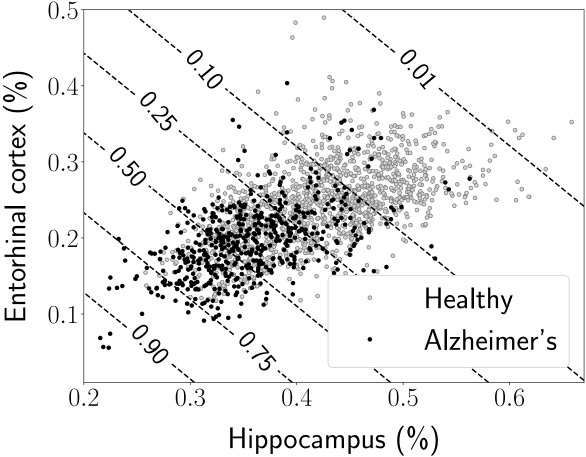
3. Boosting: Train *complementary* trees that fit residuals of previous trees (scaled down to avoid overfitting)

Generative Models

Discriminative Models

Evaluation

Probability estimates



How should we evaluate them?

Evaluation

We focus on binary classification (2 classes)

Evaluation

We focus on binary classification (2 classes)

- Threshold probabilities and evaluate binary classification estimates

Evaluation

We focus on binary classification (2 classes)

- Threshold probabilities and evaluate binary classification estimates
- Assess discrimination ability of the probabilities

Evaluation

We focus on binary classification (2 classes)

- Threshold probabilities and evaluate binary classification estimates
- Assess discrimination ability of the probabilities
- Assess calibration of probabilities

Metrics for binary classification estimates

Metrics for binary classification estimates

Accuracy: Fraction of total examples that are correctly classified

Metrics for binary classification estimates

Accuracy: Fraction of total examples that are correctly classified

True positive rate (TPR): Fraction of positive examples that are correctly classified

Metrics for binary classification estimates

Accuracy: Fraction of total examples that are correctly classified

True positive rate (TPR): Fraction of positive examples that are correctly classified

False positive rate (FPR): Fraction of negative examples that are incorrectly classified

Metrics for binary classification estimates

Accuracy: Fraction of total examples that are correctly classified

True positive rate (TPR): Fraction of positive examples that are correctly classified

False positive rate (FPR): Fraction of negative examples that are incorrectly classified

Precision: Fraction of examples predicted as positive that are correctly classified

Metrics for binary classification estimates

Accuracy: Fraction of total examples that are correctly classified

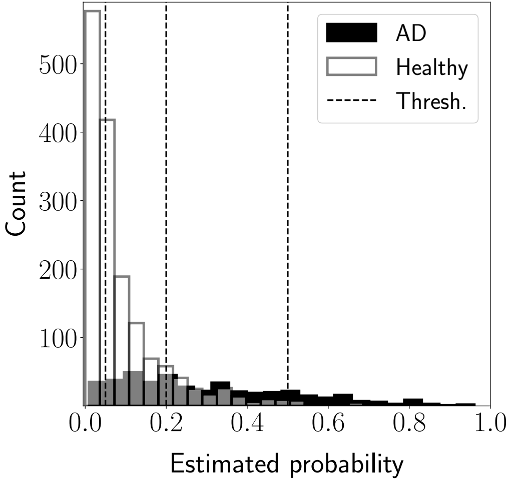
True positive rate (TPR): Fraction of positive examples that are correctly classified

False positive rate (FPR): Fraction of negative examples that are incorrectly classified

Precision: Fraction of examples predicted as positive that are correctly classified

F1 score: Harmonic mean of TPR and precision

Alzheimer's diagnostics



Metrics

Threshold = 0.05 Threshold = 0.2 Threshold = 0.5

Metrics

Threshold $= 0.05$	Threshold $= 0.2$	Threshold $= 0.5$
Accuracy $= 0.58$	Accuracy $= 0.82$	Accuracy $= 0.81$

Metrics

Threshold $= 0.05$	Threshold $= 0.2$	Threshold $= 0.5$
Accuracy $= 0.58$	Accuracy $= 0.82$	Accuracy $= 0.81$
TPR $= 0.93$	TPR $= 0.61$	TPR $= 0.19$

Metrics

Threshold = 0.05	Threshold = 0.2	Threshold = 0.5
Accuracy = 0.58	Accuracy = 0.82	Accuracy = 0.81
TPR = 0.93	TPR = 0.61	TPR = 0.19
FPR = 0.52	FPR = 0.13	FPR = 0.01

Metrics

Threshold = 0.05	Threshold = 0.2	Threshold = 0.5
Accuracy = 0.58	Accuracy = 0.82	Accuracy = 0.81
TPR = 0.93	TPR = 0.61	TPR = 0.19
FPR = 0.52	FPR = 0.13	FPR = 0.01
Precision = 0.33	Precision = 0.57	Precision = 0.78

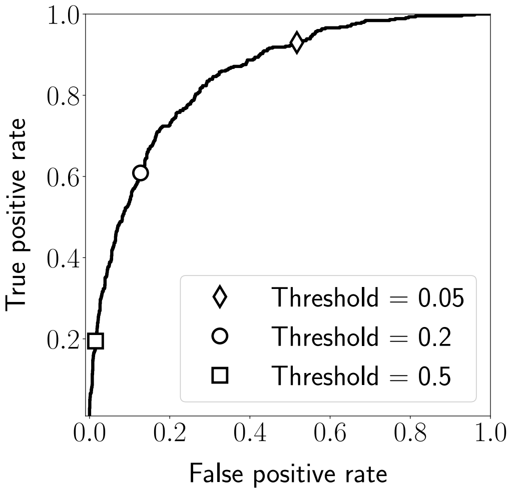
Metrics

Threshold = 0.05	Threshold = 0.2	Threshold = 0.5
Accuracy = 0.58	Accuracy = 0.82	Accuracy = 0.81
TPR = 0.93	TPR = 0.61	TPR = 0.19
FPR = 0.52	FPR = 0.13	FPR = 0.01
Precision = 0.33	Precision = 0.57	Precision = 0.78
F1 score = 0.49	F1 score = 0.59	F1 score = 0.31

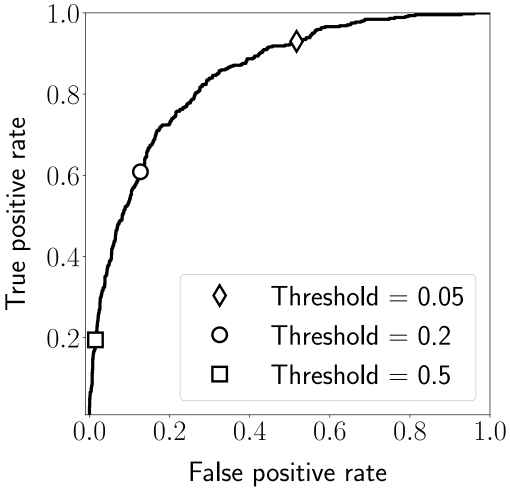
TPR - FPR tradeoff

Threshold = 0.05	Threshold = 0.2	Threshold = 0.5
Accuracy = 0.58	Accuracy = 0.82	Accuracy = 0.81
TPR= 0.93	TPR= 0.61	TPR= 0.19
FPR= 0.52	FPR= 0.13	FPR= 0.01
Precision = 0.33	Precision = 0.57	Precision = 0.78
F1 score = 0.49	F1 score = 0.59	F1 score = 0.31

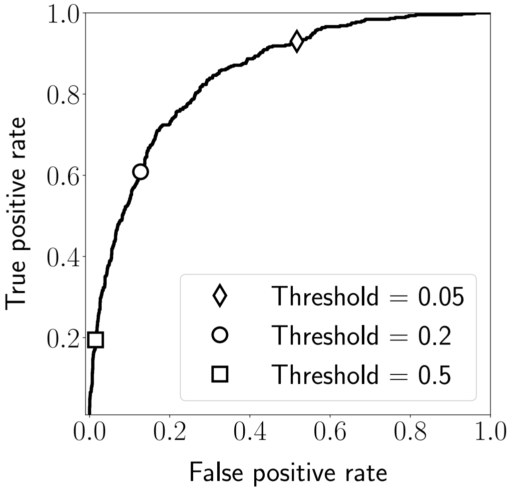
Receiver operating characteristic (ROC) curve



Area under ROC curve (AUROC or AUC) = 0.847

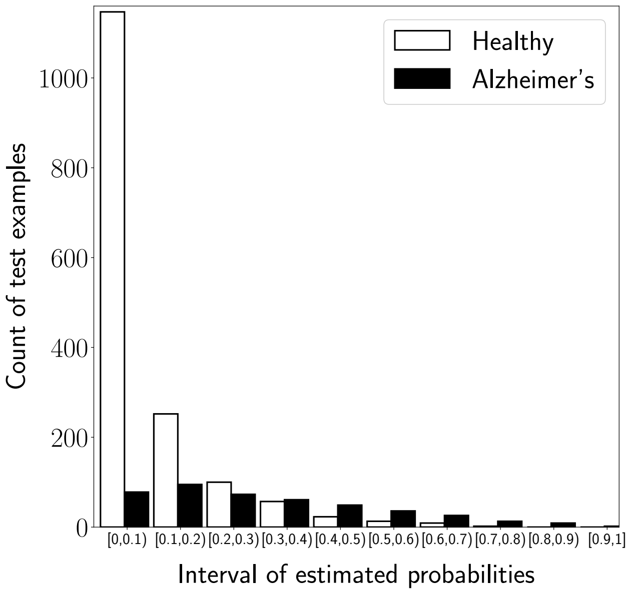


Area under ROC curve (AUROC or AUC) = 0.847

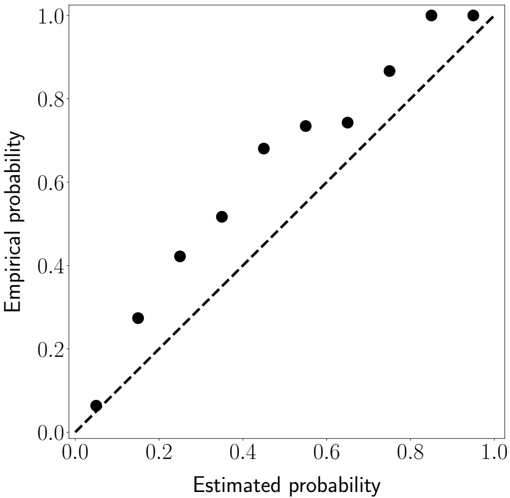


Fraction of negative - positive examples such that estimated probability is higher for positive example

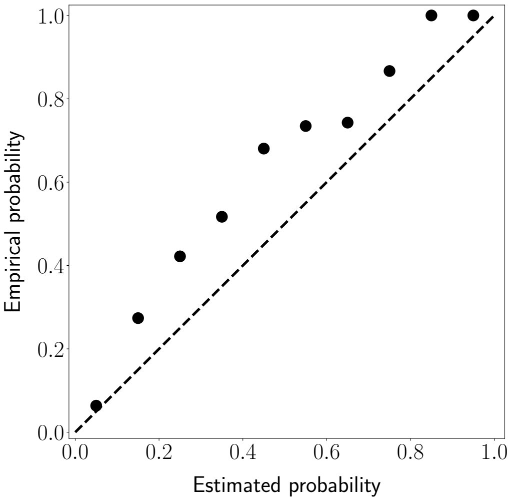
Estimated probabilities vs empirical probabilities



Calibration



Calibration



Brier score evaluates both discrimination and calibration

What have we learned?

- Generative models:

What have we learned?

- Generative models:
  - Naive Bayes

What have we learned?

- Generative models:
  - Naive Bayes
  - Gaussian discriminant analysis

What have we learned?

- Generative models:
  - Naive Bayes
  - Gaussian discriminant analysis
- Discriminative models:

What have we learned?

- Generative models:
  - Naive Bayes
  - Gaussian discriminant analysis
- Discriminative models:
  - Linear: Logistic regression / Softmax regression

What have we learned?

- Generative models:
  - Naive Bayes
  - Gaussian discriminant analysis
- Discriminative models:
  - Linear: Logistic regression / Softmax regression
  - Nonlinear: Neural networks / Classification trees

What have we learned?

- Generative models:
  - Naive Bayes
  - Gaussian discriminant analysis
- Discriminative models:
  - Linear: Logistic regression / Softmax regression
  - Nonlinear: Neural networks / Classification trees
- Evaluation