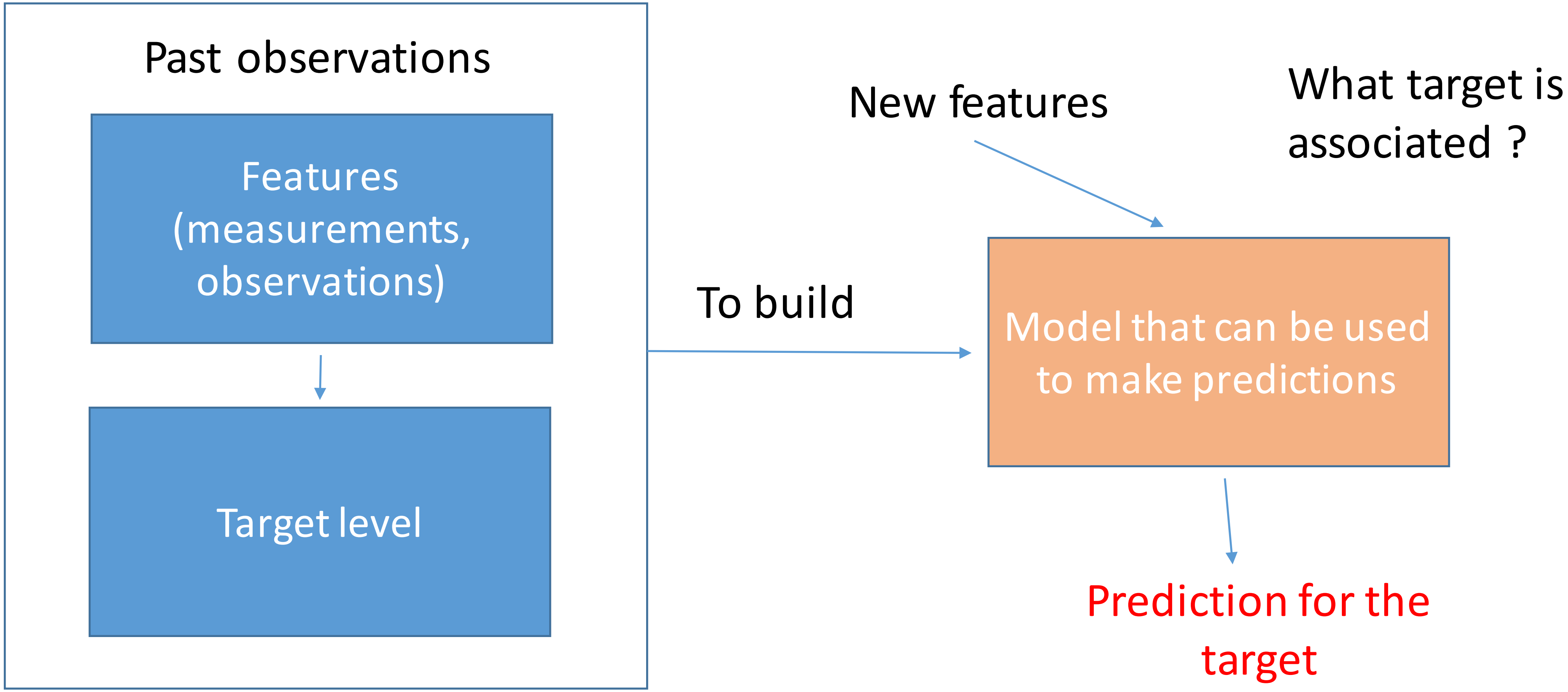
Machine learning with Bayes Naïve models

A. Hees

UCLA Machine Learning Group, May 5th 2017

Main idea



Past observations

Features (measurements, observations)

Target level

To build

Model that can be used to make predictions

New features

What target is associated ?

Prediction for the target

Rq.: in this chapter, the target is always 1D and discrete

One intuitive way to do it

The predicted “target” is the one that has the maximal probability given the features (observations) – ***MAP estimator***

$$t_{opt} = argmax\ P[t|f_1, f_2, \dots, f_n]$$

And we can use Bayes theorem

$$t_{opt} = argmax\ M(t) = argmax\ P[f_1, f_2, \dots, f_n|t]P[t]$$

The denominator does not play a role (independent of t)

$P[t]$is a prior (usually very easy to determine – count the number of occurrences in the set of past observations

Case where f_i is discrete

One can count the ratio of the number of “features” in the set of past observations versus the number of total past observations (“brute force”)

$$P[f_1, f_2, \dots, f_n | t]$$

Or use the chain rule

$$P[f_1, f_2, \dots, f_n | t] = P[f_1 | t] P[f_2 \, | t, f_1] \dots P[f_n | t, f_1, f_2, \dots,]$$

Problem: curse of dimensionality. When the number of features grow, it is difficult to have enough past observations to not end up with a 0.

One way to reduce the curse of dim. problem is to assume conditional independence

$$P[X|Y,Z] = P[X|Z] \text{ and } P[X,Y|Z] = P[X|Z]P[Y|Z]$$

Condition less strong than strict independence. This condition is usually satisfies when X and Y are produced by Z.

This simplifies the chain rules

$$P[f_1, f_2, \ldots, f_n|t]\text{=}P[f_1|t]P[f_2\ |t] \ldots P[f_n|t]$$

And leads to a specific machine learning algorithm

The Naïve Bayes Model

Use the MAP estimator under the assumption of conditional independence of all the features

$$t_{opt} = argmax\ M(t) = argmax\ P[f_1|t]\ P[f_2|t]\ \ldots P[f_n|t]P[t]$$

To "build" the model, we just need to determine the priors P[t] and all the conditional probability P[f_i | t_j] for all targets t_j.

The conditional independence of all variables is quite strong but … and the M(t) is not always very representative but the argmax is pretty insensitive to this change.

Algorithm easy to understand and code, "computationally cheap" and producing good results in most cases

The Naïve Bayes Model

Constructing the model

1) Determine the priors P[t] (counting in the dataset)

2) Determine P[f|t] (for all features and targets)
   A) If f is discrete: counting in the dataset
   B) If f is continuous: two possibilities
      i) Use a continuous probability distribution function (pdf)
         - choose the form of the pdf (use an histogram)
         - determine the parameters from the pdf
      ii) Use a binning -> come back to a discrete case

When counting, one may want to use a smoothing

The Naïve Bayes Model

Making prediction from new features:

1) Compute for all targets t_i the “score” M(t_i) (not really a probability)

$$M(t_i) = P[f_1|t_i]\,P[f_2|t_i]\,\dots P[f_n|t_i]P[t_i]$$

2) Choose the t_i with the largest score: this is the prediction !

Two possibilities when f$_i$ is continuous

1) Introduce a probability distribution function (pdf)

$$P[x < f_i < x + dx] = p_{f_i}(x)dx$$

Rq.: Which dx to use is not really defined but in our case, it does not matter because it does not impact the "argmax"

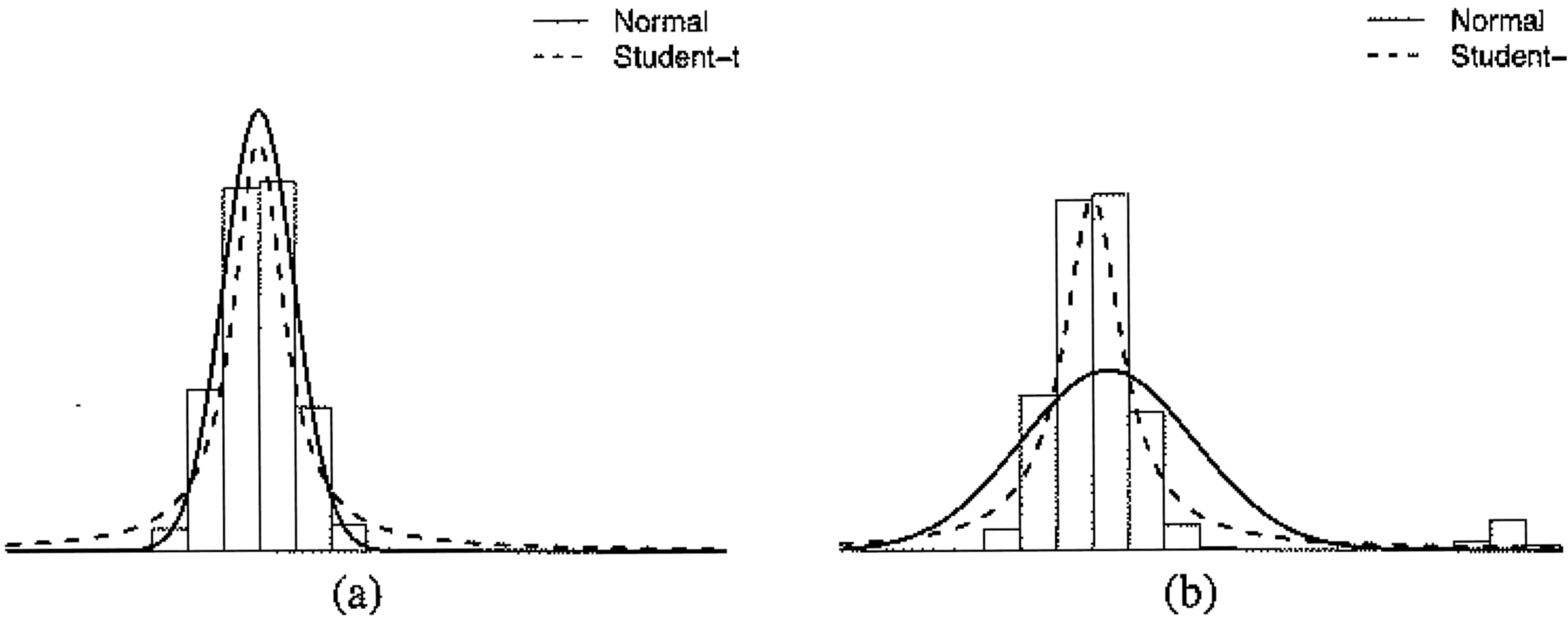
$$t_{opt} = argmax\ P[f_1|t] \dots P[f_n|t]P[t] = argmax\ p_{f_1}(x_1)dx_1 p_{f_2}(x_2)dx_2 \dots p_{f_1}(x_n)dx_n P[t]$$
$$= argmax\ p_{f_1}(x_1)p_{f_2}(x_2) \dots p_{f_1}(x_n)P[t]$$

2) Binning the continuous data

Examples of continuous pdf

1) With 1 peak:
   A) Normal pdf : light tails, very sensitive to outlier
   B) Student-t pdf: fat tails, robust to outlier

The parameters of the pdf can be determined from the mean and std

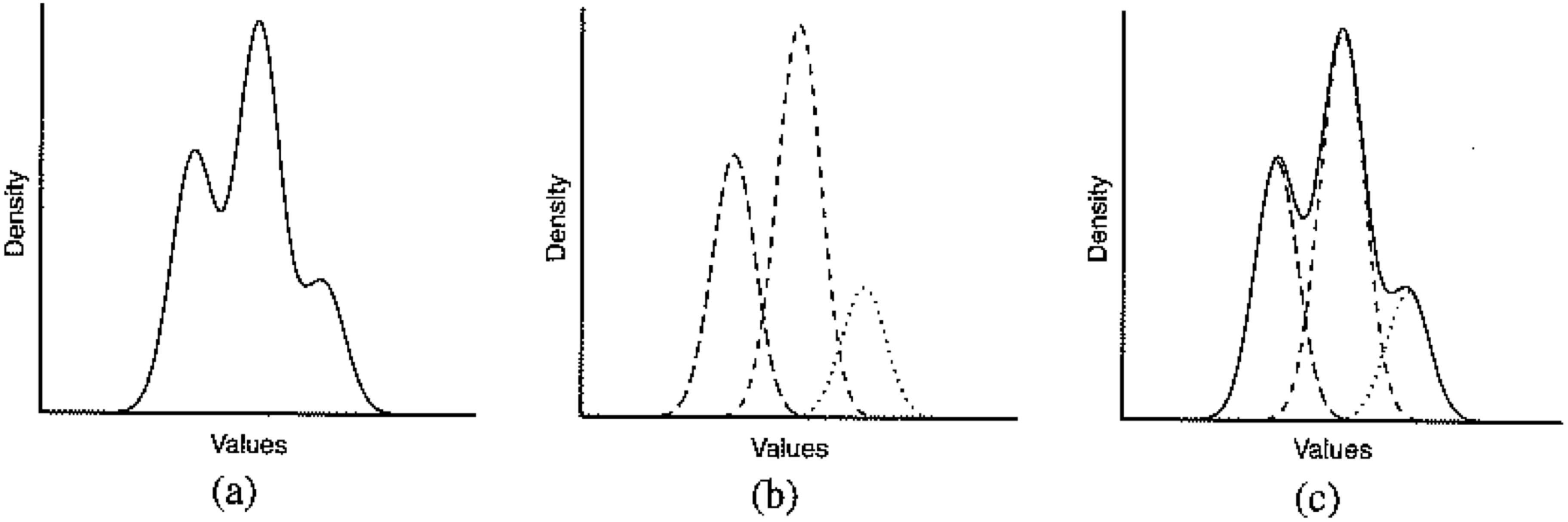


Kolmogorov-Smirnov test can help to decide between Normal and student

Examples of continuous pdf

2) With multiple peaks: (multi mode) – sum of weighted normal pdf's (with the sum of the weights =1)

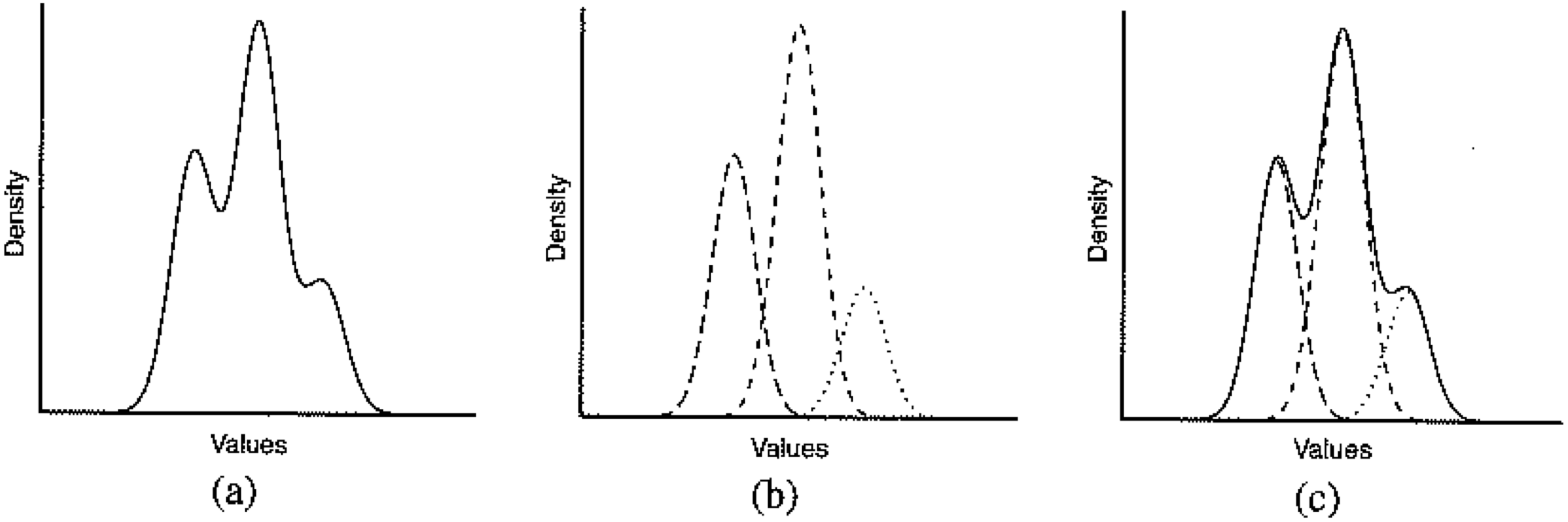
The parameters of the pdf needs to be determined by a fit (gradient algorithm)



Examples of continuous pdf

2) With multiple peaks: (multi mode) – sum of weighted normal pdf’s (with the sum of the weights =1)

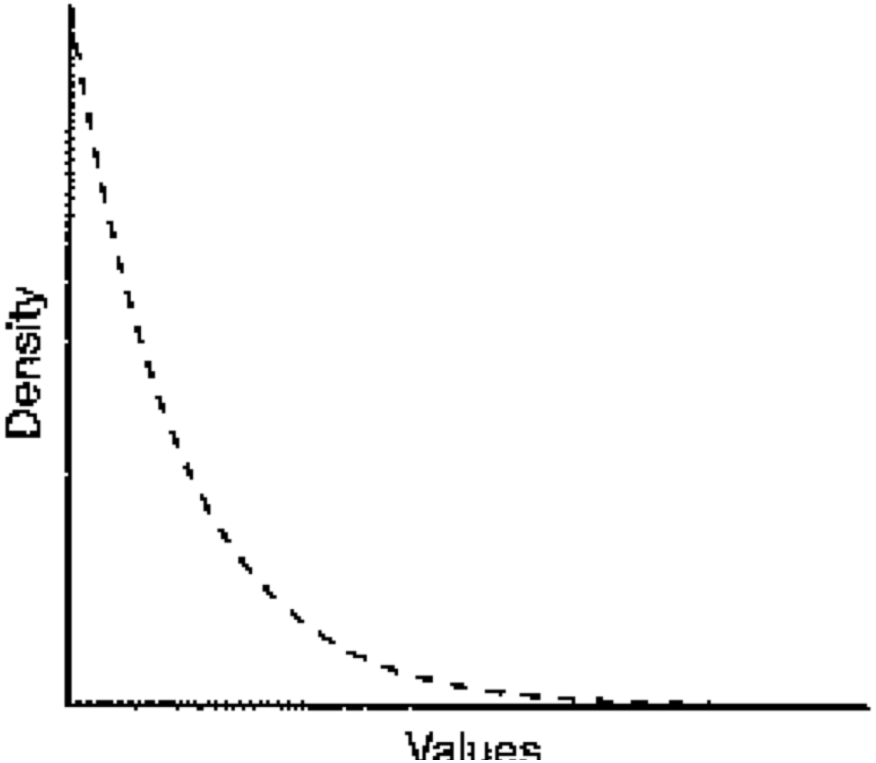
The parameters of the pdf needs to be determined by a fit (gradient algorithm)



Examples of continuous pdf

2) Exponential pdf: depends on one parameter, the rate

The parameter of the pdf can be determined as 1/mean

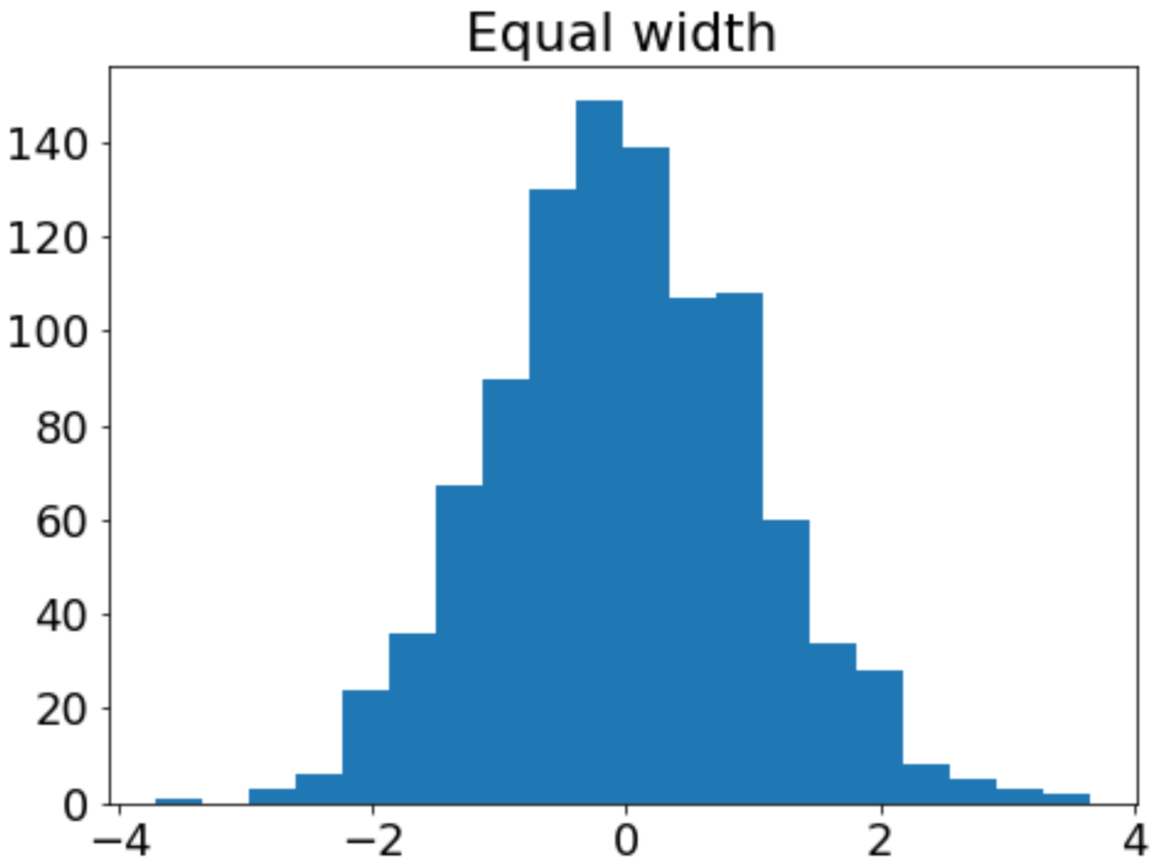


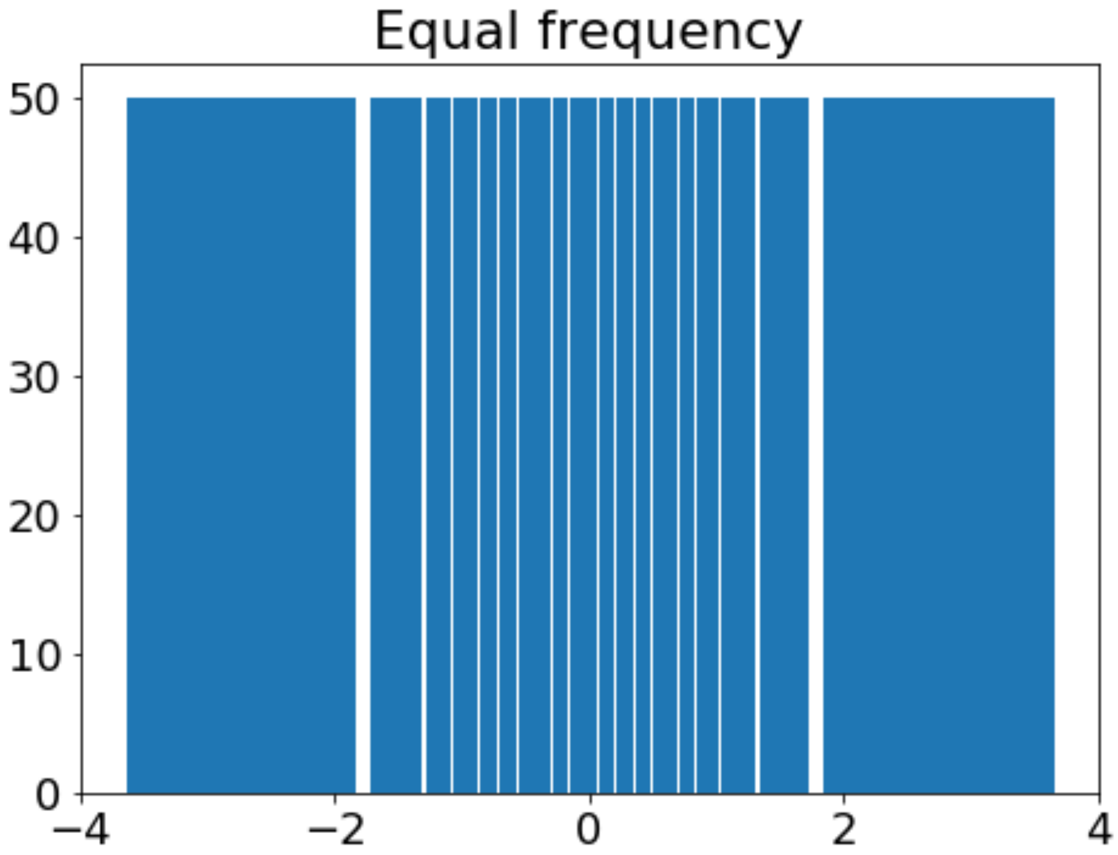
Often used to model waiting time, lifetime of components, …

Two ways to bin a continuous distribution

1) equal-width binning

2) equal-frequency binning: much better to avoid bin with low number of occurences





Like a discrete distribution: count to determine P[f|t]

The Naïve Bayes Model

Constructing the model

1) Determine the priors P[t] (counting in the dataset)

2) Determine P[f|t] (for all features and targets)
   A) If f is discrete: counting in the dataset
   B) If f is continuous: two possibilities
      i) Use a continuous probability distribution function (pdf)
         - choose the form of the pdf (use an histogram)
         - determine the parameters from the pdf
      ii) Use a binning -> come back to a discrete case

When counting, one may want to use a smoothing

The Naïve Bayes Model

Making prediction from new features:

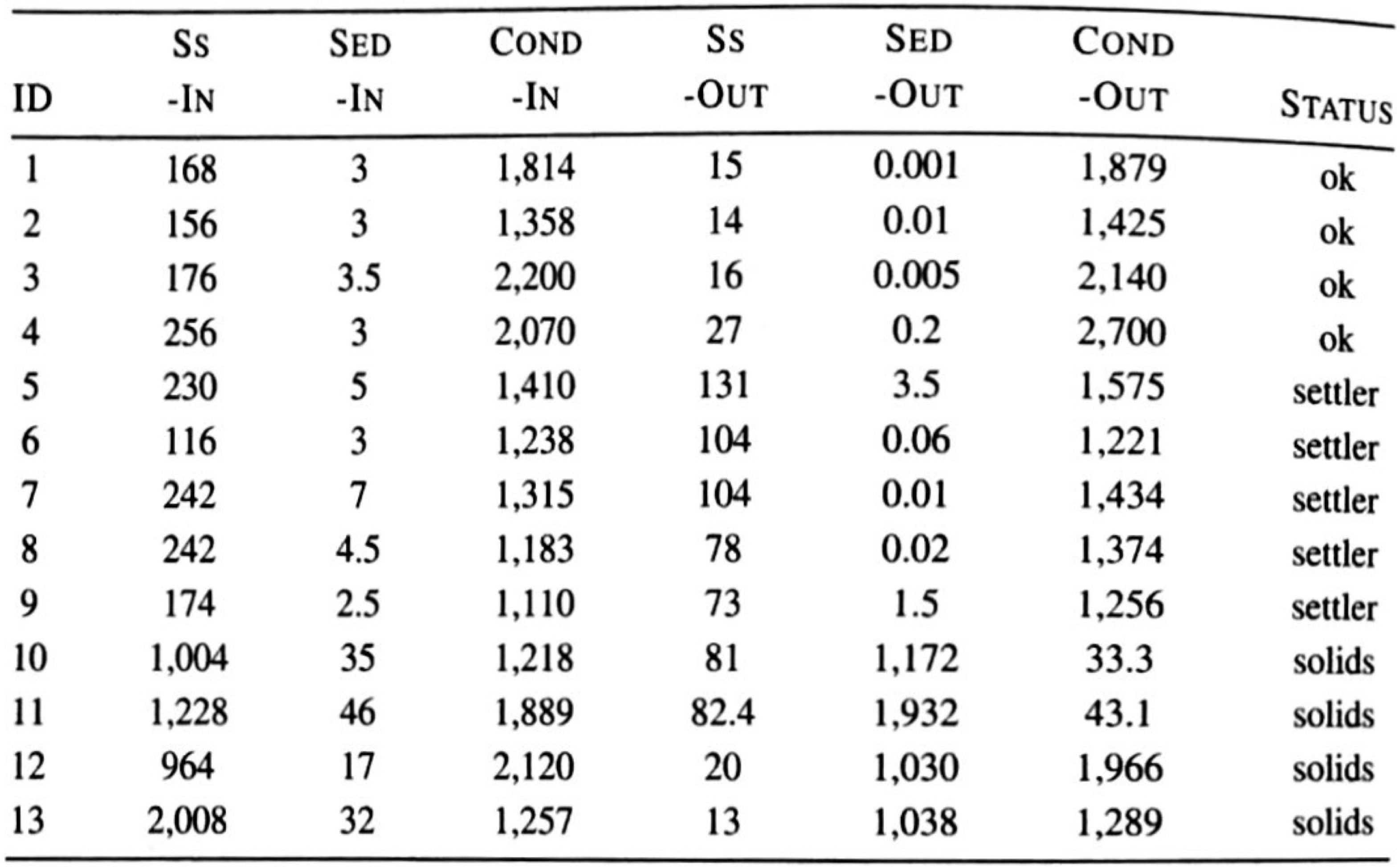
1) Compute for all targets t_i the “score” $M(t_i)$ (not really a probability)

$$M(t_i) = P[f_1|t_i]\,P[f_2|t_i]\ldots P[f_n|t_i]P[t_i]$$

2) Choose the t_i with the largest score: this is the prediction !

The exercise

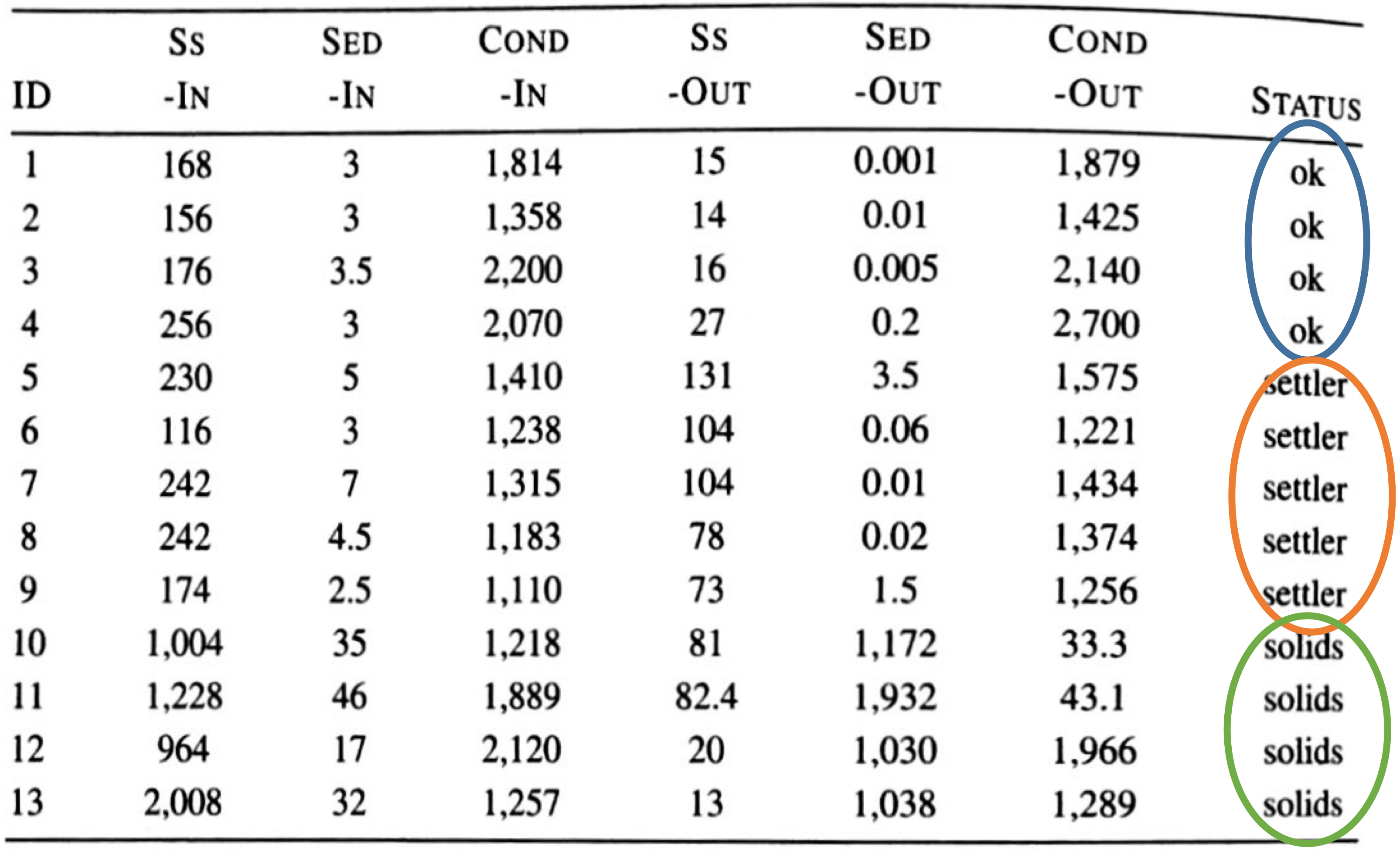
6 features, 3 targets + hypothesis that all P[f|t] are normally distributed



ID	Ss -In	Sed -In	Cond -In	Ss -Out	Sed -Out	Cond -Out	Status
1	168	3	1,814	15	0.001	1,879	ok
2	156	3	1,358	14	0.01	1,425	ok
3	176	3.5	2,200	16	0.005	2,140	ok
4	256	3	2,070	27	0.2	2,700	ok
5	230	5	1,410	131	3.5	1,575	settler
6	116	3	1,238	104	0.06	1,221	settler
7	242	7	1,315	104	0.01	1,434	settler
8	242	4.5	1,183	78	0.02	1,374	settler
9	174	2.5	1,110	73	1.5	1,256	settler
10	1,004	35	1,218	81	1,172	33.3	solids
11	1,228	46	1,889	82.4	1,932	43.1	solids
12	964	17	2,120	20	1,030	1,966	solids
13	2,008	32	1,257	13	1,038	1,289	solids

The exercise: the prior

6 features, 3 targets + hypothesis that all P[f|t] are normally distributed



ID	SS -IN	SED -IN	COND -IN	SS -OUT	SED -OUT	COND -OUT	STATUS
1	168	3	1,814	15	0.001	1,879	ok
2	156	3	1,358	14	0.01	1,425	ok
3	176	3.5	2,200	16	0.005	2,140	ok
4	256	3	2,070	27	0.2	2,700	ok
5	230	5	1,410	131	3.5	1,575	settler
6	116	3	1,238	104	0.06	1,221	settler
7	242	7	1,315	104	0.01	1,434	settler
8	242	4.5	1,183	78	0.02	1,374	settler
9	174	2.5	1,110	73	1.5	1,256	settler
10	1,004	35	1,218	81	1,172	33.3	solids
11	1,228	46	1,889	82.4	1,932	43.1	solids
12	964	17	2,120	20	1,030	1,966	solids
13	2,008	32	1,257	13	1,038	1,289	solids

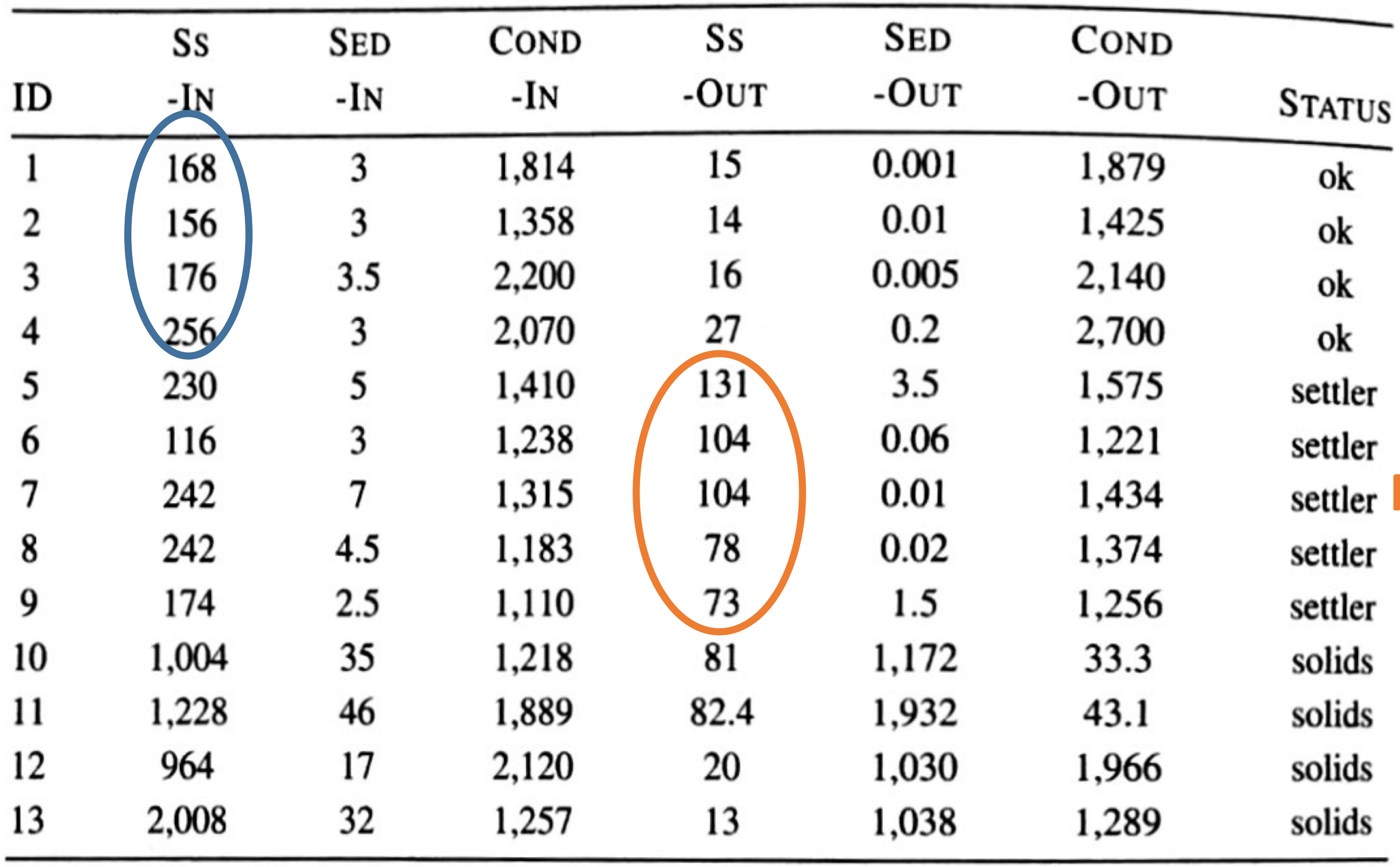
P[ok]=4/13

P[set]=5/13

P[sol]=4/13

The exercise: the conditional probabilities

6 features, 3 targets + hypothesis that all P[f|t] are normally distributed



	Ss	Sed	Cond	Ss	Sed	Cond	
ID	-In	-In	-In	-Out	-Out	-Out	Status
1	168	3	1,814	15	0.001	1,879	ok
2	156	3	1,358	14	0.01	1,425	ok
3	176	3.5	2,200	16	0.005	2,140	ok
4	256	3	2,070	27	0.2	2,700	ok
5	230	5	1,410	131	3.5	1,575	settler
6	116	3	1,238	104	0.06	1,221	settler
7	242	7	1,315	104	0.01	1,434	settler
8	242	4.5	1,183	78	0.02	1,374	settler
9	174	2.5	1,110	73	1.5	1,256	settler
10	1,004	35	1,218	81	1,172	33.3	solids
11	1,228	46	1,889	82.4	1,932	43.1	solids
12	964	17	2,120	20	1,030	1,966	solids
13	2,008	32	1,257	13	1,038	1,289	solids

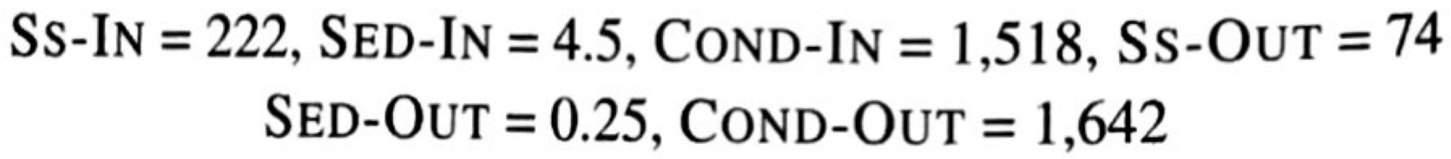
P[SS-IN | OK] ~ N (189, 45.42)

P[ss-out | set]~N(98,23.38)

Etc… 18 terms

The exercise: the prediction

We observe



Ss-In = 222, Sed-In = 4.5, Cond-In = 1,518, Ss-Out = 74
Sed-Out = 0.25, Cond-Out = 1,642

For each target, we need to compute the “score”

$$M(t_i) = P[f_1|t_i]\ P[f_2|t_i] \ldots P[f_n|t_i]P[t_i]$$

Example

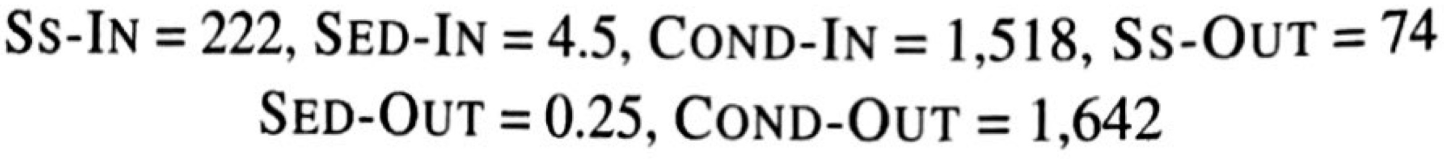
$$M(OK) = P[SS - in = 222|OK] \ldots P[C - OUT = 1642|OK]P[OK]$$

P[ss-in | set]~N(189,45.42)

P[ok]=4/13

The exercise: the prediction

We observe



SS-IN = 222, SED-IN = 4.5, COND-IN = 1,518, SS-OUT = 74
SED-OUT = 0.25, COND-OUT = 1,642

For each target, we need to compute the “score”

$$M(t_i) = P[f_1|t_i]\; P[f_2|t_i] \ldots P[f_n|t_i]P[t_i]$$

Example

M[ok]=3.4 E-36

M[settler]=1.5 E-13

M[solids]=1. E-21