Learning Gaussian Bayes Classifiers

Andrew W. Moore
Associate Professor
School of Computer Science
Carnegie Mellon University
www.cs.cmu.edu/~awm
awm@cs.cmu.edu
412-268-7599

Sep 10th, 2001

Maximum Likelihood learning of Gaussians for Classification

- Why we should care
- 3 seconds to teach you a new learning algorithm
- What if there are 10,000 dimensions?
- What if there are categorical inputs?
- Examples "out the wazoo"

Why we should care

- One of the original "Data Mining" algorithms
- Very simple and effective
- Demonstrates the usefulness of our earlier groundwork

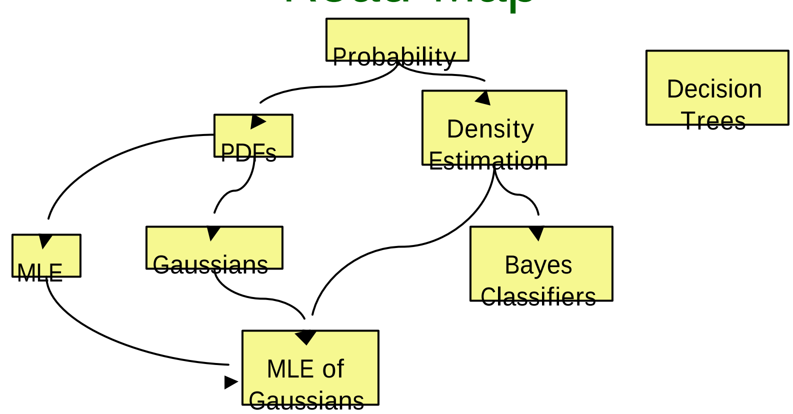
Where we were at the end of the MLE lecture...

	Categorical inputs only	Real-valued inputs only	Mixed Real / Cat okay
Inputs → Classifier → Predict category	Joint BC Naïve BC		Dec Tree
Inputs → Density Estimator → Prob-ability	Joint DE Naïve DE	Gauss DE	
Inputs → Regressor → Predict real no.			

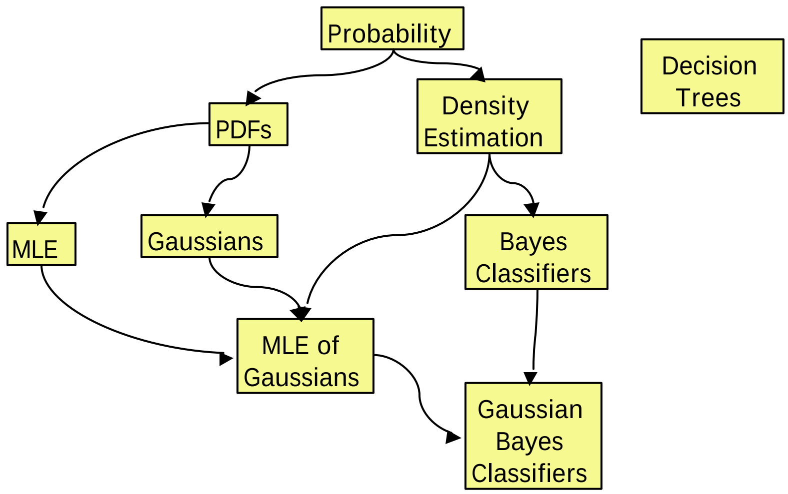
This lecture...

	Categorical inputs only	Real-valued inputs only	Mixed Real / Cat okay
Inputs → Classifier → Predict category	Joint BC Naïve BC	Gauss BC	Dec Tree
Inputs → Density Estimator → Prob-ability	Joint DE Naïve DE	Gauss DE	
Inputs → Regressor → Predict real no.			

Road Map



Road Map



Gaussian Bayes Classifier Assumption

- The i'th record in the database is created using the following algorithm

1. Generate the output (the "class") by drawing y_i~$Multinomial(p_1,p_2,\ldots p_{Ny})$
2. Generate the inputs from a Gaussian PDF that depends on the value of y_i :

$$\boldsymbol{x}_i \sim N(m_i, S_i).$$

Test your understanding. Given N_y classes and m input attributes, how many distinct scalar parameters need to be estimated?

MLE Gaussian Bayes Classifier

Let DB_i = Subset of database DB in which the output class is y = i

$p_i^{mle} = \frac{|DB_i|}{|DB|}$

th … created

g …

1. Generate the output (th… "class") by drawing y_i~$Multinomial(p_1,p_2,...p_{Ny})$
2. Generate the inputs from a Gaussian PDF that depends on the value of y_i :

$$\boldsymbol{x}_i \sim N(m_i, S_i).$$

Test your understanding. Given N_y classes and m input attributes, how many distinct scalar parameters need to be estimated?

MLE Gaussian Bayes Classifier

Let DB_i = Subset of database DB in which the output class is y = i

the database is created

g algorithm

(m_i^{mle}, S_i^{mle})= MLE Gaussian for DB_i

$p_{Ny})$

2. Generate the inputs fr… Gaussian PDF that depends on the v…ue of y_i :

$$\boldsymbol{x}_i \sim N(m_i, S_i).$$

Test your understanding. Given N_y classes and m input attributes, how many distinct scalar parameters need to be estimated?

MLE Gaussian Bayes Classifier

Let DB_i = Subset of database DB in which the output class is y = i

the database is created

g algorithm

$(\mu_i^{mle}, \Sigma_i^{mle})$ = MLE Gaussian for DB_i

2. Generate the inputs fr Gaussian PDF that depends on the v ue of y_i :

$$\mathbf{x}_i \sim N(m_i, S_i).$$

$$\boldsymbol{\mu}_i^{mle} = \frac{1}{|\mathrm{DB}_i|} \sum_{\mathbf{x}_k \in \mathrm{DB}_i}^{R} \mathbf{x}_k$$

$$\mathbf{S}_i^{mle} = \frac{1}{|\mathrm{DB}_i|} \sum_{\mathbf{x}_k \in \mathrm{DB}_i}^{R} \left(\mathbf{x}_k - \boldsymbol{\mu}_i^{mle}\right)\left(\mathbf{x}_k - \boldsymbol{\mu}_i^{mle}\right)^T$$

ndin

many d

Gaussian Bayes Classification

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

Gaussian Bayes Classification

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

$$P(y=i\,|\,\mathbf{x}) = \frac{\dfrac{1}{(2p)^{m/2}\,\|\,\mathbf{S}_i\,\|^{1/2}}\exp\left[-\dfrac{1}{2}(\mathbf{x}_k-\boldsymbol{\mu}_i)^T\,\mathbf{S}_i(\mathbf{x}_k-\boldsymbol{\mu}_i)\right]p_i}{p(\mathbf{x})}$$

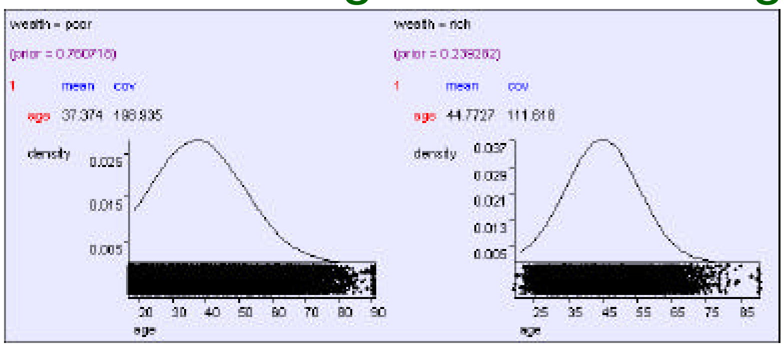
How do we deal with that?

Here is a dataset

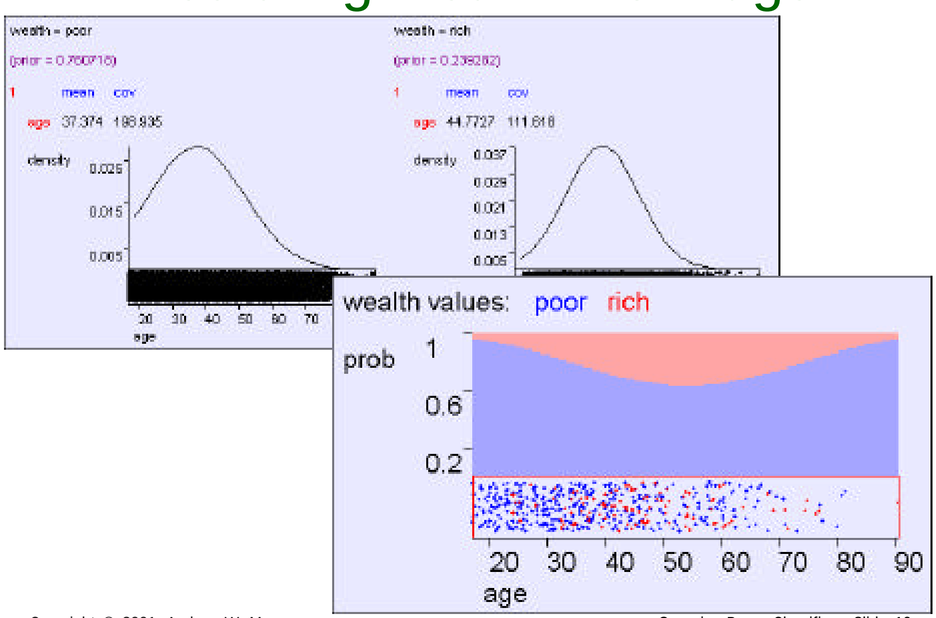
age	employme	education	edur	marital	...	job	relation	race	gender	hour	country	wealth
					...							
39	State_gov	Bachelors	13	Never_mar	...	Adm_cleric	Not_in_far	White	Male	40	United_Sta	poor
51	Self_emp_	Bachelors	13	Married	...	Exec_man	Husband	White	Male	13	United_Sta	poor
39	Private	HS_grad	9	Divorced	...	Handlers_c	Not_in_far	White	Male	40	United_Sta	poor
54	Private	11th	7	Married	...	Handlers_c	Husband	Black	Male	40	United_Sta	poor
28	Private	Bachelors	13	Married	...	Prof_speci	Wife	Black	Female	40	Cuba	poor
38	Private	Masters	14	Married	...	Exec_man	Wife	White	Female	40	United_Sta	poor
50	Private	9th	5	Married_sp	...	Other_serv	Not_in_far	Black	Female	16	Jamaica	poor
52	Self_emp_	HS_grad	9	Married	...	Exec_man	Husband	White	Male	45	United_Sta	rich
31	Private	Masters	14	Never_mar	...	Prof_speci	Not_in_far	White	Female	50	United_Sta	rich
42	Private	Bachelors	13	Married	...	Exec_man	Husband	White	Male	40	United_Sta	rich
37	Private	Some_coll	10	Married	...	Exec_man	Husband	Black	Male	80	United_Sta	rich
30	State_gov	Bachelors	13	Married	...	Prof_speci	Husband	Asian	Male	40	India	rich
24	Private	Bachelors	13	Never_mar	...	Adm_cleric	Own_child	White	Female	30	United_Sta	poor
33	Private	Assoc_acc	12	Never_mar	...	Sales	Not_in_far	Black	Male	50	United_Sta	poor
41	Private	Assoc_voc	11	Married	...	Craft_repai	Husband	Asian	Male	40	*MissingVa	rich
34	Private	7th_8th	4	Married	...	Transport_	Husband	Amer_India	Male	45	Mexico	poor
26	Self_emp_	HS_grad	9	Never_mar	...	Farming_fis	Own_child	White	Male	35	United_Sta	poor
33	Private	HS_grad	9	Never_mar	...	Machine_o	Unmarried	White	Male	40	United_Sta	poor
38	Private	11th	7	Married	...	Sales	Husband	White	Male	50	United_Sta	poor
44	Self_emp_	Masters	14	Divorced	...	Exec_man	Unmarried	White	Female	45	United_Sta	rich
41	Private	Doctorate	16	Married	...	Prof_speci	Husband	White	Male	60	United_Sta	rich
:	:	:	:	:	:	:	:	:	:	:	:	:

48,000 records, 16 attributes [Kohavi 1995]

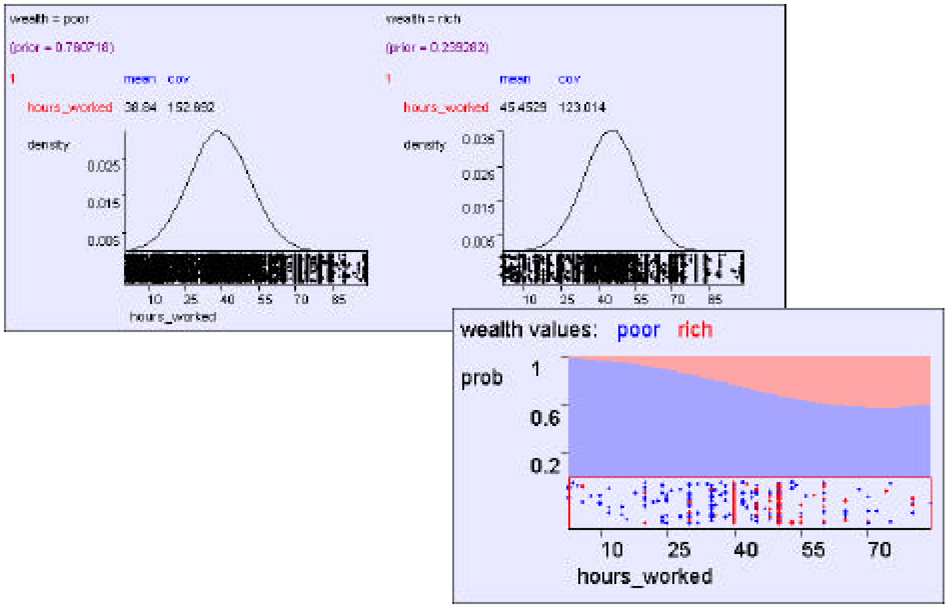
Predicting wealth from age



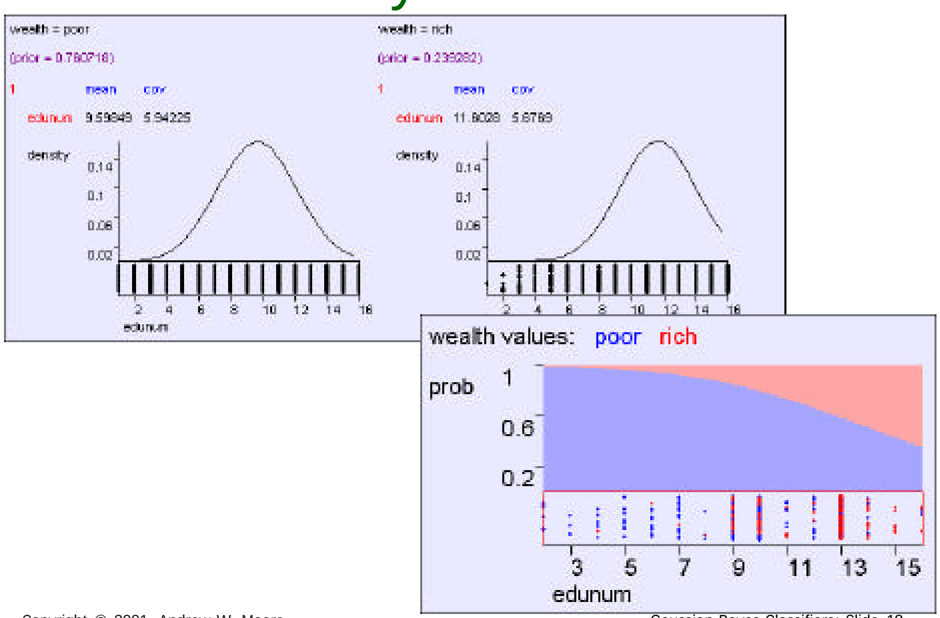
Predicting wealth from age



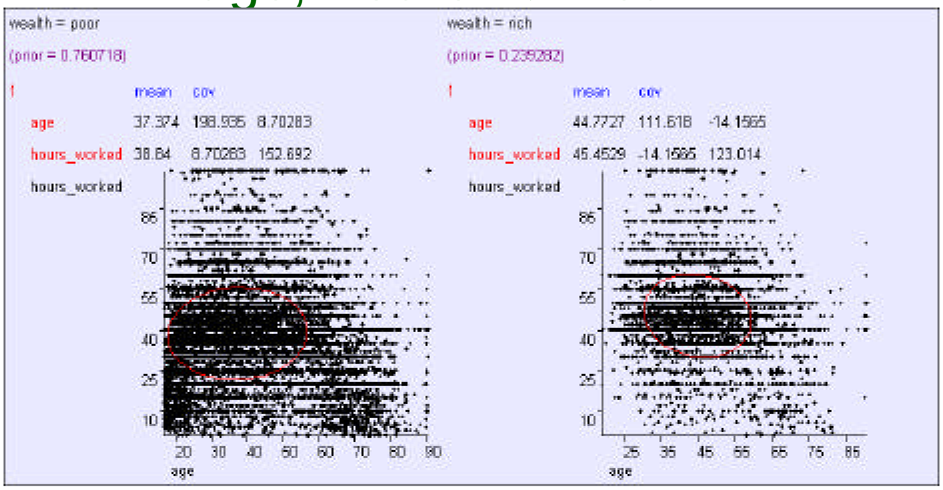
Wealth from hours worked



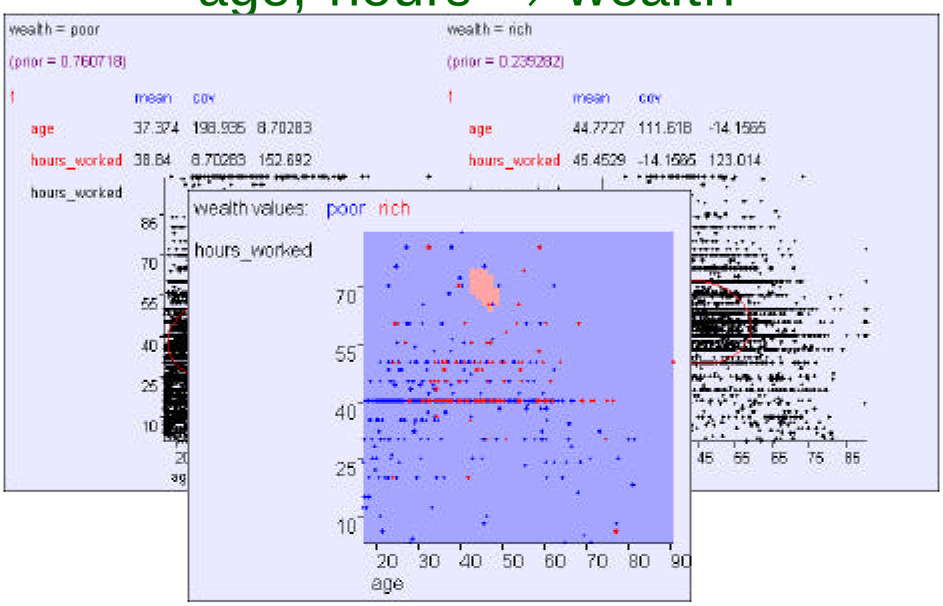
Wealth from years of education



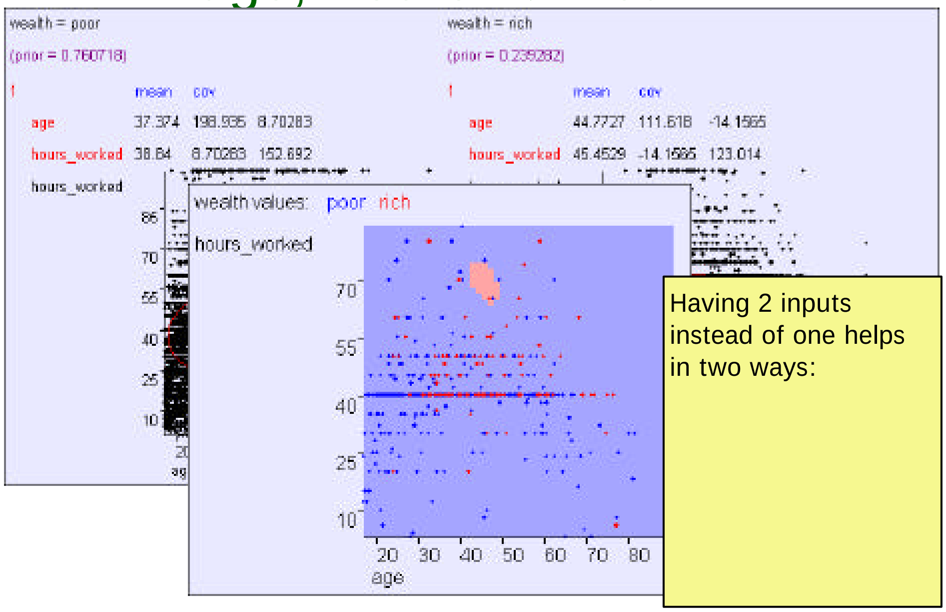
age, hours → wealth



age, hours → wealth

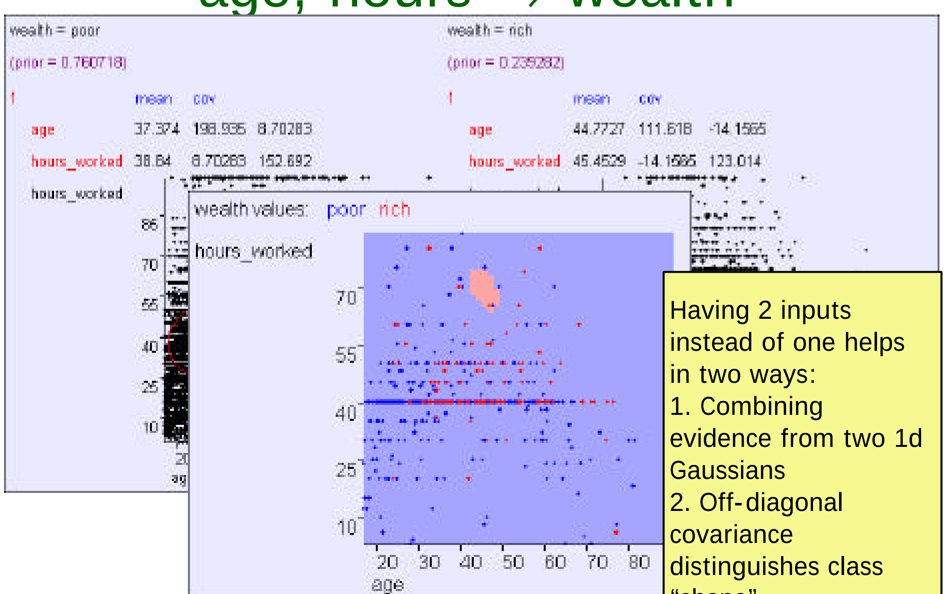


age, hours → wealth



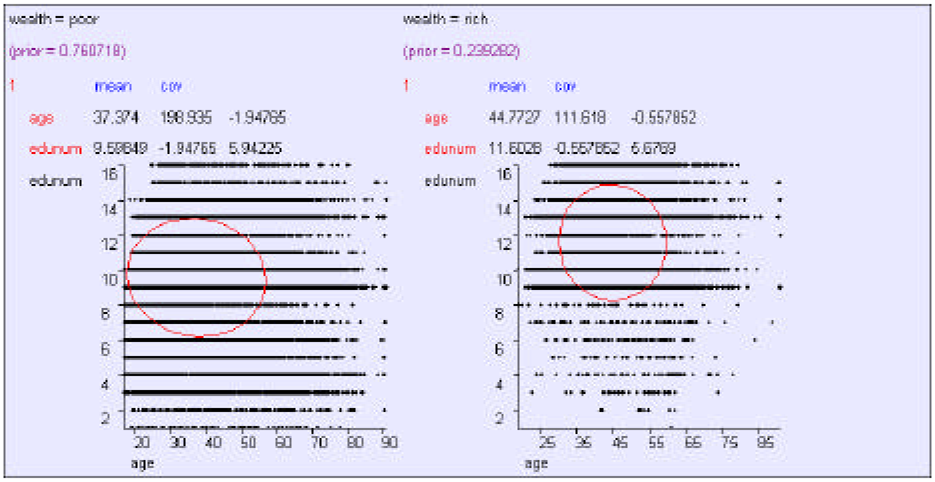
Having 2 inputs instead of one helps in two ways:

age, hours → wealth

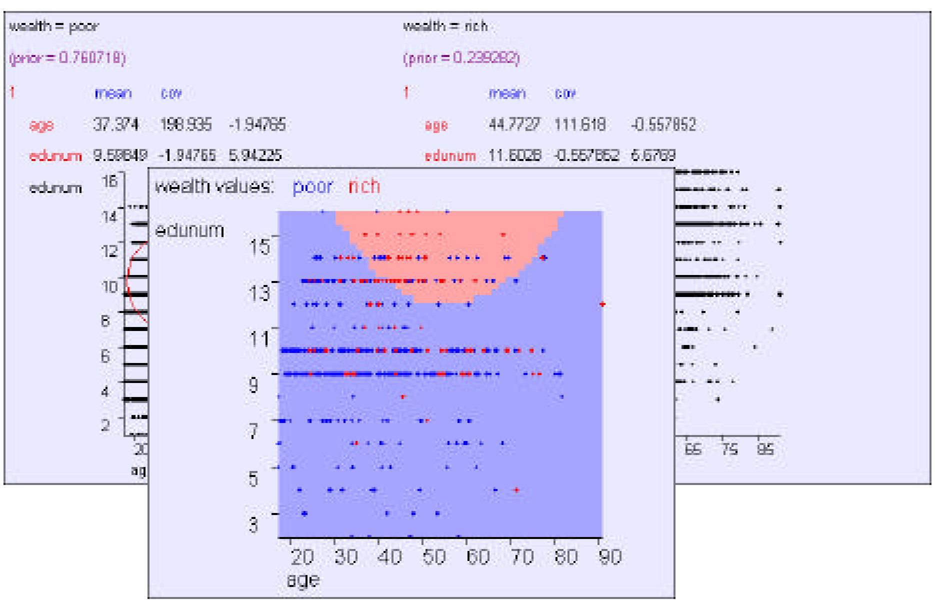


Having 2 inputs instead of one helps in two ways:
1. Combining evidence from two 1d Gaussians
2. Off-diagonal covariance distinguishes class "shape"

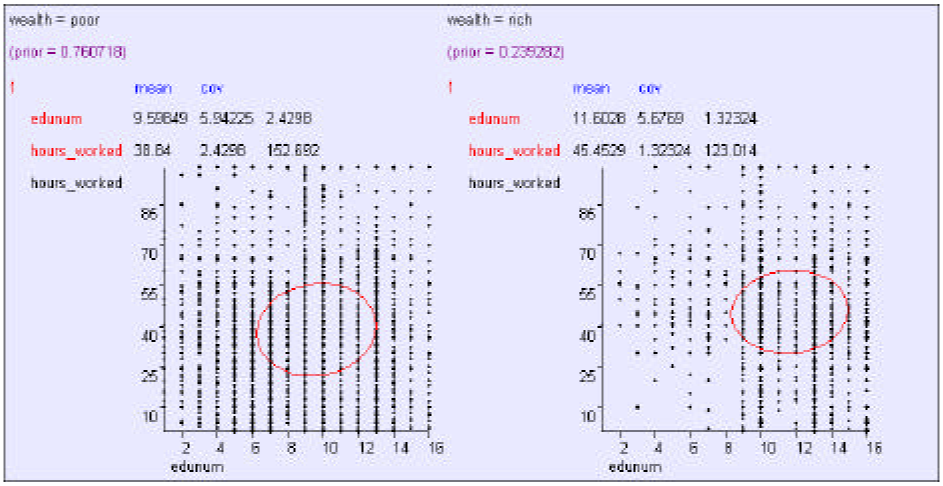
age, edunum → wealth



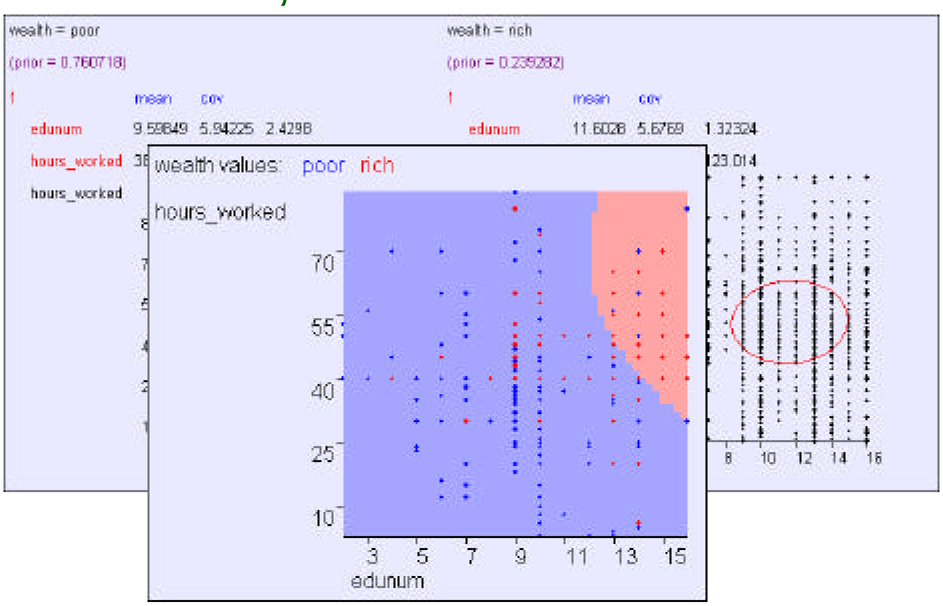
age, edunum → wealth



hours, edunum → wealth



hours, edunum → wealth



Accuracy

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

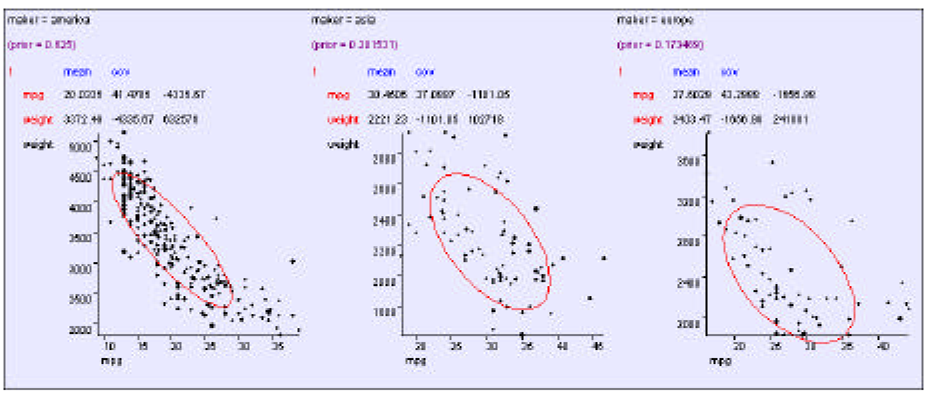
Name	Model	Parameters	FracRight
age+hours	bayesclass	density=joint submodel=gauss gausstype=general	0.760452 +/- 0.00319521

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

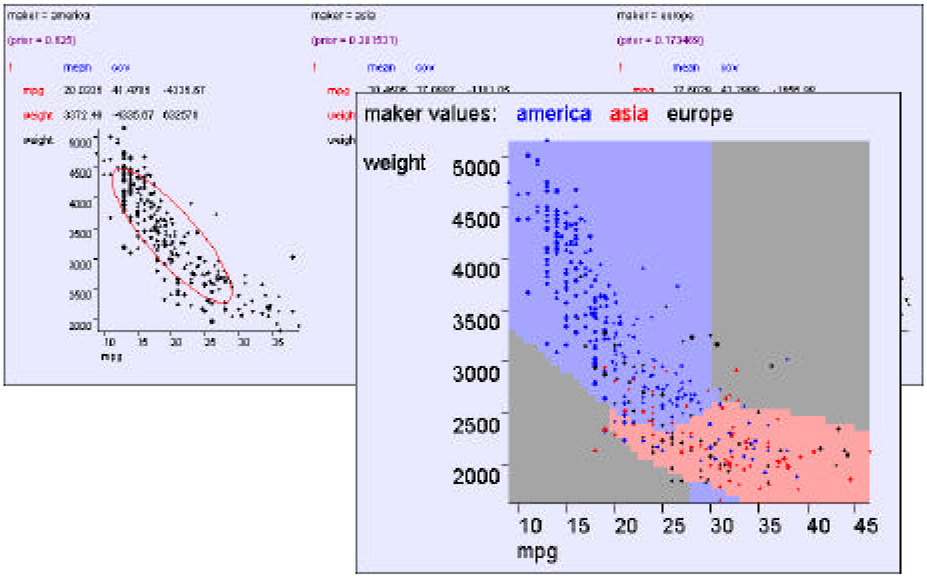
Name	Model	Parameters	FracRight
a+h+e+capgain	bayesclass	density=joint submodel=gauss gausstype=general	0.793518 +/- 0.00319241

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

An "MPG" example



An “MPG” example



An “MPG” example

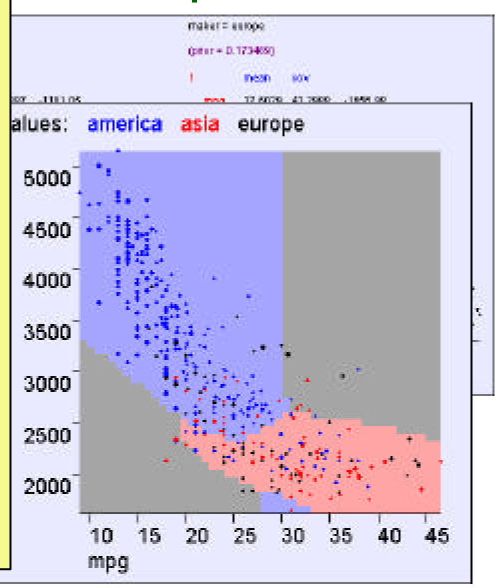
Things to note:

•Class Boundaries can be weird shapes (hyperconic sections)

•Class regions can be non-simply-connected

•But it’s impossible to model arbitrarily weirdly shaped regions

•**Test your understanding:** With one input, must classes be simply connected?



Overfitting dangers

- Problem with “Joint” Bayes classifier:

  #parameters exponential with #dimensions.

  This means we just memorize the training data, and can overfit.

Overfitting dangers

- Problem with “Joint” Bayes classifier:

  #parameters exponential with #dimensions.

  This means we just memorize the training data, and can overfit.

- Problemette with Gaussian Bayes classifier:

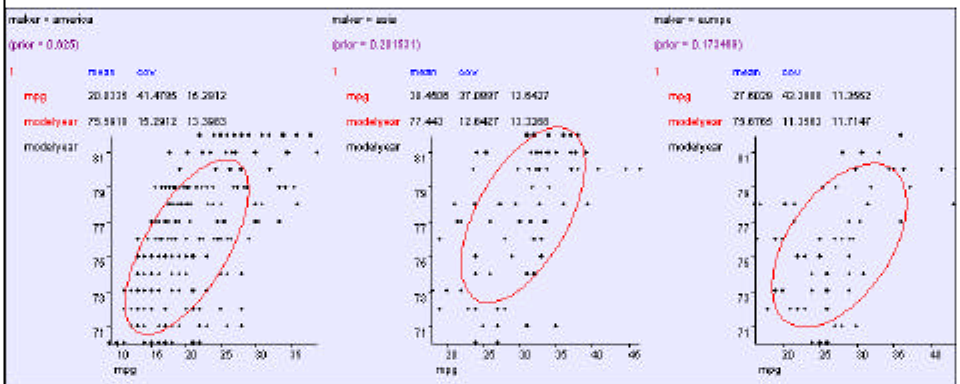
  #parameters quadratic with #dimensions.

  With 10,000 dimensions and only 1,000 datapoints we could overfit.

Question: Any suggested solutions?

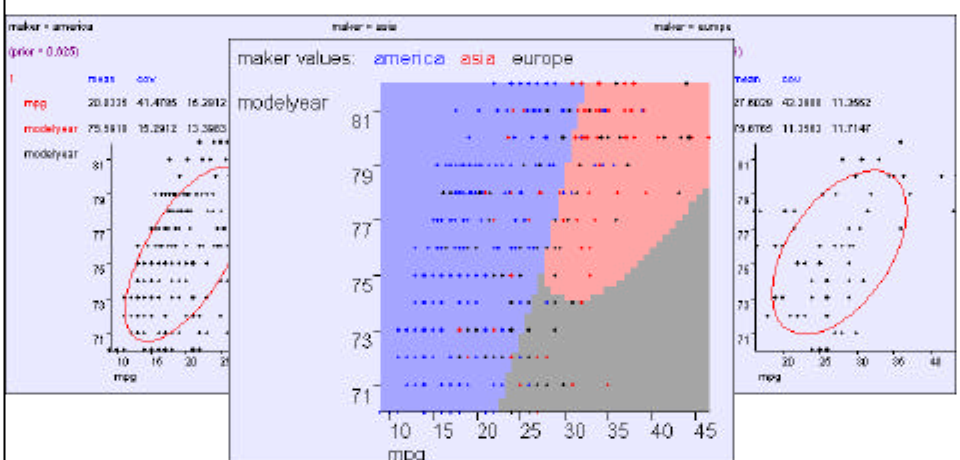
General: $O(m^2)$ parameters

$$\mathbf{S} = \begin{pmatrix} s^2{}_1 & s_{12} & \cdots & s_{1m} \\ s_{12} & s^2{}_2 & \cdots & s_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ s_{1m} & s_{2m} & \cdots & s^2{}_m \end{pmatrix}$$



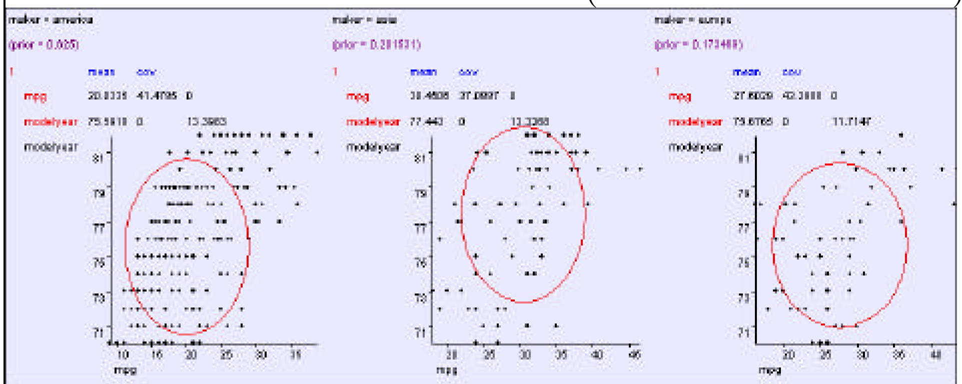
General: $O(m^2)$ parameters

$$\mathbf{S} = \begin{pmatrix} s^2{}_1 & s_{12} & \cdots & s_{1m} \\ s_{12} & s^2{}_2 & \cdots & s_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ s_{1m} & s_{2m} & \cdots & s^2{}_m \end{pmatrix}$$



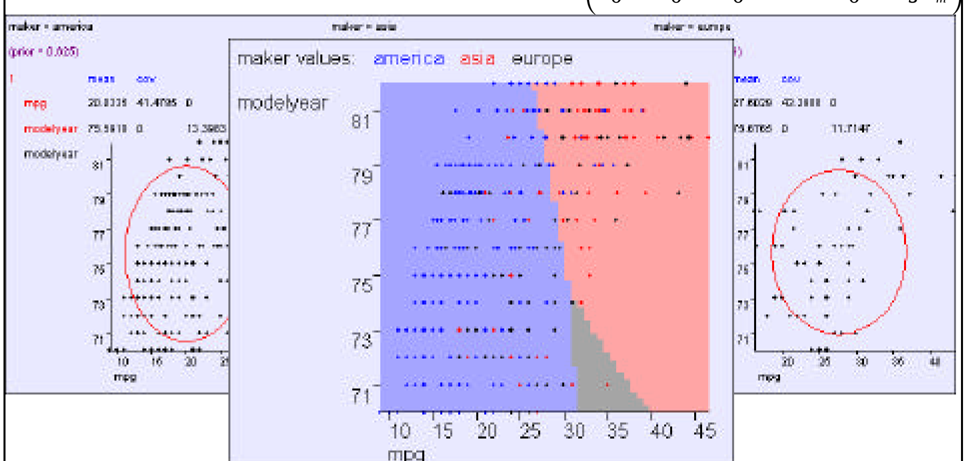
Aligned: *O(m)* parameters

$$\mathbf{S} = \begin{pmatrix} s^2{}_1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & s^2{}_2 & 0 & \cdots & 0 & 0 \\ 0 & 0 & s^2{}_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & s^2{}_{m-1} & 0 \\ 0 & 0 & 0 & \cdots & 0 & s^2{}_m \end{pmatrix}$$



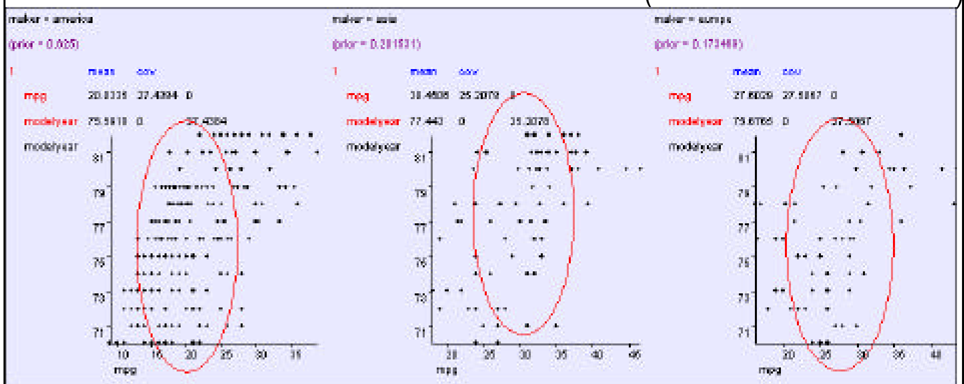
Aligned: *O(m)* parameters

$$\mathbf{S} = \begin{pmatrix} s^2{}_1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & s^2{}_2 & 0 & \cdots & 0 & 0 \\ 0 & 0 & s^2{}_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & s^2{}_{m-1} & 0 \\ 0 & 0 & 0 & \cdots & 0 & s^2{}_m \end{pmatrix}$$



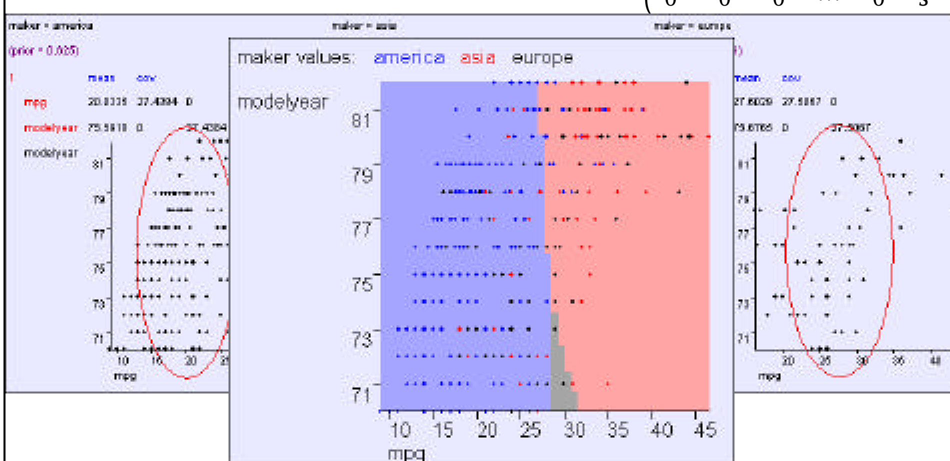
Spherical: *O(1)* cov parameters

$$\Sigma = \begin{pmatrix} \sigma^2 & 0 & 0 & \cdots & 0 & 0 \\ 0 & \sigma^2 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \sigma^2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma^2 & 0 \\ 0 & 0 & 0 & \cdots & 0 & \sigma^2 \end{pmatrix}$$



Spherical: *O(1)* cov parameters

$$\Sigma = \begin{pmatrix} \sigma^2 & 0 & 0 & \cdots & 0 & 0 \\ 0 & \sigma^2 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \sigma^2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma^2 & 0 \\ 0 & 0 & 0 & \cdots & 0 & \sigma^2 \end{pmatrix}$$



BCs that have both real and categorical inputs?

	Categorical inputs only	Real-valued inputs only	Mixed Real / Cat okay
Inputs Classifier Predict category	Joint BC Naïve BC	Gauss BC	Dec Tree BC Here???
Inputs Density Estimator Prob-ability	Joint DE Naïve DE	Gauss DE	
Inputs Regressor Predict real no.			

BCs that have both real and categorical inputs?

	Categorical inputs only	Real-valued inputs only	Mixed Real / Cat okay
Inputs Classifier Predict category	Joint BC Naïve BC	Gauss BC	Dec Tree BC Here???
Inputs Density Estimator Prob-ability	Joint DE Naïve DE	Gauss DE	
Inputs Regressor Predict real no.			

Easy!

Guess how?

BCs that have both real and categorical inputs?

	Categorical inputs only	Real-valued inputs only	Mixed Real / Cat okay
Inputs → Classifier → Predict category	Joint BC Naïve BC	Gauss BC	Dec Tree Gauss/Joint BC Gauss Naïve BC
Inputs → Density Estimator → Prob-ability	Joint DE Naïve DE	Gauss DE Gauss DE	Gauss/Joint DE Gauss Naïve DE
Inputs → Regressor → Predict real no.			

BCs that have both real and categorical inputs?

	Categorical inputs only	Real-valued inputs only	Mixed Real / Cat okay
Inputs → Classifier → Predict category	Joint BC Naïve BC	Gauss BC	Dec Tree Gauss/Joint BC Gauss Naïve BC
Inputs → Density Estimator → Prob-ability	Joint DE Naïve DE	Gauss DE Gauss DE	Gauss/Joint DE Gauss Naïve DE
Inputs → Regressor → Predict real no.			

Mixed Categorical / Real Density Estimation

- Write $\boldsymbol{x} = (\boldsymbol{u},\boldsymbol{v}) = (\underbrace{u_1, u_2, \ldots u_q}_{\text{Real valued}}, \underbrace{v_1, v_2 \ldots v_{m-q}}_{\text{Categorical valued}})$

$$P(\boldsymbol{x} \mid M) = P(\boldsymbol{u},\boldsymbol{v} \mid M)$$

(where M is any Density Estimation Model)

Not sure which tasty DE to enjoy? Try our…

Joint / Gauss DE Combo

$$P(\boldsymbol{u},\boldsymbol{v} \mid M) = \underbrace{P(\boldsymbol{u} \mid \boldsymbol{v}, M)}_{\text{Gaussian with parameters depending on } \boldsymbol{v}} \underbrace{P(\boldsymbol{v} \mid M)}_{\text{Big “m-q”-dimensional lookup table}}$$

MLE learning of the Joint / Gauss DE Combo

$$P(\mathbf{u},\mathbf{v}\,|M) = P(\mathbf{u}\,|\mathbf{v}\,,M)\;P(\mathbf{v}\,|M)$$

$\mu_{\mathbf{v}}$	=	Mean of ***u*** among records matching ***v***
$\Sigma_{\mathbf{v}}$	=	Cov. of ***u*** among records matching ***v***
$q_{\mathbf{v}}$	=	Fraction of records that match ***v***

$$\mathbf{u}\,|\mathbf{v}\,,M \sim N(\mu_{\mathbf{v}}\,,\,\Sigma_{\mathbf{v}})\quad,\; P(\mathbf{v}\,|M) = q_{\mathbf{v}}$$

MLE learning of the Joint / Gauss DE Combo

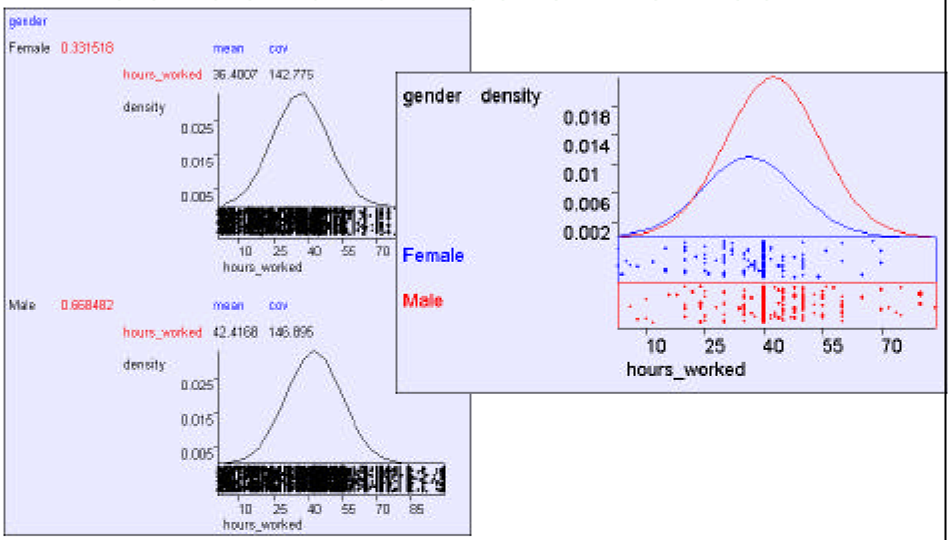
$$P(\mathbf{u},\mathbf{v}\,|M) = P(\mathbf{u}\,|\mathbf{v}\,,M)\;P(\mathbf{v}\,|M)$$

$\mu_{\mathbf{v}}$	=	Mean of ***u*** among records matching ***v***	=	$\dfrac{1}{R_{\mathbf{v}}}\sum_{k \text{ s.t. } \mathbf{v}_k=\mathbf{v}} \mathbf{u}_k$
$\Sigma_{\mathbf{v}}$	=	Cov. of ***u*** among records matching ***v***	=	$\dfrac{1}{R_{\mathbf{v}}}\sum_{k \text{ s.t. } \mathbf{v}_k=\mathbf{v}} (\mathbf{u}_k-\boldsymbol{\mu}_{\mathbf{v}})(\mathbf{u}_k-\boldsymbol{\mu}_{\mathbf{v}})^T$
$q_{\mathbf{v}}$	=	Fraction of records that match ***v***	=	$\dfrac{R_{\mathbf{v}}}{R}$

$R_{\mathbf{v}}$ = # records that match **v**

$$\mathbf{u}\,|\mathbf{v}\,,M \sim N(\mu_{\mathbf{v}}\,,\,\Sigma_{\mathbf{v}})\quad,\; P(\mathbf{v}\,|M) = q_{\mathbf{v}}$$

Gender and Hours Worked*



*As with all the results from the UCI "adult census" dataset, we can't draw any real-world conclusions since it's such a non-real-world sample

What we just did → Joint / Gauss DE Combo

What we do next →

Joint / Gauss **BC** Combo

Joint / Gauss **BC** Combo

$$P(Y=i \mid \mathbf{u},\mathbf{v}) = \frac{p(\mathbf{u},\mathbf{v} \mid M_i)P(Y=i)}{p(\mathbf{u},\mathbf{v})}$$

$$= \frac{p(\mathbf{u}, \mid \mathbf{v}, M_i)\, p(\mathbf{v} \mid M_i) P(Y=i)}{p(\mathbf{u},\mathbf{v})}$$

$$= \frac{N(\mathbf{u}; \boldsymbol{\mu}_{i,\mathbf{v}}, \mathbf{S}_{i,\mathbf{v}})\, q_{i,\mathbf{v}}\ p_i}{p(\mathbf{u},\mathbf{v})}$$

Joint / Gauss **BC** Combo

$$P(Y=i\,|\,\mathbf{u},\mathbf{v}) = \frac{p(\mathbf{u},\mathbf{v}\,|\,M_i)P(Y=i)}{p(\mathbf{u},\mathbf{v})}$$

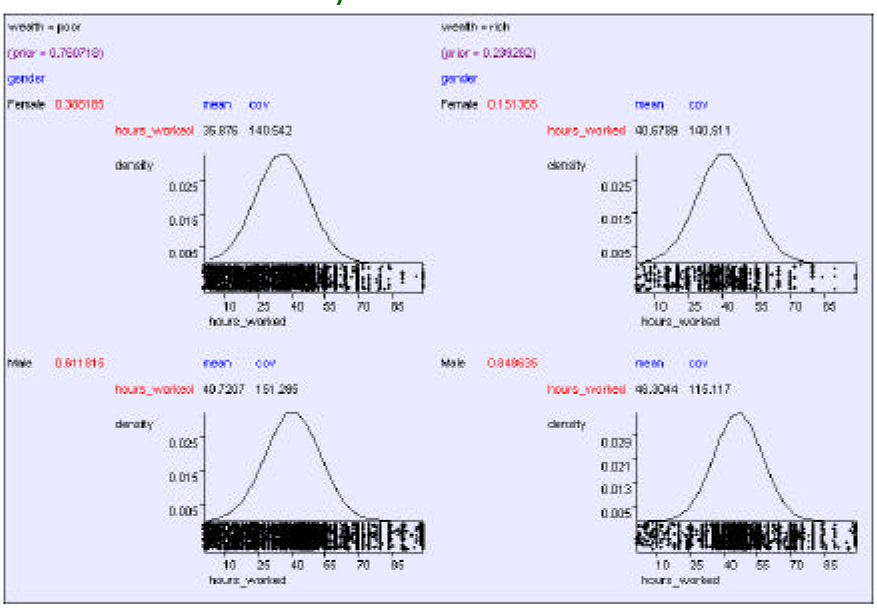
$$= \frac{p(\mathbf{u},|\,\mathbf{v},M_i)\,p(\mathbf{v}\,|\,M_i)P(Y=i)}{p(\mathbf{u},\mathbf{v})}$$

$$= \frac{N(\mathbf{u};\boldsymbol{\mu}_{i,\mathbf{v}},\mathbf{S}_{i,\mathbf{v}})\,q_{i,\mathbf{v}}\ p_i}{p(\mathbf{u},\mathbf{v})}$$

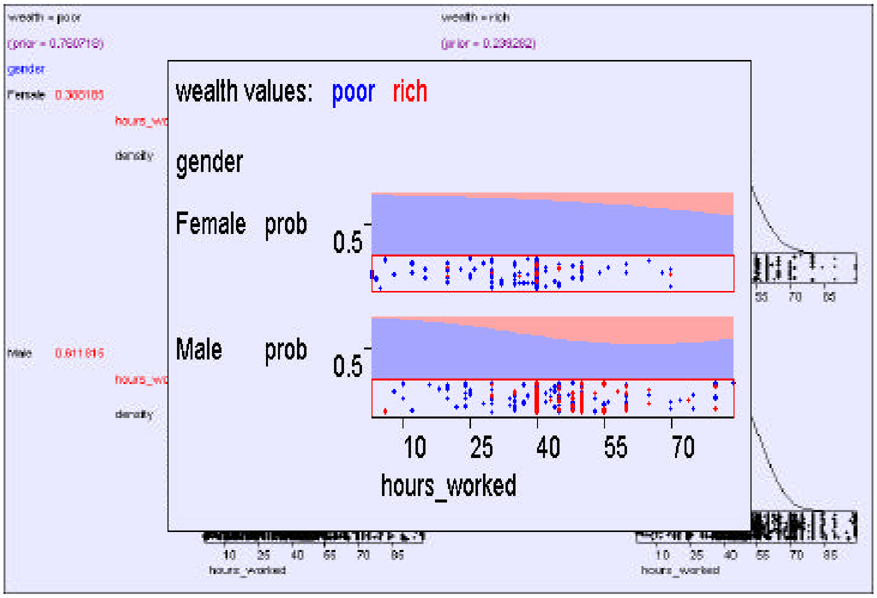
$m_{i,v}$	=	Mean of ***u*** among records matching ***v*** and in which *y=i*
$S_{i,v}$	=	Cov. of ***u*** among records matching ***v*** and in which *y=i*
$q_{i,v}$	=	Fraction of "*y=i*" records that match ***v***
p_i	=	Fraction of records that match "*y=i*"

Rather so-so-notation for "Gaussian with mean $m_{i,v}$ and covariance $S_{i,v}$ evaluated at ***u***"

Gender, Hours→Wealth



Gender, Hours→Wealth



Joint / Gauss DE Combo and Joint / Gauss BC Combo: The downside

- (Yawn...we've done this before...)

  More than a few categorical attributes blah blah blah massive table blah blah lots of parameters blah blah just memorize training data blah blah blah do worse on future data blah blah need to be more conservative blah

Naïve/Gauss combo for Density Estimation

Real　Categorical

$$p(\mathbf{u},\mathbf{v}\mid M)=\left(\prod_{j=1}^{q} p(u_j\mid M)\right)\left(\prod_{j=1}^{m-q} P(v_j\mid M)\right)$$

$$u_j \mid M \sim N(m_j, s_j^2) \quad v_j \mid M \sim \text{Multinomial}[q_{j1}, q_{j2}, \ldots, q_{jN_j}]$$

How many parameters?

Naïve/Gauss combo for Density Estimation

Real　Categorical

$$p(\mathbf{u},\mathbf{v}\mid M)=\left(\prod_{j=1}^{q} p(u_j\mid M)\right)\left(\prod_{j=1}^{m-q} P(v_j\mid M)\right)$$

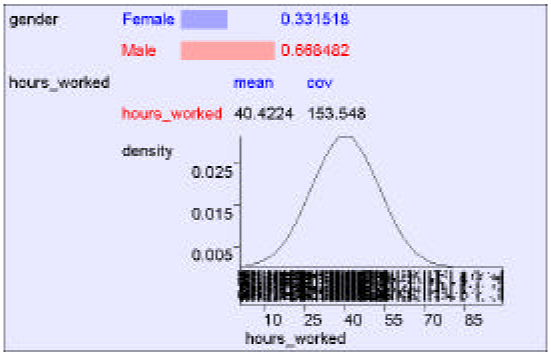
$$u_j \mid M \sim N(m_j, s_j^2) \quad v_j \mid M \sim \text{Multinomial}[q_{j1}, q_{j2}, \ldots, q_{jN_j}]$$

$$m_j = \frac{1}{R}\sum_k u_{kj}$$

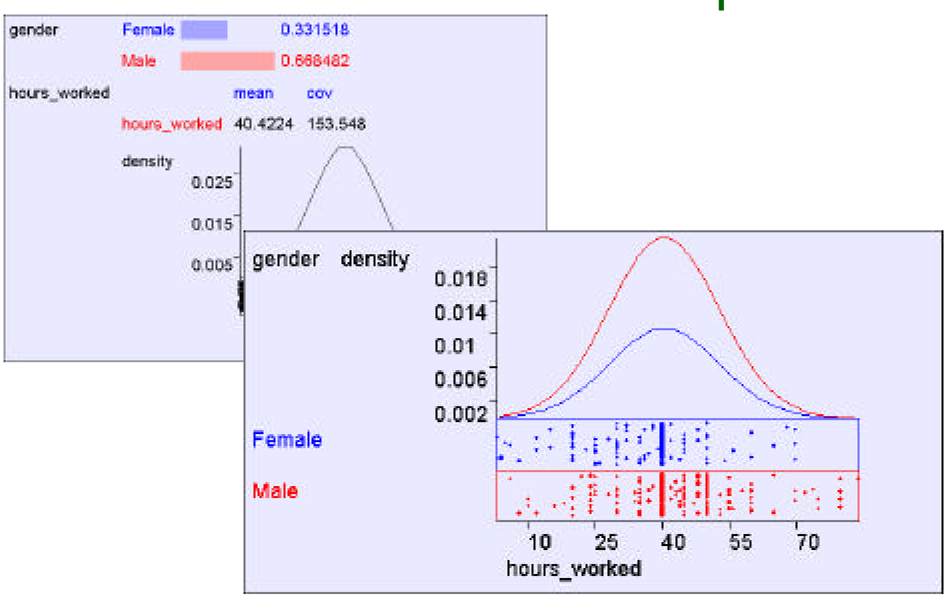
$$s_j^2 = \frac{1}{R}\sum_k (u_{kj} - m_j)^2$$

$$q_{jh} = \frac{\#\text{ of records in which } v_j = h}{R}$$

Naïve/Gauss DE Example



Naïve/Gauss DE Example



Naïve / Gauss BC

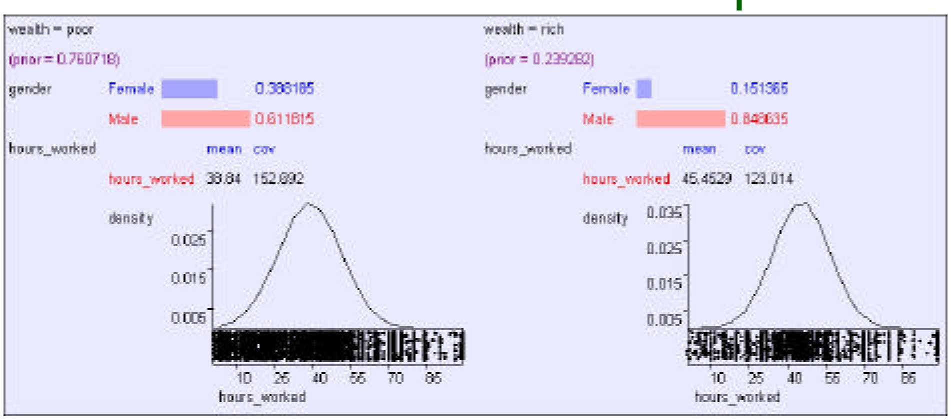
$$P(Y=i\,|\,\mathbf{u},\mathbf{v}) = \frac{p(\mathbf{u},\mathbf{v}\,|\,Y=i)P(Y=i)}{p(\mathbf{u},\mathbf{v})}$$

$$= \frac{1}{p(\mathbf{u},\mathbf{v})} \prod_{j=1}^{q} p(u_j\,|\,m_{ij}, s_{ij}^2) \quad \prod_{j=1}^{m-q} P(v_j\,|\,\mathbf{q}_{ij}) \quad P(Y=i)$$

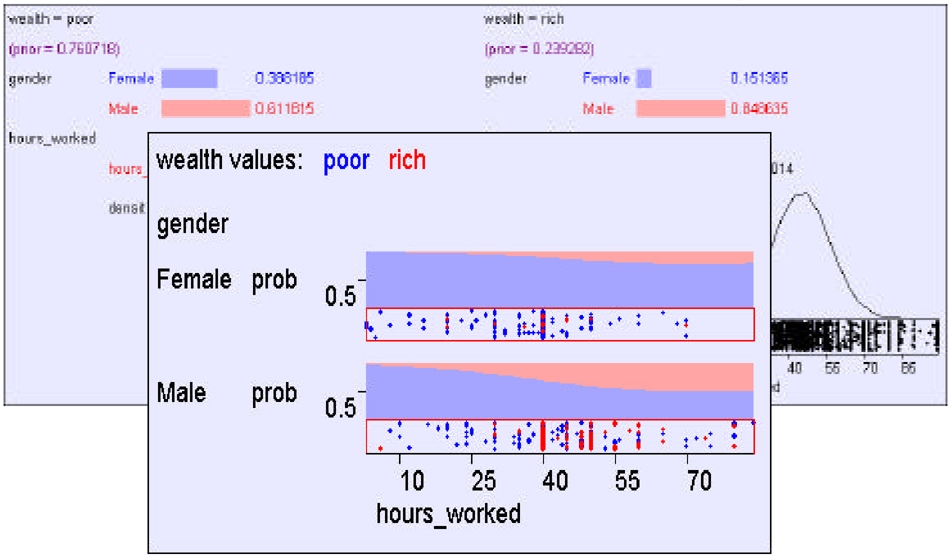
$$= \frac{1}{p(\mathbf{u},\mathbf{v})} \prod_{j=1}^{q} N(u_j; m_{ij}, s_{ij}^2) \quad \prod_{j=1}^{m-q} q_{ij}[v_j] \quad p_i$$

m_{ij}	= Mean of u_j among records in which $y=i$
s^2_{ij}	= Var. of u_j among records in which $y=i$
$q_{ij}[h]$	= Fraction of "$y=i$" records in which $v_j = h$
p_i	= Fraction of records that match "$y=i$"

Gauss / Naïve BC Example



Gauss / Naïve BC Example



Learn Wealth from 15 attributes

Name	Model	Parameters	FracRight
Model1	bayesclass	density=joint submodel=gauss gausstype=general	0.718009 +/- 0.00570714
Model2	bayesclass	density=naive submodel=gauss gausstype=general	0.832234 +/- 0.00288377
Model3	dtree	max_children=4 ne_splits=y max_pchance=0.05 adjust_chi=y max_nodes=50	0.850702 +/- 0.00364538

Learn Wealth from 15 attributes

Name	Model	Parameters	FracRight		
Model1	bayesclass	density=joint submodel=gauss gausstype=general	0.718009	+/-	0.00570714
Model2	bayesclass	density=naive submodel=gauss gausstype=general	0.832234	+/-	0.00288377
Model3	dtree	max_children=4 ne_splits=y max_pchance=0.05 adjust_chi=y max_nodes=50	0.850702	+/-	0.00364538

Same data, except all real values discretized to 3 levels

Name	Model	Parameters	FracRight		
	bayesclass	density=joint submodel=gauss gausstype=general	0.800418	+/-	0.00321903
	bayesclass	density=naive submodel=gauss gausstype=general	0.819745	+/-	0.00240386
	dtree	max_children=4 ne_splits=y max_pchance=0.05 adjust_chi=y max_nodes=50	0.826113	+/-	0.00327583

Learn Race from 15 attributes

Name	Model	Parameters	FracRight		
Model1	bayesclass	density=joint submodel=gauss gausstype=general	0.391303	+/-	0.00586792
Model2	bayesclass	density=naive submodel=gauss gausstype=general	0.788686	+/-	0.00560675
Model3	dtree	max_children=4 ne_splits=y max_pchance=0.05 adjust_chi=y max_nodes=50	0.860919	+/-	0.00272011

What you should know

- A lot of this should have just been a corollary of what you already knew
- Turning Gaussian DEs into Gaussian BCs
- Mixing Categorical and Real-Valued

Questions to Ponder

- Suppose you wanted to create an example dataset where a BC involving Gaussians crushed decision trees like a bug. What would you do?
- Could you combine Decision Trees and Bayes Classifiers? How? (maybe there is more than one possible way)