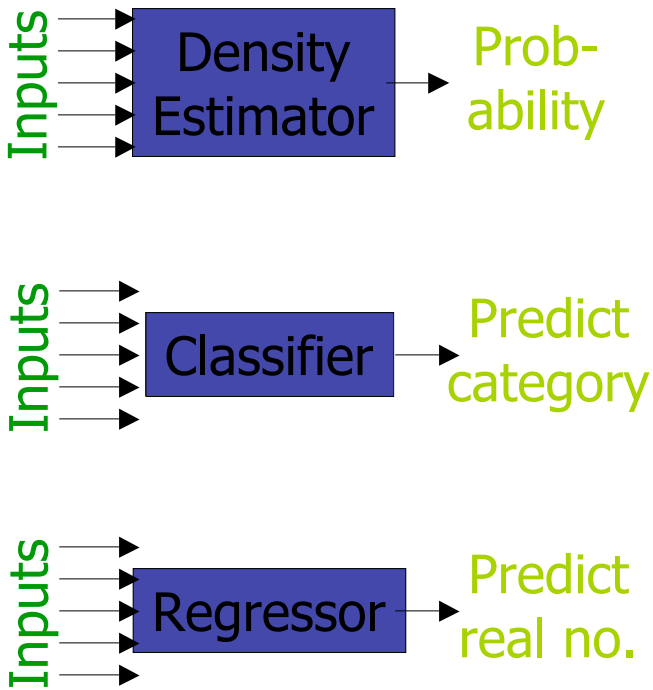
15-381: Artificial Intelligence

Naïve Bayesian classifier

Where we are



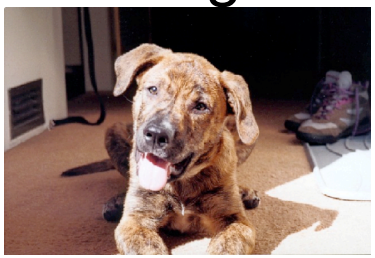
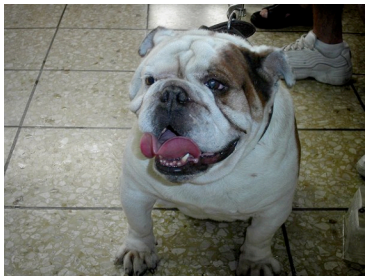
√

Today

Later

Classification

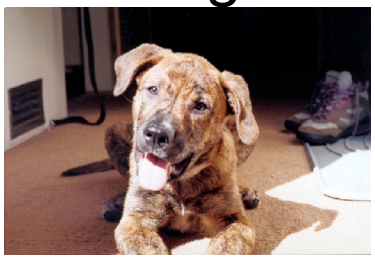
- Assume we want to teach a computer to distinguish between cats and dogs …




Classification

- Assume we want to teach a computer to distinguish between cats and dogs …

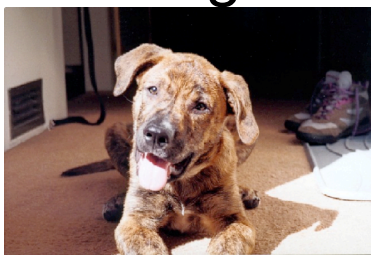



Several steps:

1. feature transformation
2. Model / classifier specification
3. Model / classifier estimation (with regularization)
4. feature selection

Classification

- Assume we want to teach a computer to distinguish between cats and dogs …

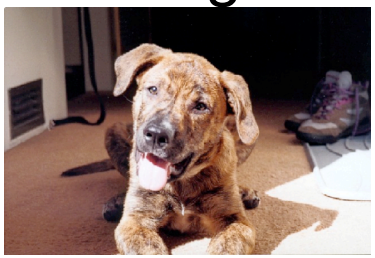
Several steps:

1. feature transformation
2. Model / classifier specification
3. Model / classifier estimation (with regularization)
4. feature selection

How do we encode the picture? A collection of pixels? Do we use the entire image or a subset? …

Classification

- Assume we want to teach a computer to distinguish between cats and dogs …

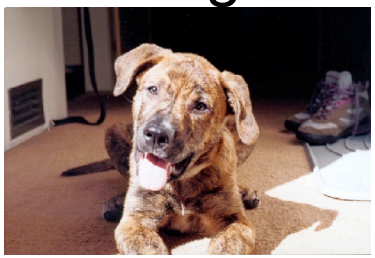
Several steps:

1. feature transformation
2. Model / classifier specification
3. Model / classifier estimation (with regularization)
4. feature selection

What type of classifier should we use?

Classification

- Assume we want to teach a computer to distinguish between cats and dogs …

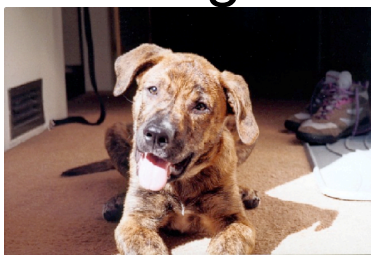
Several steps:

1. feature transformation
2. Model / classifier specification
3. Model / classifier estimation (with regularization)
4. feature selection

How do we learn the parameters of our classifier? Do we have enough examples to learn a good model?

Classification

- Assume we want to teach a computer to distinguish between cats and dogs …

Several steps:

1. feature transformation
2. Model / classifier specification
3. Model / classifier estimation (with regularization)
4. feature selection

Do we really need all the features? Can we use a smaller number and still achieve the same (or better) results?

Types of classifiers

- We can divide the large variety of classification approaches into roughly two main types

  1. Generative:
     - build a generative statistical model
     - e.g., Bayesian networks

  2. Discriminative
     - directly estimate a decision rule/boundary
     - e.g., decision tree

Types of classifiers

- We can divide the large variety of classification approaches into roughly two main types

  1. Generative:
     - build a generative statistical model
     - e.g., Bayesian networks

  2. Discriminative
     - directly estimate a decision rule/boundary
     - e.g., decision trees

Text classification

- Text classification (information retrieval):
  - A few labeled documents $D_l = \{(x_1,y_1),...,(x_n,y_n)\}$
  - Many unlabeled data points: $D_t = \{(x_{n+1}),...,(x_{n+m})\}$

- Problem formulations:
  – train with D_l
  – classify all the unlabeled examples in D_t

  (there are also other ways to formulate this problem)

- Several steps:
  1. feature transformation
  2. model/classifier specification
  3. model/classifier estimation (with regularization)
  4. feature selection

Feature transformation

- We can construct *m* (about 10,000) indicator features (basis functions) $\{\phi_i(x)\}$ for whether a word appears in the document
  - $\phi_i(x) = 1$, if word *i* appears in document *x*; zero otherwise
- $\Phi(x) = [\phi_1(x), \ldots, \phi_m(x)]^T$ is the resulting feature vector
- For notational simplicity we will replace each document *x* with a fixed length vector $\Phi = [\phi_1, \ldots, \phi_m]^T$, where $\phi_i = \phi_i(x)$.

Example

Assume we would like to classify documents as election related or not.

Dictionary

- Washington
- Congress

…

54. Mccain

55. Obama

56. Nader

$$\phi_{54}=\phi_{54}(x) = 1$$
$$\phi_{55}=\phi_{55}(x) = 1$$
$$\phi_{56}=\phi_{56}(x) = 0$$

Selecting a classification model: Bayes Classifier

- We can determine which class a document belongs to by finding the class that maximizes the likelihood of the label (class) given the observed features:

$$\hat{y} = \arg\max_v p(y = v \mid \Phi)$$

Our feature vector

- This can be simplified by using Bayes rule:

$$\begin{aligned}\hat{y} &= \arg\max_v p(y = v \mid \Phi) \\ &= \arg\max_v \frac{p(\Phi \mid y = v)\, p(y = v)}{p(\Phi)} \\ &= \arg\max_v p(\Phi \mid y = v)\, p(y = v)\end{aligned}$$

Easier to compute (part of our generative model)

Naïve Bayes Classifier

• In order to find the class of the document we would like to find *v* such that:

$$\begin{aligned}\hat{y} &= \arg\max_v \, p(y = v \mid \Phi) \\ &= \arg\max_v \frac{p(\Phi \mid y = v)\, p(y = v)}{p(\Phi)} \\ &= \arg\max_v \, p(\Phi \mid y = v)\, p(y = v)\end{aligned}$$

• If we assume that within each class of documents, the presence/absence of each word is independent of other words then we can use a "Naïve Bayes" model over documents and labels:

$$p(\Phi \mid y)\, p(y) = \left[\prod_i p(\phi_i \mid y, i) \right] p(y)$$

Binary value (whether or not word *i* appeared)

Specific model for feature *i*

Example: Classifying Election (E) or Sports (S)

Assume we have

$\phi_{bush} = 1$ P(1 | E, bush) = 0.8, P(1 | S, bush) = 0.1

$\phi_{obama} = 1$ P(1 | E, obama) = 0.9, P(1 | S, obama) = 0.05

$\phi_{mccain} = 1$ P(1 | E, mccain) = 0.9, P(1 | S, bush) = 0.05

$\phi_{football} = 0$ P(1 | E, football) = 0.1, P(1 | S, football) = 0.7

P(Y = E | 1,1,1,0) = 0.8*0.9*0.9*0.9P(E)

P(Y = S | 1,1,1,0) = 0.1*0.05*0.05*0.3P(S)

Assuming equal priors (P(E)=P(S)=0.5)) we get:

P(Y = E | 1,1,1,0) = 0.5832

P(Y = S | 1,1,1,0)* = 0.000075

Determining the parameters

- If we have a two class problem then $y \in \{0,1\}$.
- Thus, we can write the individual conditional probabilities compactly as

$$p(\phi_i \mid y, i) = \prod_i \theta_{i|y}^{\phi_i} (1 - \theta_{i|y})^{1-\phi_i}$$

  where $\theta_{i|y}$ is the conditional probability that ϕ_i = 1given the class y
- We can find the parameters that maximize the likelihood of our data in a similar way to what we did in the density estimation case

Parameter estimation

- The log likelihood can be written as:

$$l(\Phi, y) = \sum_y N_i(1, y)\log(\theta_{i|y}) + N_i(0, y)\log(1 - \theta_{i|y})$$

Log likelihood of training data (feature *i* + label)

Note: we do this independently for each feature *i*

$N_i(1,y)$ = # of documents containing word *i* and labeled y

$N_i(0,y)$ = # of documents without word *i* and labeled y

- Taking the derivative w.r.t. $\theta_{i|y}$ gives:

$$\frac{\partial l(\Phi, y)}{\partial \theta_{i|y}} = \frac{N_i(1, y)}{\theta_{i|y}} - \frac{N_i(0, y)}{1 - \theta_{i|y}}$$

$$\Rightarrow \hat{\theta}_{i|y} = \frac{N_i(1, y)}{N_i(1, y) + N_i(0, y)}$$

Specific models for Naïve Bayes classifiers: Binomial and Gaussian

- So far we assumed a binomial distribution

  - the features are generated by drawing $y_i \sim Binomial(p_1)$
- Another very useful assumption is the Gaussian generative model
- In this model we assume that the *i*'th record in the database is created using the following algorithm

  - Generate the features from a Gaussian PDF that depends on the value of y_i :

$$\boldsymbol{x}_i \sim N(\boldsymbol{m}_i, \boldsymbol{S}_i).$$

MLE for Gaussian Naïve Bayes Classifier

- We need to estimate one global value and two values for each feature in each class:
- Estimate the prior probability for each class (global parameter)
- For each feature
  - Estimate its mean for each of the two classes
  - Estimate its variance for each of the two classes

Gaussian Bayes Classification

$$P(y = v \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid y = v)P(y = v)}{p(\mathbf{x})}$$

Prior for each class

$$P(y = v \mid \mathbf{x}) = \frac{p_v \prod_i \frac{1}{(2\pi)^{1/2}\sigma_{i,v}} \exp\left[-\frac{\left(\mathbf{x}_i - \mu_{i,v}\right)^2}{2\sigma_{i,v}^2}\right]}{p(\mathbf{x})}$$

How do we deal with that?

We compute the probability for both classes and select the class with the higher probability as our predicted class

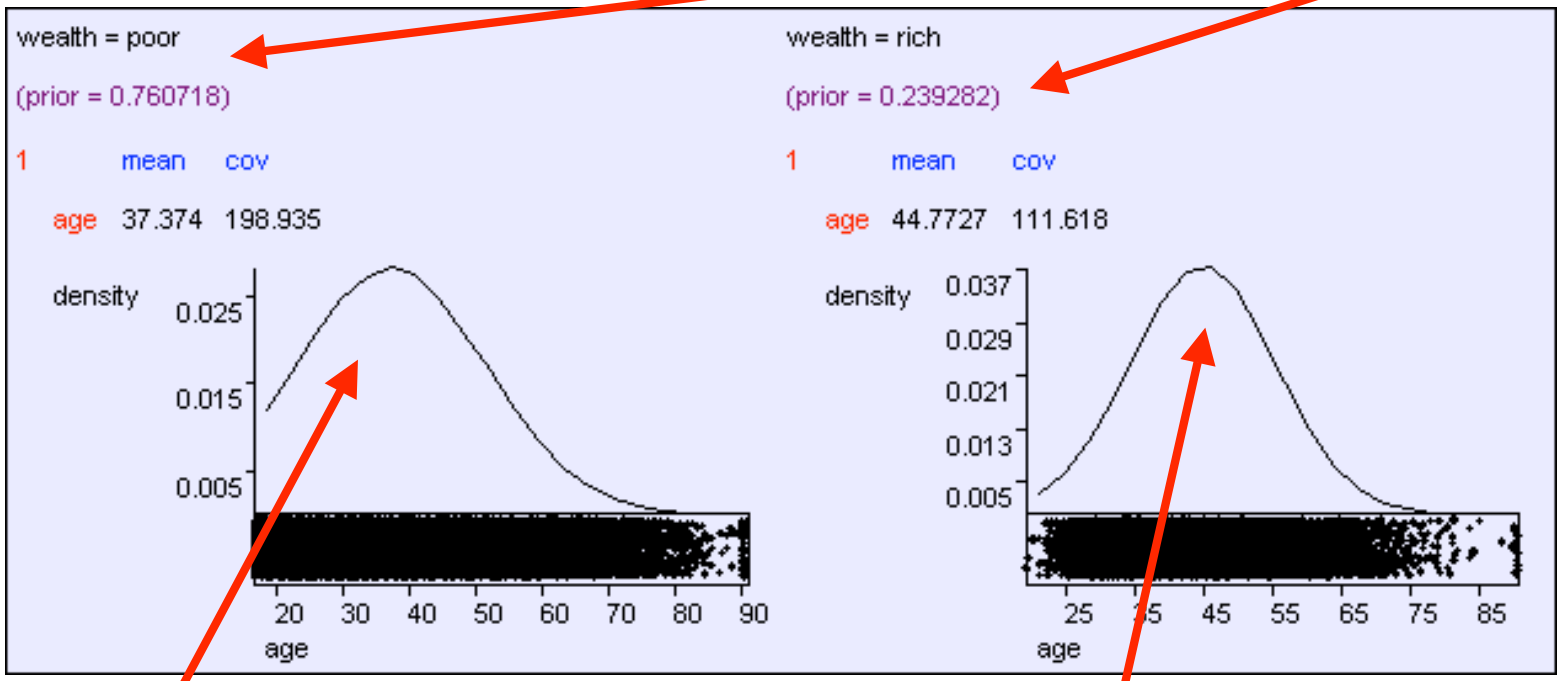
Here is a dataset

age	employment	education	edunu	marital	…	job	relation	race	gender	hours	country	wealth
					…							
39	State_gov	Bachelors	13	Never_marri	…	Adm_clerica	Not_in_famil	White	Male	40	United_Stat	poor
51	Self_emp_nc	Bachelors	13	Married	…	Exec_mana	Husband	White	Male	13	United_Stat	poor
39	Private	HS_grad	9	Divorced	…	Handlers_cle	Not_in_famil	White	Male	40	United_Stat	poor
54	Private	11th	7	Married	…	Handlers_cle	Husband	Black	Male	40	United_Stat	poor
28	Private	Bachelors	13	Married	…	Prof_specialt	Wife	Black	Female	40	Cuba	poor
38	Private	Masters	14	Married	…	Exec_mana	Wife	White	Female	40	United_Stat	poor
50	Private	9th	5	Married_spo	…	Other_servic	Not_in_famil	Black	Female	16	Jamaica	poor
52	Self_emp_nc	HS_grad	9	Married	…	Exec_mana	Husband	White	Male	45	United_Stat	rich
31	Private	Masters	14	Never_marri	…	Prof_specialt	Not_in_famil	White	Female	50	United_Stat	rich
42	Private	Bachelors	13	Married	…	Exec_mana	Husband	White	Male	40	United_Stat	rich
37	Private	Some_colleg	10	Married	…	Exec_mana	Husband	Black	Male	80	United_Stat	rich
30	State_gov	Bachelors	13	Married	…	Prof_specialt	Husband	Asian	Male	40	India	rich
24	Private	Bachelors	13	Never_marri	…	Adm_clerica	Own_child	White	Female	30	United_Stat	poor
33	Private	Assoc_acdm	12	Never_marri	…	Sales	Not_in_famil	Black	Male	50	United_Stat	poor
41	Private	Assoc_voc	11	Married	…	Craft_repair	Husband	Asian	Male	40	*MissingValu	rich
34	Private	7th_8th	4	Married	…	Transport_m	Husband	Amer_Indian	Male	45	Mexico	poor
26	Self_emp_nc	HS_grad	9	Never_marri	…	Farming_fish	Own_child	White	Male	35	United_Stat	poor
33	Private	HS_grad	9	Never_marri	…	Machine_op	Unmarried	White	Male	40	United_Stat	poor
38	Private	11th	7	Married	…	Sales	Husband	White	Male	50	United_Stat	poor
44	Self_emp_nc	Masters	14	Divorced	…	Exec_mana	Unmarried	White	Female	45	United_Stat	rich
41	Private	Doctorate	16	Married	…	Prof_specialt	Husband	White	Male	60	United_Stat	rich
:	:	:	:	:	:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:	:	:	:	:	:

48,000 records, 16 attributes [Kohavi 1995]

Predicting wealth from age

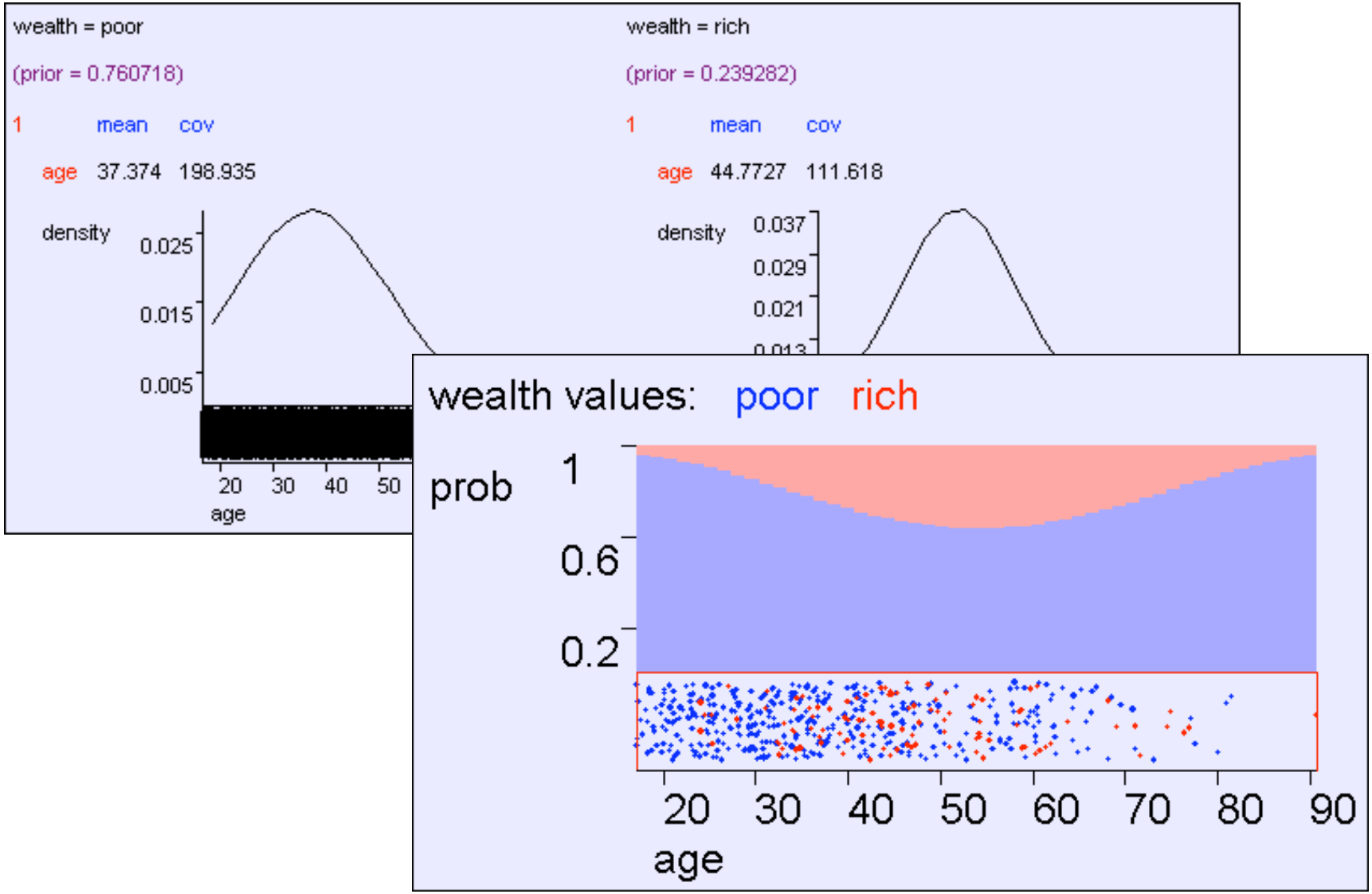
priors



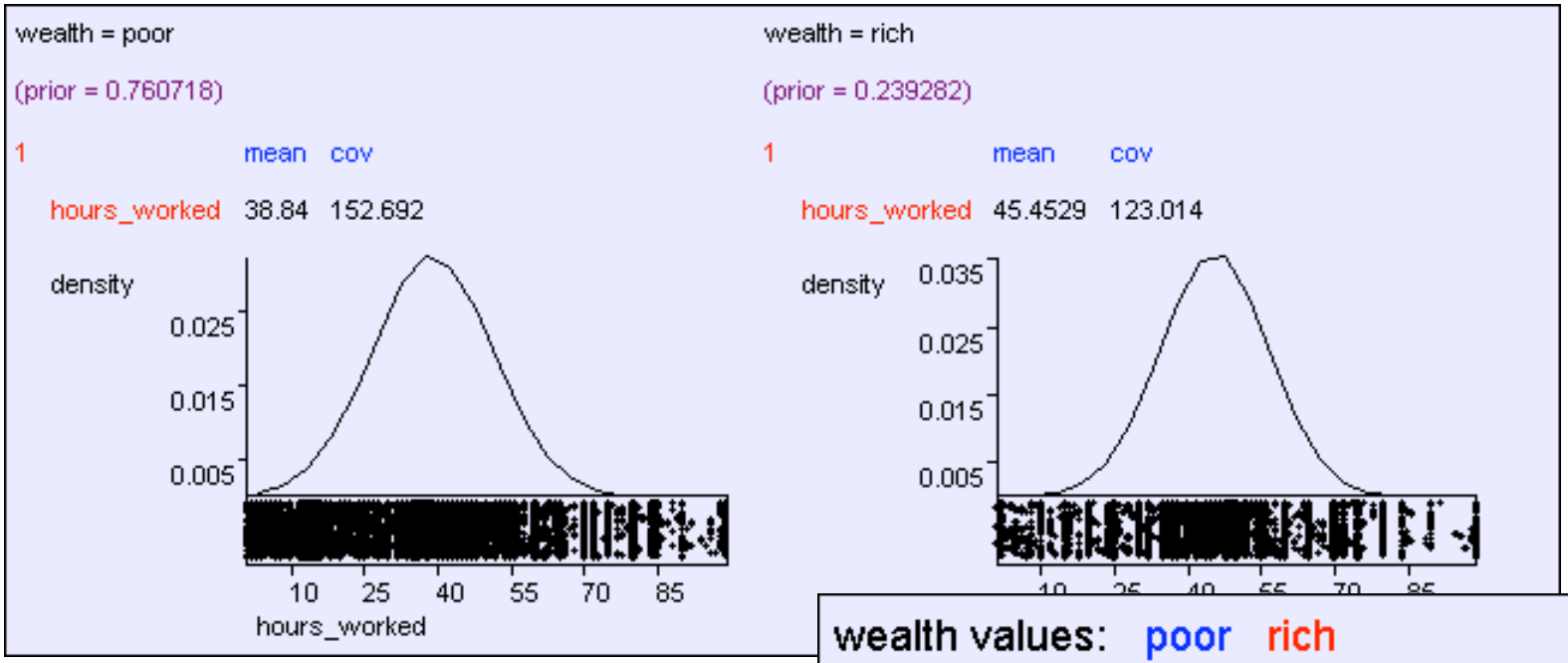
Distribution for the first class (poor)

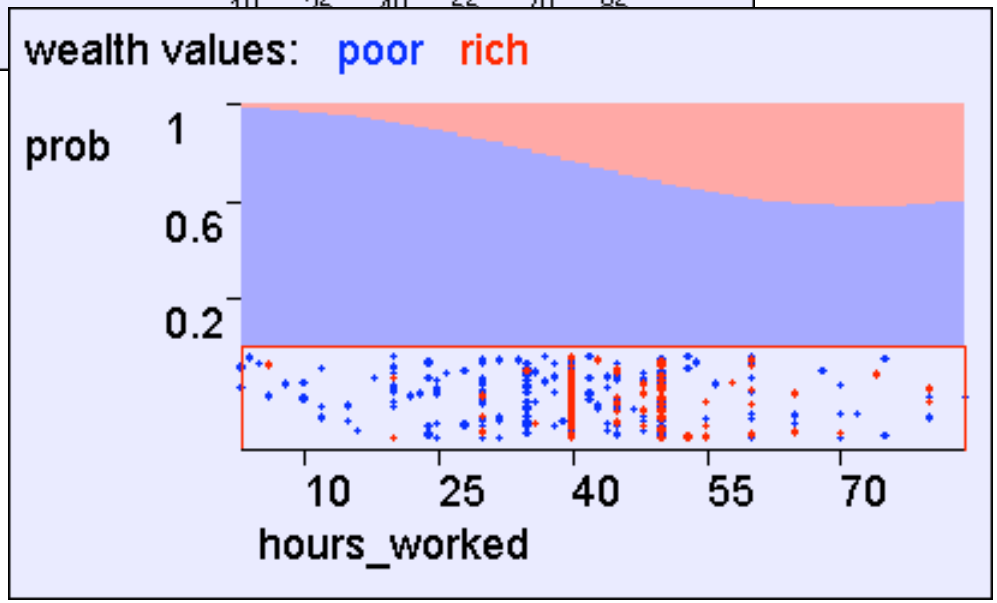
Distribution for the second class (rich)

Predicting wealth from age

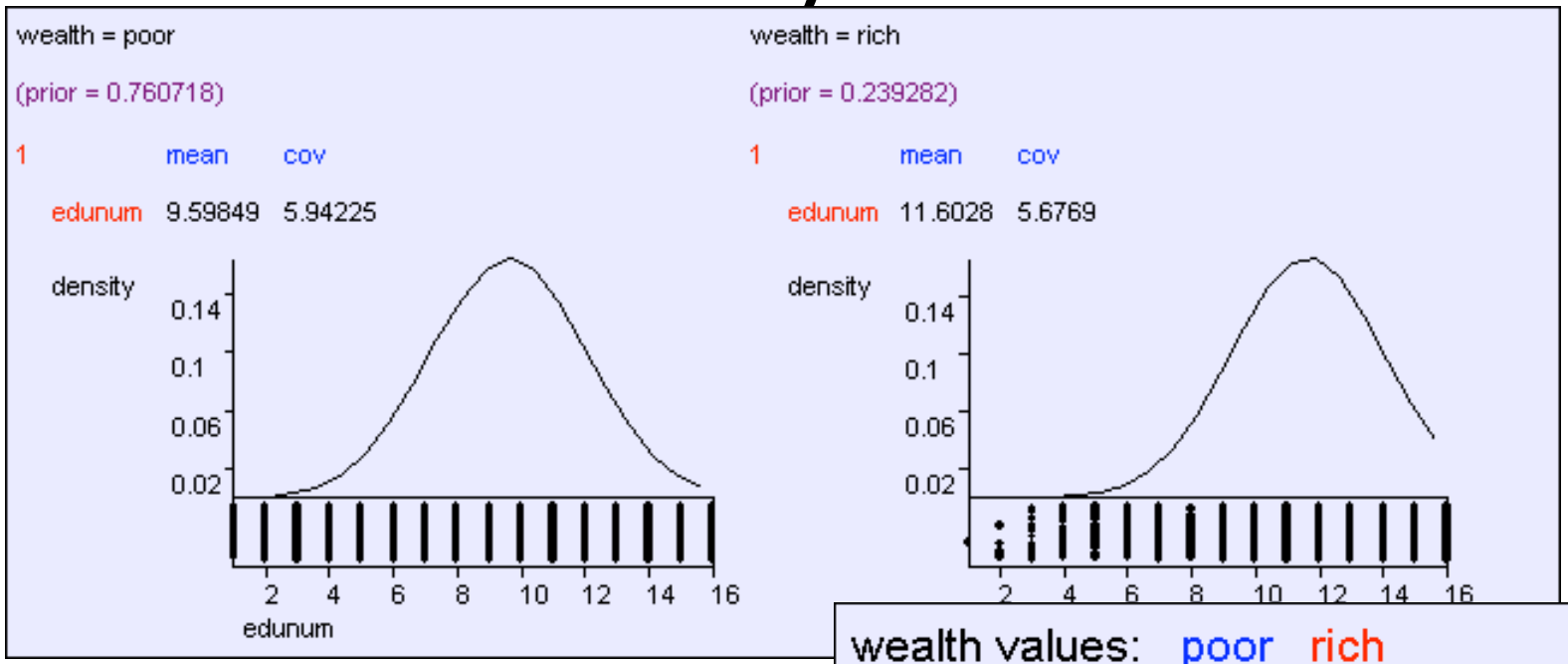


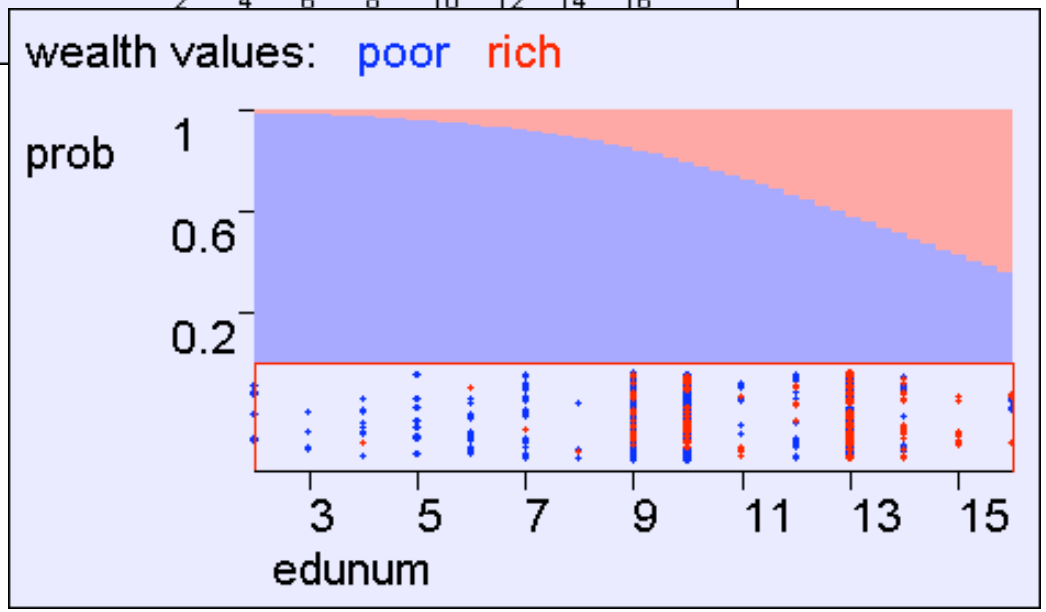
Wealth from hours worked





Wealth from years of education





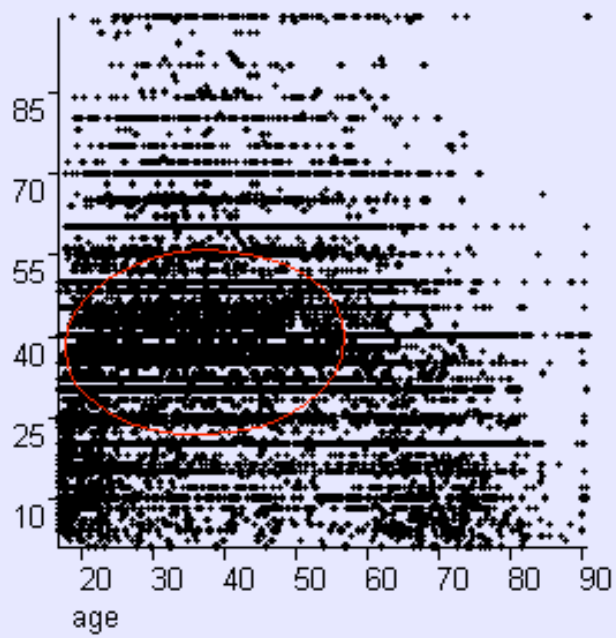
age, hours → wealth

wealth = poor

(prior = 0.760718)

1	mean	cov	
age	37.374	198.935	8.70283
hours_worked	38.84	8.70283	152.692

hours_worked

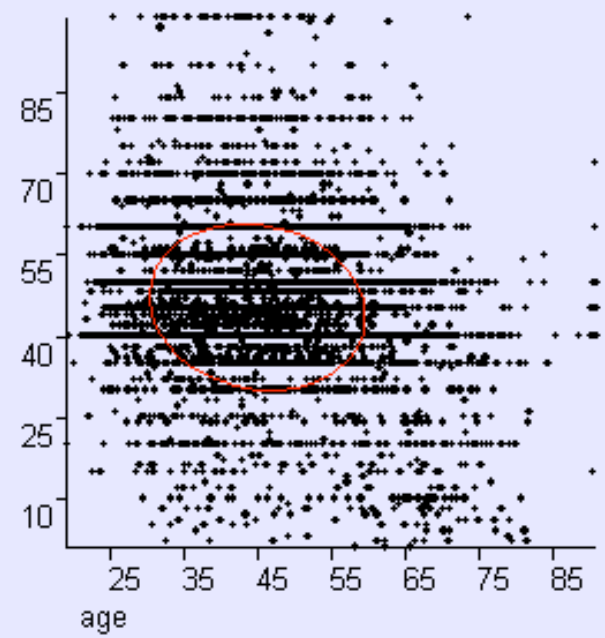


wealth = rich

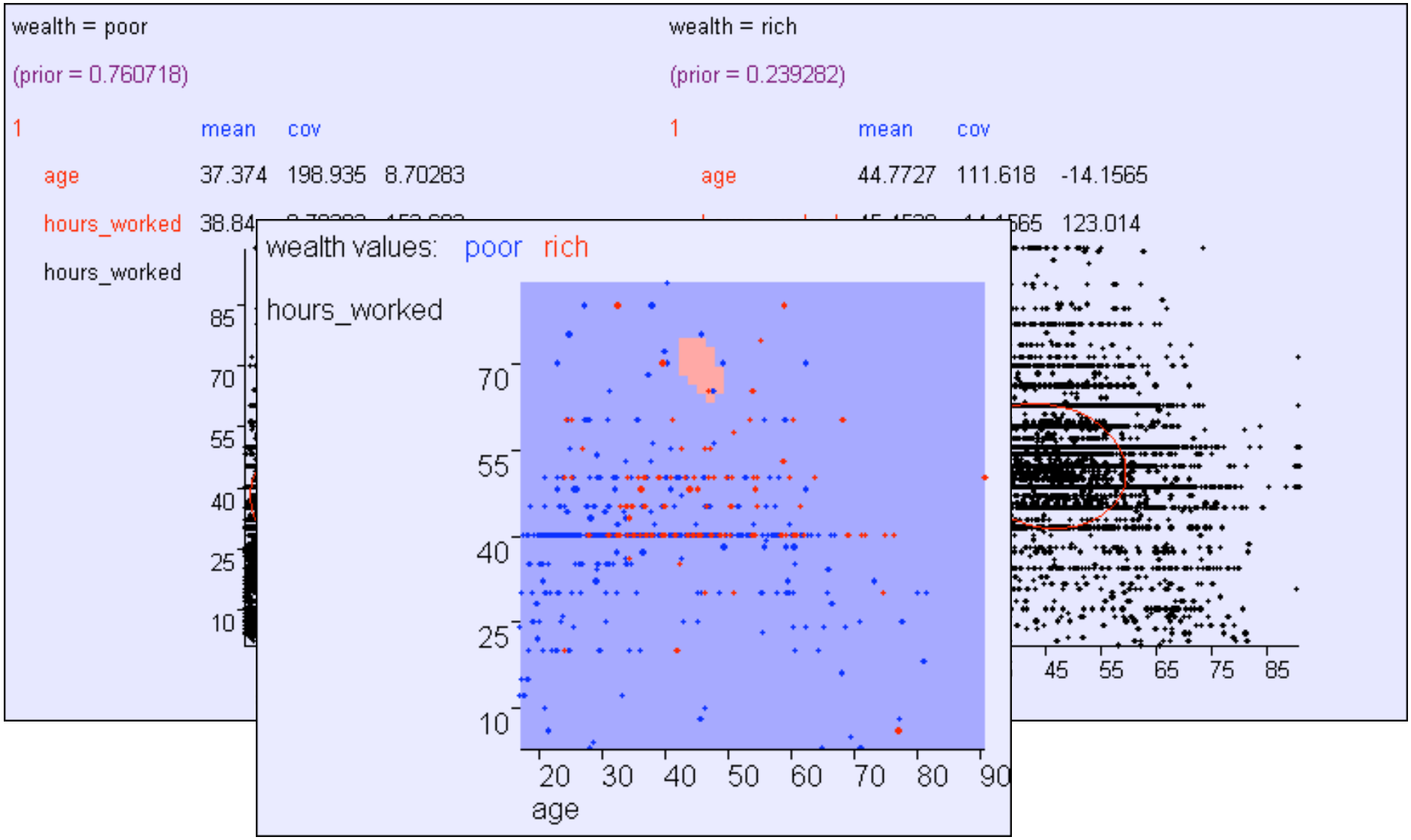
(prior = 0.239282)

1	mean	cov	
age	44.7727	111.618	-14.1565
hours_worked	45.4529	-14.1565	123.014

hours_worked



age, hours → wealth



age, hours → wealth

wealth = poor

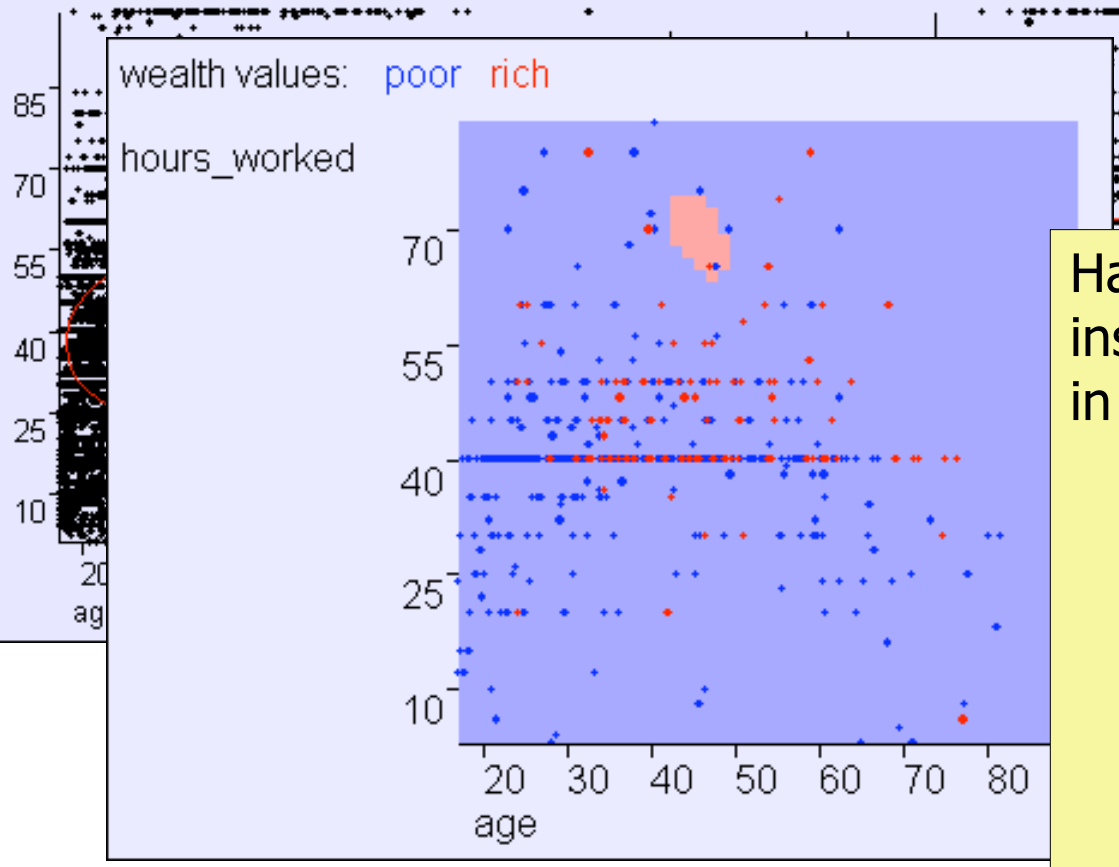
(prior = 0.760718)

1	mean	cov	
age	37.374	198.935	8.70283
hours_worked	38.84	8.70283	152.692

wealth = rich

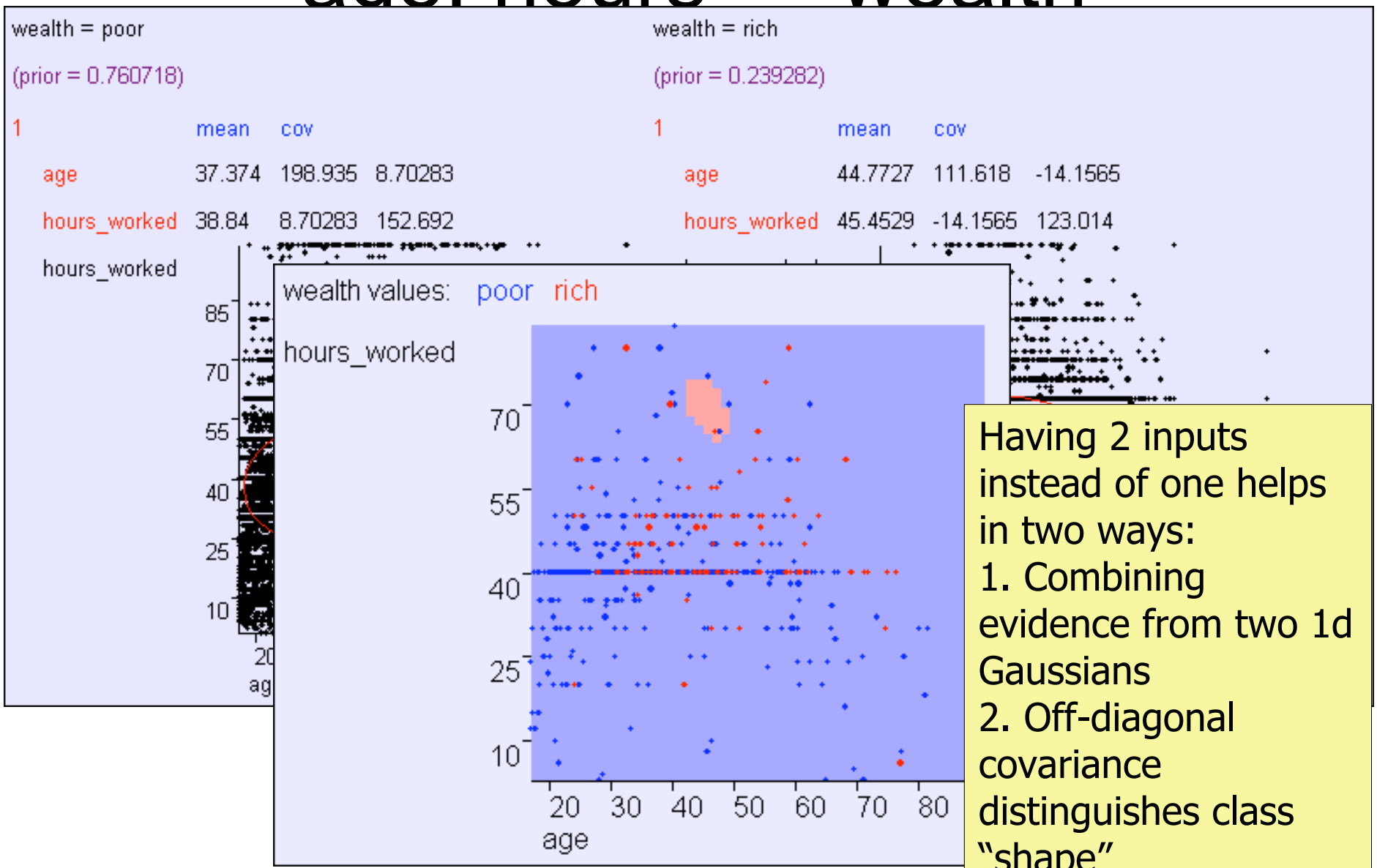
(prior = 0.239282)

1	mean	cov	
age	44.7727	111.618	-14.1565
hours_worked	45.4529	-14.1565	123.014



Having 2 inputs instead of one helps in two ways:

age, hours → wealth



Having 2 inputs instead of one helps in two ways:

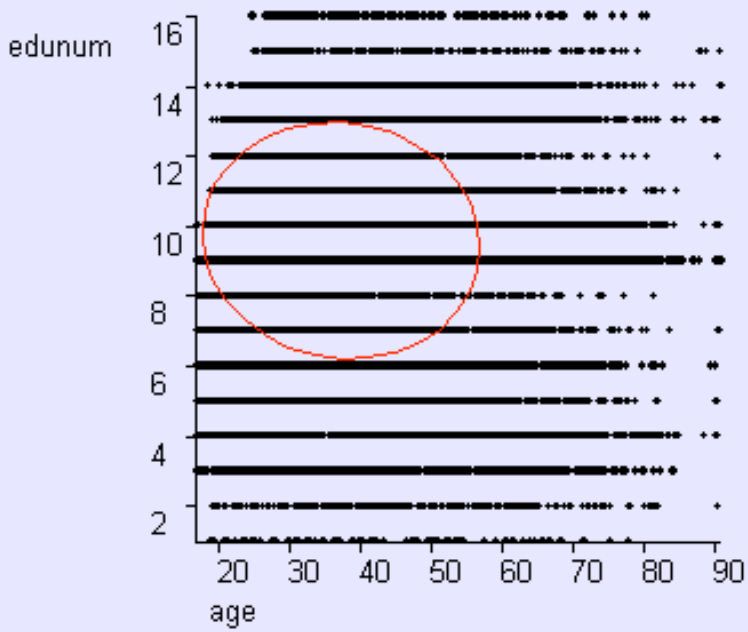
1. Combining evidence from two 1d Gaussians
2. Off-diagonal covariance distinguishes class "shape"

age, edunum → wealth

wealth = poor

(prior = 0.760718)

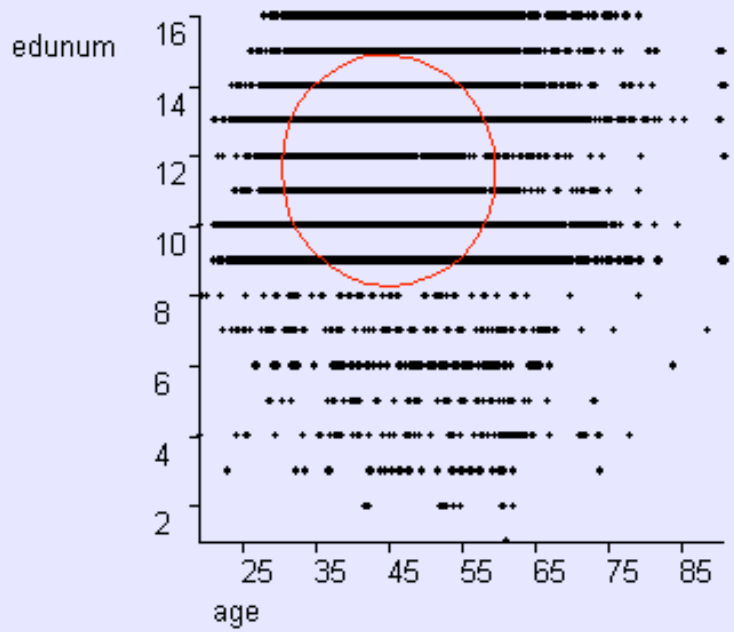
1	mean	cov	
age	37.374	198.935	-1.94765
edunum	9.59849	-1.94765	5.94225



wealth = rich

(prior = 0.239282)

1	mean	cov	
age	44.7727	111.618	-0.557852
edunum	11.6028	-0.557852	5.6769



age, edunum → wealth

wealth = poor

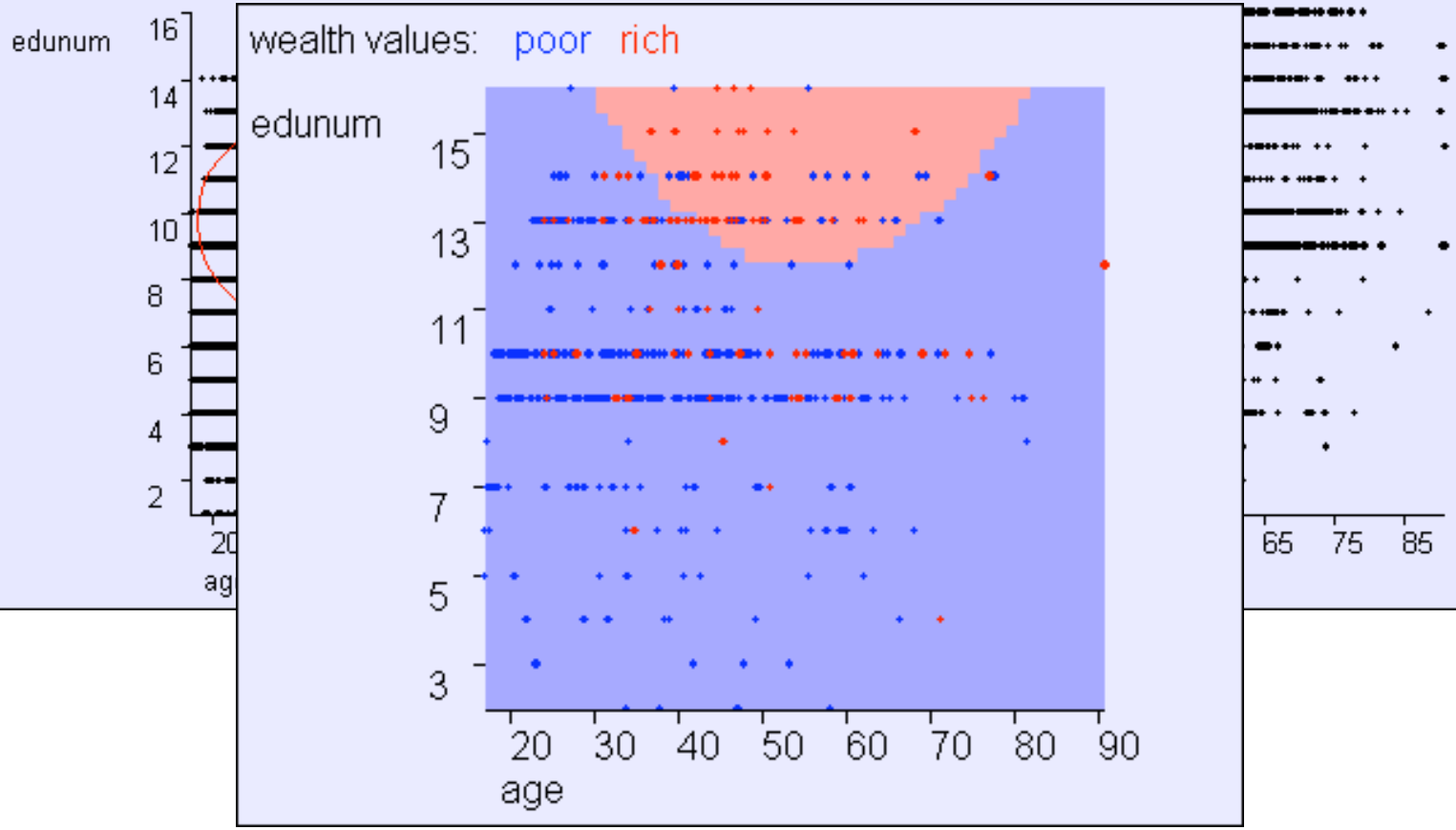
(prior = 0.760718)

1	mean	cov	
age	37.374	198.935	-1.94765
edunum	9.59849	-1.94765	5.94225

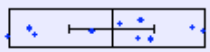
wealth = rich

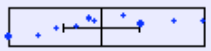
(prior = 0.239282)

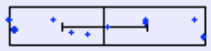
1	mean	cov	
age	44.7727	111.618	-0.557852
edunum	11.6028	-0.557852	5.6769



Accuracy

Name	Model	Parameters	FracRight			
age+hours	bayesclass	density=joint submodel=gauss gausstype=general	0.760452	+/-	0.00319521	

Name	Model	Parameters	FracRight			
age+hours+edunum	bayesclass	density=joint submodel=gauss gausstype=general	0.798513	+/-	0.00542432	

Name	Model	Parameters	FracRight			
a+h+e+capgain	bayesclass	density=joint submodel=gauss gausstype=general	0.793518	+/-	0.00319241	

Name	Model	Parameters	FracRight			
a+h+e+c+taxweight	bayesclass	density=joint submodel=gauss gausstype=general	0.793477	+/-	0.00321524