A Short Note on the Application of Rule Induction Methods to Flow Properties of Object Based Models

Y. Xie and C. V. Deutsch, University of Alberta
(ylxie@civil.ualberta.ca and cdeutsch@civil.ualberta.ca)

A. Stan Cullick, Mobil Oil Corporation
(A_stan_cullick@mobil.com)

Abstract

This short note presents an application of a rule induction algorithm to an object-based model example. Our purpose is to show how rule induction algorithms extract understandable rules from geological measurements. The data used in this example was generated by object modeling. Fluvial sand channels and shale are assigned to the models with distinct permeability values. The effective permeabilities are evaluated with flow simulation program. We know that the effective permeability of the models are determined by the geometric parameters used for the generation of the models, such as width, height, orientation, and sinuosity of the channels. The rule induction algorithm retrieves rules from the database that match our understanding of flow processes.

The generation of the training data is illustrated in detail, the methodology is breifly reviewed and the results are intepretated. Programs used for this example are documented.

Introduction

The application of data mining techniques has grown in recent years to deal with large databases. The main focus of data mining is to derive predictive models, rules and relationships from large databases, i.e., to extract knowledge from data. There are different types of mathematical modeling techniques involved in data mining. Most statistical and numerical techniques, including neural networks, are not rule based and the knowledge extracted from data is expressed as the parameters of statistical or mathematical models. Rule based techniques present knowledge in natural language with semantics and syntax understandable by people. Therefore, they provide not only predictive models, but also insight into the physics of the underlying phenomenon.

Recently, we proposed a new algorithm of rule induction for geological data and the theory of the proposed algorithm was described in detail in the companion reports [1, 2]. There are many situations in the petroleum industry where rule induction could be applied. In this report, a synthetic data set is used to illustrate the methodology and demonstrate the feasibility of the proposed algorithm.

In fluvial or deepwater depositional settings, the sandy facies occur as sand channels with associated levee and crevasse deposits. We only consider sand-filled channels embedded within a matrix of shale. The corresponding effective permeability depends on the orientation, sinuosity, width, thickness and geometry of the channels. The relationship between these geometric features and the effective permeability are *rules* to be extracted for future use. For this purpose, channel facies models are created by with a variety of channel parameters such as orientation, thickness, and sinuosity, which are called condition attributes for the rule induction system. For a given set of parameters, multiple realizations of channel facies models are generated. Flow simulation is conducted for each generated model and the effective permeability in the X direction is taken as a measure of quality (decision class) of the model. Rule induction is carried out in the constructed training set. The performance of the proposed algorithm is illustrated.

Method Summary

The method of the rule induction algorithm is described in detail in the companion reports, and will be only described conceptually here.

In the proposed rule induction algorithm, we are searching for a full set of rules that comply with all possible configurations of condition attributes. Every observation in the training set leads to a potential rule; however, there are challenges to establish the relative importance and reliablility of rules. To identify significance of the rules, we have introduced concepts like *coverage, accuracy* and *significance*. Important rules have high *accuracy*, that is predictive ability, and high *coverage*, that is, they are applicable in many situations. For simplicity, accuracy and coverage are combined into a single measure of *significance*. Second, a training data set may not cover the entire set of configurations; some condition attribute configuration may not have observations. Future observations, however, may happen to have that configuration. Therefore, blanks in the full rule set must be filled in. It is reasonable that the blanks in the full rule table would be filled in by rules generated from subsets of data variable configurations. To proceed in this direction, an *information* measure has been defined to evaluate the importance of subsets of condition attributes. Third, for some rule induction problem, the distinguishablility between decision classes may be questionable. The similarity among decision classes could be measured by the same information measure.

The rule induction procedure begins by enumerating all possible configurations of condition attributes and counting the observations of each condition configuration and decision class. The number of instances of each configuration received is called *coverage*, which determines the reliability of rules derived from such a configuration. Conditional probability of each decision class for given condition configuration is then calculated. The original data table is then reduced into a configuration table. The conditional probabilities are transformed into *accuracy* measures by scaling them into a range between -1 and +1. *Significance* is then calculated based on em accuracy and *coverage*. Rules are sorted by significance.

Data Generation for Channel Systems

The data set consists of the geometric features of channel sands and the associated effective permeability. The object modeling program `fluvsim` was used to generate multiple facies models. For simplicity, but without loss of generatlity, only channel sand and shale facies were considered. The simulation domain is a 100 by 100 two-dimensional horizontal section. All parameters of `fluvsim` are fixed except orientation (O), sinuosity (S), and width (W) (width/thickness ratio). Three triangular distributions are set for each of these three adjustable parameters representing *low*, *expected*, and *high* values. There are 27 combinations (sets) of parameters. For each set of parameters, the actual parameter values were randomly drawn and 50 realizations generated. Thus, a total of 1350 different facies models were created. For each facies model, permeability values of 100 md and 1 md were assigned to channel sand and shale, respectively. A flow simulation was conducted in the X direction with no-flow boundaries, using the `flowsim`. The effective permeability in the X direction was taken as a measure of the quality or productivity of each facies model.

A C shell script is written to automate the process. Figure 1 illustrates the flow-chart of the shell script and Figure 2 and Figure 3 show the code of the shell script.

The training data were organized as a data table with 1350 rows, each representing one facies model, and 4 columns, where the first three denotes the values of three condition attributes, i.e., the orientation, sinuosity and width of the channels, and the fourth column was the value of the decision attribute, i.e., the effective permeability in X direction.

Figure 4, Figure 5 and Figure 6 show one of 50 facies models for each of 27 parameter sets. The composite histogram of the 1350 realizations and the histograms of effective permeability of the 27 parameter sets are shown in Figures 7, 8, 9, and 10, respectively.

No	Cond. Att.			Basic Statistics		Effective permeability		
	O	S	W	mean K	Std. K	0 (≤ 4)	1(4-14)	2 (≥ 14)
1	0	0	0	3.90	1.53	32 (64%)	18 (36%)	0 (0%)
2	0	0	1	2.22	0.48	49 (98%)	1 (2%)	0 (0%)
3	0	0	2	1.91	0.31	50(100%)	0 (0%)	0 (0%)
4	0	1	0	8.50	2.57	4 (8%)	46 (92%)	0 (0%)
5	0	1	1	5.14	2.53	23 (46%)	26 (52%)	1 (2%)
6	0	1	2	4.29	2.87	34 (68%)	16 (32%)	0 (0%)
7	0	2	0	9.40	2.30	1 (2%)	49 (98%)	0 (0%)
8	0	2	1	9.69	4.26	3 (6%)	39 (78%)	8 (16%)
9	0	2	2	8.16	4.75	14 (28%)	30 (60%)	6 (12%)
10	1	0	0	8.47	2.75	2 (4%)	47 (94%)	1 (2%)
11	1	0	1	6.94	5.05	23 (46%)	21 (42%)	6 (12%)
12	1	0	2	4.55	4.01	34 (68%)	12 (24%)	4 (8%)
13	1	1	0	8.21	1.95	0 (0%)	50 (100%)	0 (0%)
14	1	1	1	7.88	3.28	6 (12%)	42 (84%)	2 (4%)
15	1	1	2	5.28	3.49	25 (50%)	23 (46%)	2 (4%)
16	1	2	0	5.02	1.00	5 (10%)	45 (90%)	0 (0%)
17	1	2	1	5.66	1.76	6 (12%)	44 (88%)	0 (0%)
18	1	2	2	5.97	2.52	15 (30%)	35 (70%)	0 (0%)
19	2	0	0	17.46	2.17	0 (0%)	2 (4%)	48 (96%)
20	2	0	1	18.74	4.05	0 (0%)	9 (18%)	41 (82%)
21	2	0	2	21.19	5.80	0 (0%)	6 (12%)	44 (88%)
22	2	1	0	8.66	1.44	0 (0%)	50 (100%)	0 (0%)
23	2	1	1	8.54	2.64	3 (6%)	46 (92%)	1 (2%)
24	2	1	2	9.08	4.17	8 (16%)	34 (68%)	8 (16%)
25	2	2	0	3.97	0.77	29 (58%)	21 (42%)	0 (0%)
26	2	2	1	3.66	1.06	35 (70%)	15 (30%)	0 (0%)
27	2	2	2	3.23	0.85	42 (84%)	8 (16%)	0 (0%)

Table 1: Parameters (codes) for `fluvsim` and effective permeability values

Discretization of Continuous Attributes

The rule induction method requires that the data be "binned" into categorical variables. It is reasonable and straightforward to classify the three geometrical parameters into three categories: columns 2 to 4 of Table 1 list the codes of the parameters used in `fluvsim`. The three categorical values of *orientation* are 2 (good: azimuth angle of 0 ± 10, which is aligned with the X direction), 1 (medium: azimuth angle of 45 ± 10) and 0 (bad: Y azimuth angle of 90 ± 10, which is perpendicular to the X direction). The three categorical values for *sinuosity* are 0 (low: low deviation 10 ± 5 and long length of 50 ± 5 units), 1 (medium: medium deviation 20 ± 5 and medium length of 35 ± 5 units), 2 (high: high deviation 30 ± 5 and short length of 20 ± 5 units). The three categorical values for *width* are 2 (large: 25 ± 5), 1 (medium: 15 ± 5) and 0 (small: 5-10).

The method of "binning" the permeability data affects the final rule induction results. For simplicity, three categories were determined for the effective permeability by inspecting the permeability histograms. Summary statistics of the effective permeability for the 50 realizations of each configuration are listed in columns 5 and 6 of Table 1. The three categories determined are: 2 (high: $k \geq 14$), 1 (medium: $4 < k < 14$) and 0 (low: $k \leq 4$). The number of instances and proportions of the observations of each of the three decision categories for each configuration are tabulated in the last three columns of Table 1.

By inspecting Table 1, we can expect certain rules like those listed in Table 2.

Good O (2)	and	low S (0)	and	No matter W	leads to	high K (2)
Bad O (0)	and	low S (0)	and	No matter W	leads to	low K (0)
Good O (2)	and	high S (2)	and	No matter W	leads to	low K (0)
Medium O (1)					mostly leads to	medium K (1)

Table 2: The expected rules

Note that the "*No matter W*" in Table 2 indicates that the width can have any of the categorical values; thus, width is found not to be that important in determining outcomes.

Result of Rule Induction

The algorithm `ruleind` was applied to the 1350 by 4 synthesized data table. Figure 11 shows the parameter file used for the rule induction. The coverage, relative coverage, and conditional probabilities for each decision category, accuracy and significance were calculated. Table 3 lists the configurations of condition attributes, the occurrence frequency C_j^o, and the significance $s_{o|j}$ of decision category o when given configuration j. By construction, all 27 possible configurations of condition attributes in this example have equal, non-zero sample coverage. This will not be the case for most data systems.

The full set of rules is sorted by significance for each decision category, which is listed in Table 4. The positive rules shown in Table 5 are taken from the top portion of Table 4, which have the highest significance values.

The negative rules as tabulated in Table 6 results are taken from the bottom portion of Table 4, which have the lowest significance values.

The rule set is used to estimate the permeability class of training objects and table 7 lists the accuracy rate of the classification. Even though this accuracy rate is not a good measure to evaluate the modeling, it still can provide some indication of the prediction model.

For this synthetic data set, the training data covers all configurations by construction. Therefore, the rule set is complete from the set with all condition attributes and there are no blanks in the rule sets. Figure 12 shows the information value of all 7 subsets of this synthesized data set and the full set with all three condition attributes has the highest information value. In the situation of the full set does not cover all configurations of condition attributes, such information values will be used to rank the subsets for retrieving corresponding rules from the subsets. Figure 13 shows the information value change when decision classes are lumped pair-wise for the full condition attribute set. The yellow color in diagonal elements set the basis for comparison which correspond to a situation without decision-class lumping. Off-diagonal elements show the change in the information values when the attributes are lumped. Colors warmer than yellow indicate an increase in the value of information and those cooler colors denote a decrease in the information value.

Discussion and Future Work

The proposed algorithm and significance measure work well for this simple example. The most important positive and negative rules are retrieved successfully. The proposed significance definition combines both measures of accuracy and coverage and serves as a quality measure of the rules. Also, the significance identifies positive and negative rules, similar to the positive region and negative region in rough sets.

The proposed rule induction technique is suited to geological data where it is assumed that most attributes are significant. The proposed significance measure can be used in combination with other

No	Cond. Att.			Occurrence $C_{j,o}$			Significance $S_{o\|j}$		
	O	S	W	Low K (0)	Med. K (1)	High K (2)	Low K (0)	Med. K (1)	High K (2)
1	0	0	0	32	18	0	.640	.000	-1.000
2	0	0	1	49	1	0	.980	-.980	-1.000
3	0	0	2	50	0	0	1.000	-1.000	-1.000
4	0	1	0	4	46	0	.000	.920	-1.000
5	0	1	1	23	26	1	.000	.000	-.980
6	0	1	2	34	16	0	.680	.000	-1.000
7	0	2	0	1	49	0	-.980	.980	-1.000
8	0	2	1	3	39	8	.000	.780	.000
9	0	2	2	14	30	6	.000	.600	.000
10	1	0	0	2	47	1	.000	.940	-.980
11	1	0	1	23	21	6	.000	.000	.000
12	1	0	2	34	12	4	.680	.000	.000
13	1	1	0	0	50	0	-1.000	1.000	-1.000
14	1	1	1	6	42	2	.000	.840	.000
15	1	1	2	25	23	2	.500	.000	.000
16	1	2	0	5	45	0	.000	.900	-1.000
17	1	2	1	6	44	0	.000	.880	-1.000
18	1	2	2	15	35	0	.000	.700	-1.000
19	2	0	0	0	2	48	-1.000	.000	.960
20	2	0	1	0	9	41	-1.000	.000	.820
21	2	0	2	0	6	44	-1.000	.000	.880
22	2	1	0	0	50	0	-1.000	1.000	-1.000
23	2	1	1	3	46	1	.000	.920	-.980
24	2	1	2	8	34	8	.000	.680	.000
25	2	2	0	29	21	0	.580	.000	-1.000
26	2	2	1	35	15	0	.700	.000	-1.000
27	2	2	2	42	8	0	.840	.000	-1.000

Table 3: Potential rules for `fluvsim` data set

No	Cond. Att.			Dec. Value	$S_{o\|j}$	No	Cond. Att.			Dec. Value	$S_{o\|j}$
	O	S	W				O	S	W		
19	2	0	0	2	.2623	27	2	2	2	1	-.5200
21	2	0	2	2	.2098	11	1	0	1	2	-.6400
20	2	0	1	2	.1740	14	1	1	1	0	-.6400
3	0	0	2	0	.1129	9	0	2	2	2	-.6400
2	0	0	1	0	.1073	17	1	2	1	0	-.6400
27	2	2	2	0	.0721	21	2	0	2	1	-.6400
22	2	1	0	1	.0680	16	1	2	0	0	-.7000
13	1	1	0	1	.0680	4	0	1	0	0	-.7600
7	0	2	0	1	.0647	12	1	0	2	2	-.7600
10	1	0	0	1	.0582	23	2	1	1	0	-.8200
23	2	1	1	1	.0551	8	0	2	1	0	-.8200
4	0	1	0	1	.0551	15	1	1	2	2	-.8800
16	1	2	0	1	.0520	19	2	0	0	1	-.8800
17	1	2	1	1	.0491	10	1	0	0	0	-.8800
26	2	2	1	0	.0435	14	1	1	1	2	-.8800
14	1	1	1	1	.0434	7	0	2	0	0	-.9400
12	1	0	2	0	.0399	5	0	1	1	2	-.9400
6	0	1	2	0	.0399	10	1	0	0	2	-.9400
8	0	2	1	1	.0356	2	0	0	1	1	-.9400
1	0	0	0	0	.0332	23	2	1	1	2	-.9400
18	1	2	2	1	.0262	25	2	2	0	2	-1.000
25	2	2	0	0	.0242	3	0	0	2	2	-1.000
24	2	1	2	1	.0241	26	2	2	1	2	-1.000
9	0	2	2	1	.0163	21	2	0	2	0	-1.000
15	1	1	2	0	.0141	7	0	2	0	2	-1.000
5	0	1	1	1	.0099	16	1	2	0	2	-1.000
11	1	0	1	0	.0099	13	1	1	0	2	-1.000
5	0	1	1	0	.0099	13	1	1	0	0	-1.000
15	1	1	2	1	.0059	17	1	2	1	2	-1.000
25	2	2	0	1	.0037	19	2	0	0	0	-1.000
11	1	0	1	1	.0037	4	0	1	0	2	-1.000
1	0	0	0	1	.0010	2	0	0	1	2	-1.000
6	0	1	2	1	-.0400	1	0	0	0	2	-1.000
18	1	2	2	0	-.1000	22	2	1	0	2	-1.000
26	2	2	1	1	-.1000	20	2	0	1	0	-1.000
9	0	2	2	0	-.1600	18	1	2	2	2	-1.000
12	1	0	2	1	-.2800	6	0	1	2	2	-1.000
20	2	0	1	1	-.4600	27	2	2	2	2	-1.000
24	2	1	2	2	-.5200	3	0	0	2	1	-1.000
8	0	2	1	2	-.5200	22	2	1	0	0	-1.000
24	2	1	2	0	-.5200						

Table 4: Full set of rules sorted according to significance value

Good O (2)	and	low S (0)	and	No matter W	leads to	high K (2)
Bad O (0)	and	low S (0)	and	large W (2)	leads to	low K (0)
Good O (2)	and	high S (2)	and	medium W (1)	leads to	low K (0)
Good O (2)	and	high S (2)	and	large W (2)	leads to	low K (0)
Good O (2)	and	medium S (1)	and	no matter W	leads to	medium K (1)
Medium O (1)	and	no matter S	and	small W (0)	leads to	medium K (1)
Bad O (0)	and	high S (2)	and	small W (0)	leads to	medium K (1)

Table 5: The derived positive rules

Good O (2)	and	medium S (1)	and	small W (0)	Never leads to	low K (0)
Bad O (0)	and	small S (0)	and	large W (2)	Never leads to	low K (1)
Good O (2)	and	high S (2)	and	large W (2)	Never leads to	high K (2)
Bad O (0)	and	medium S (1)	and	large W (2)	Never leads to	high K (2)
Medium O (1)	and	high S (2)	and	large W (2)	Never leads to	high K (2)
Good O (2)	and	small S (0)	and	medium W (1)	Never leads to	low K (0)

Table 6: The derived negative rules

	1	2	3	Accuracy
1	353	90	0	79.68%
2	135	583	17	79.32%
3	12	27	133	77.33%

Table 7: Results of classification

rule induction techniques and serves as a ranking measure to single out the most important rules. In general, however, the algorithm will need to be extended to include attribute reduction.

The significance lies between -1 to + 1. In principle, the potential rules with largest absolute significance values will be chosen as important rules, but there exists ambiguity of chosing a significance threshold.

References

[1] A. Stan Cullick, Y. L. Xie, and C. V. Deutsch. Rule induction algorithm for application to geology and petrophysical data: methodology. *Mobil Strategic Research Center (SRC) memorandum*, December 1999.

[2] A. Stan Cullick, Y. L. Xie, and C. V. Deutsch. Rule induction algorithm for application to geology and petrophysical data: facies assignment from wire line logs. *Mobil Strategic Research Center (SRC) memorandum*, December 1999.

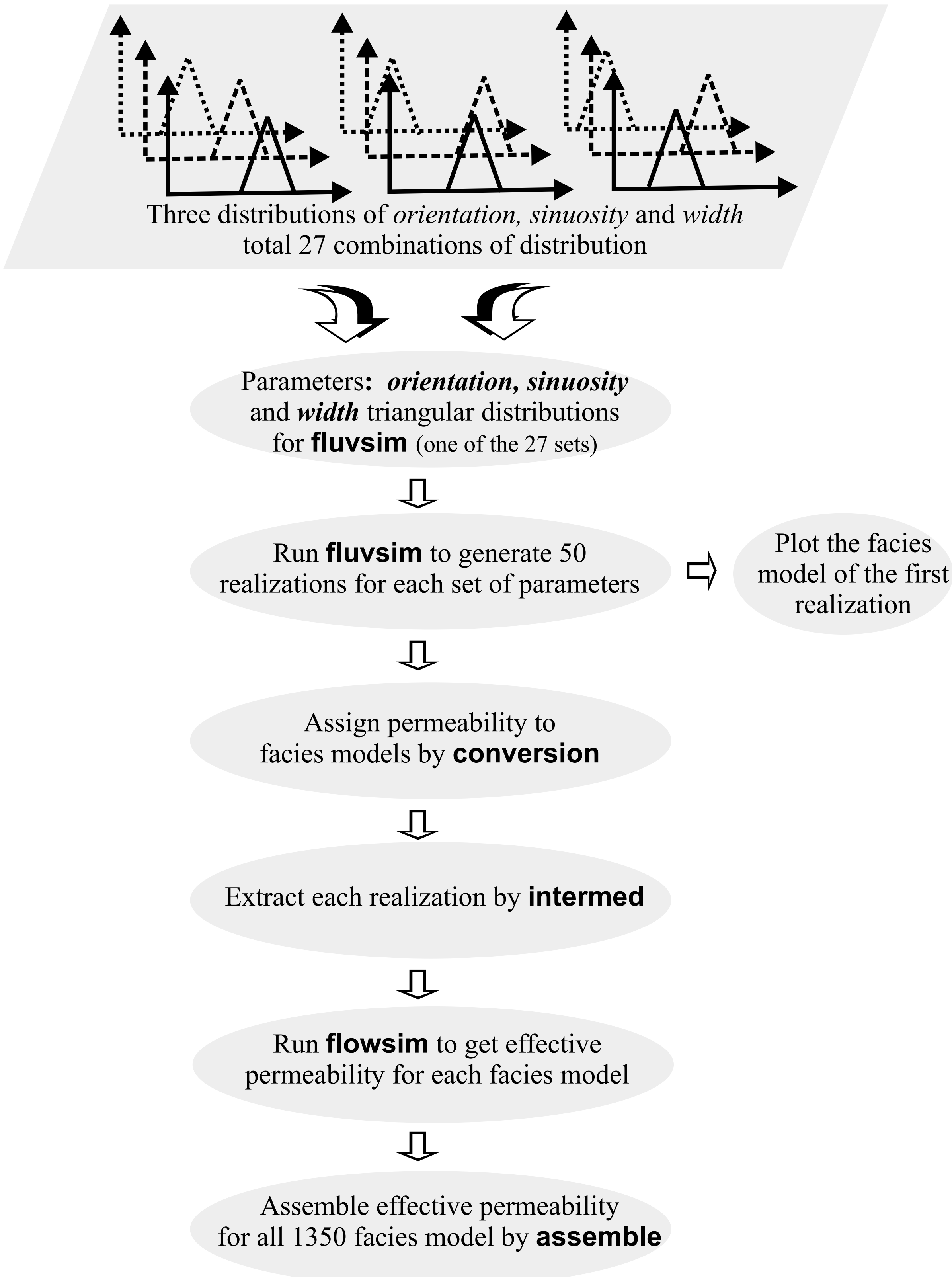

Figure 1: Flow chart of data generation process

Shell script for data generation

```
#!/bin/csh
##
@ orien = 1
@ no_orien = 3
@ no_realization = 50
@ batch = 0
while ( $orien <= $no_orien )
@ sinu = 1
        @ no_sinu = 3
        if ( $orien == 1 ) then
                @ orien1 = -10; @ orien2 = 0; @ orien3 = 10
        endif
        if ( $orien == 2 ) then
                @ orien1 = 35; @ orien2 = 45; @ orien3 = 55
        endif
        if ( $orien == 3 ) then
                @ orien1 = 80; @ orien2 = 90; @ orien3 = 100
        endif

        while ( $sinu <= $no_sinu )
                @ width = 1
                @ no_width =3
                if ( $sinu == 1 ) then
                        @ sinu_dev1 = 5;   @ sinu_dev2 = 10; @ sinu_dev3 = 15
                        @ sinu_len1 = 45;  @ sinu_len2 = 50;  @ sinu_len3 = 55
                endif
                if ( $sinu == 2 ) then
                        @ sinu_dev1 = 15; @ sinu_dev2 = 20; @ sinu_dev3 = 25
                        @ sinu_len1 = 30;  @ sinu_len2 = 35;  @ sinu_len3 = 40
                endif
                if ( $sinu == 3 ) then
                        @ sinu_dev1 = 25; @ sinu_dev2 = 30; @ sinu_dev3 = 35
                        @ sinu_len1 = 15;  @ sinu_len2 = 20;  @ sinu_len3 = 25
                endif

                while ( $width <= $no_width )
                        if ( $width == 1) then
                                @ width1 = 5; @ width2 = 5; @ width3 = 10
                        endif
                        if ( $width == 2 ) then
                                @ width1 = 10; @ width2 = 15; @ width3 = 20
                        endif
                        if ( $width == 3) then
                                @ width1 = 20; @ width2 = 25; @ width3 = 30
                        endif
                        #       ==================================================
                        #       for ech batch get a new parameter file for fluvsim
                        #       ==================================================
                        sed -e "s/*orien1/$orien1/g"   -e "s/*orien2/$orien2/g"   -e "s/*orien3/$orien3/g"   fluvsim.tmp > int1;
                        sed -e "s/*sinud1/$sinu_dev1/g" -e "s/*sinud2/$sinu_dev2/g" -e "s/*sinud3/$sinu_dev3/g" int1 > int2;
                        sed -e "s/*sinul1/$sinu_len1/g" -e "s/*sinul2/$sinu_len2/g" -e "s/*sinul3/$sinu_len3/g" int2 > int3;
                        sed -e "s/*width1/$width1/g"   -e "s/*width2/$width2/g"   -e "s/*width3/$width3/g"   int3 > int4;

                        @ width = $width + 1
                        @ batch = $batch + 1

                        echo " batch = " $batch
                        sed -e "s/\#/$batch/g" int4 > fluvsim.par;
```

Figure 2: Shell code for data generation process

Shell script for data generation (continue)

```
rm int1
rm int2
rm int3
rm int4

echo " ****************************************"
echo " parameter file is ready for batch " $batch
echo " Fluvsim is RUNNING......"
run-fluvsim

# ===================================================
#           for each batch, we plot the first realization
# ===================================================
sed -e "s/\#/$batch/g" pixelplt.tmp > pixelplt.par
run-pixelplt

# ===================================================
#           for each batch, we have got facies model by running fluvsim
#           and then permeability is assigned for facies by conversion
 ===================================================
echo " convert the facies model into permeability values "
sed -e "s/\#/$batch/g" conversion.tmp > conversion.par;

echo "**************************"
run-conversion


#===================================================
#           then we want to run intermed to extract each individual
#           realization for each batch to get the effective permeability
# ===================================================

@ realization = 1
while ( $realization <= $no_realization )
          sed -e "s/\*/$batch/g" -e "s/\@/$realization/g" intermed.tmp > intermed.par

          echo "**************************"
          echo " Intermed is RUNNING ......."
          run-intermed

          echo "**************************"
          echo "Flowsim is running ......."
          run-flowsim

          echo "**************************"
          echo "ASSEMBLE is RUNNING........"
          assemble
          mv temp.out kapeff.out
          @ realization = $realization + 1
end
      end
      @ sinu = $sinu + 1
   end
   @ orien = $orien + 1
End
```

Figure 3: Shell code for data generation process(Continuation)

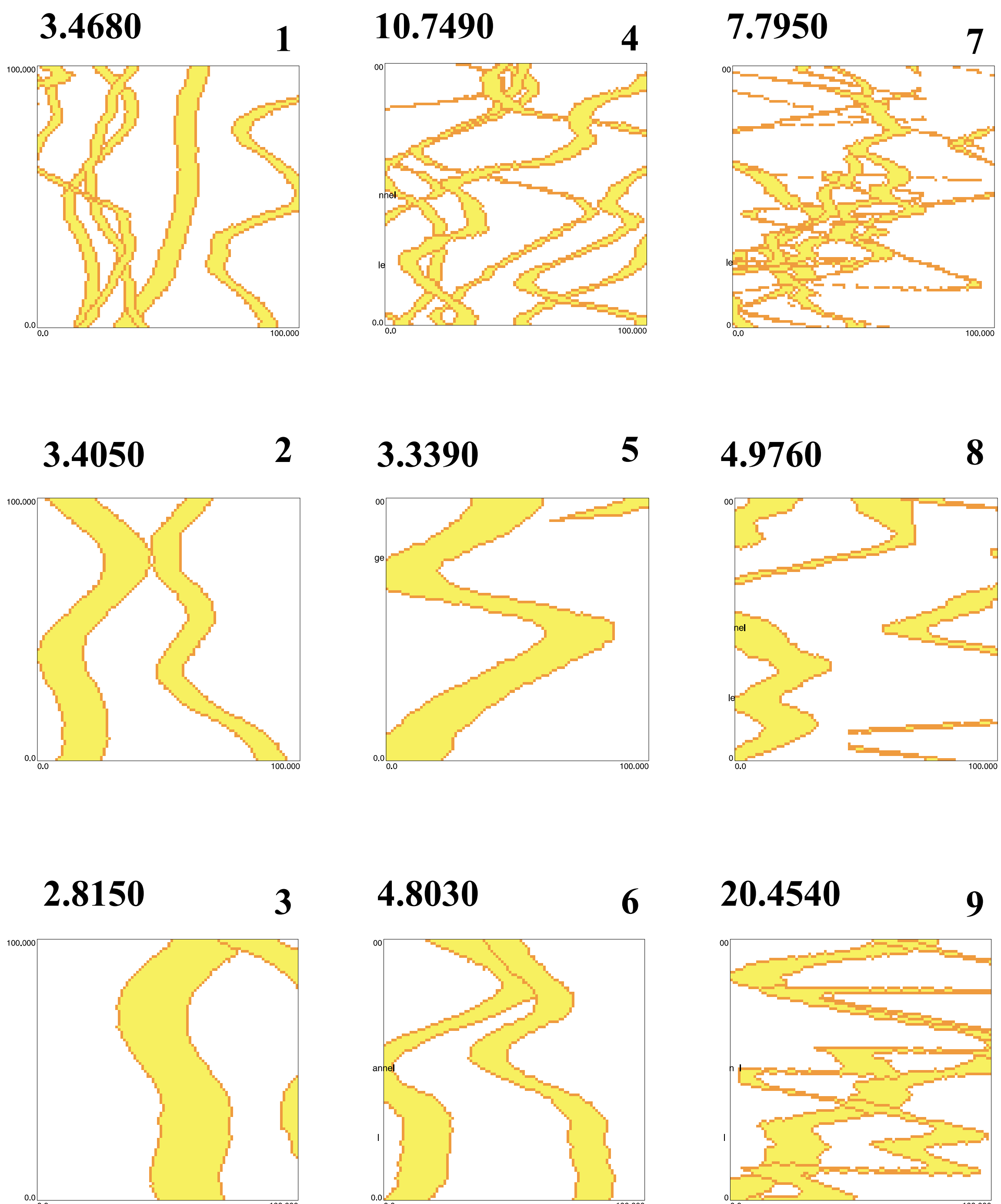

Figure 4: Facies model of realization 1 for parameter distribution sets 1 to 9. The value on the left top of each plot is the mean effective permeability of 50 realizations; the number on the right top denotes the index of parameter distribution set

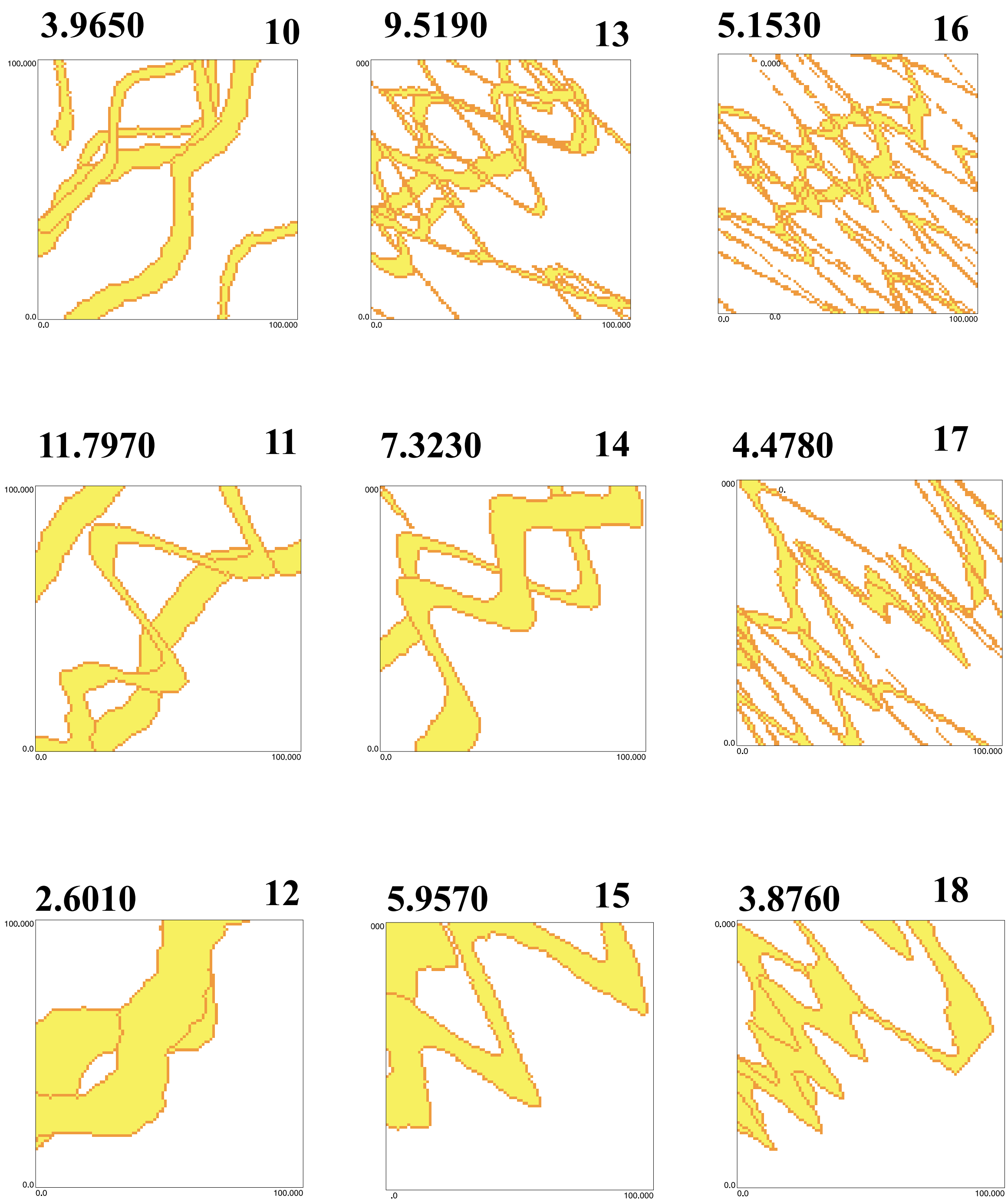

Figure 5: Facies model of realization 1 for parameter distribution sets 10 to 18. The value on the left top of each plot is the mean effective permeability of 50 realizations; the number on the right top denotes the index of parameter distribution set

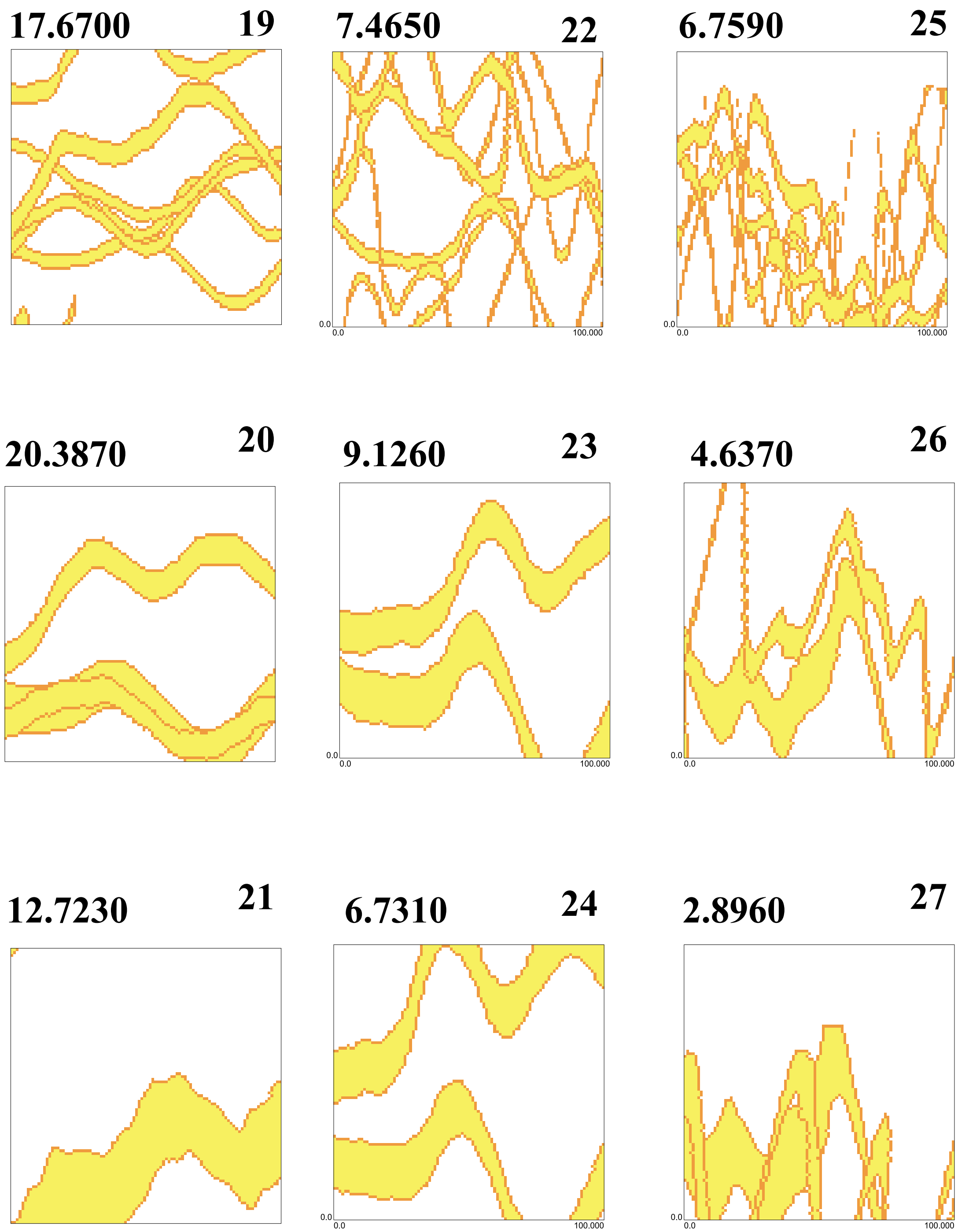

Figure 6: Facies model of realization 1 for parameter distribution sets 19 to 27. The value on the left top of each plot is the mean effective permeability of 50 realizations; the number on the right top denotes the index of parameter distribution set

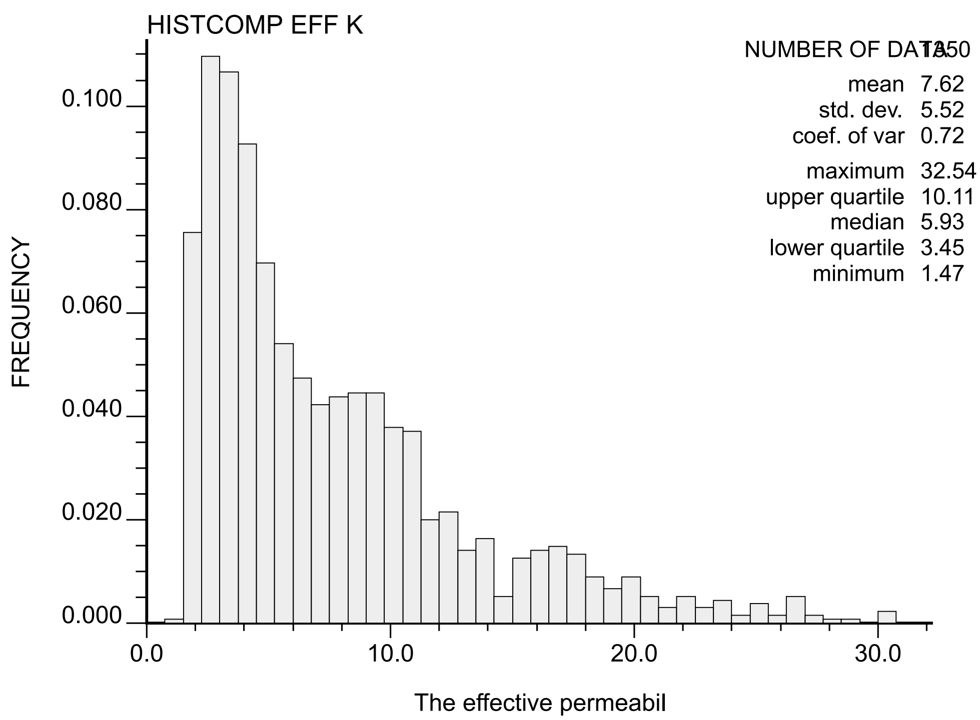

Figure 7: Composite histogram of effective permeability of overall 1350 facies models

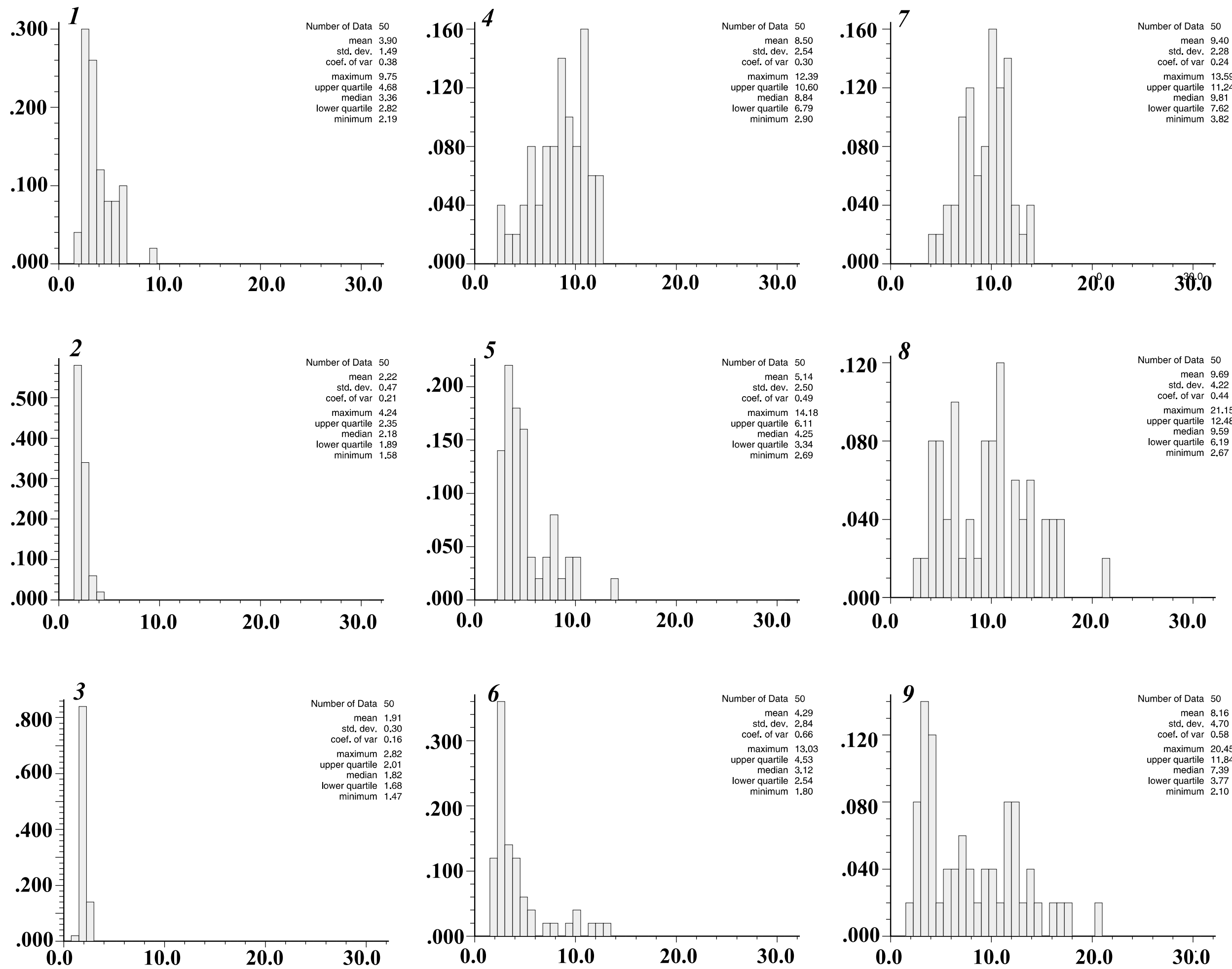

Figure 8: Histograms of effective permeability of 50 realizations for parameter distribution sets 1 to 9

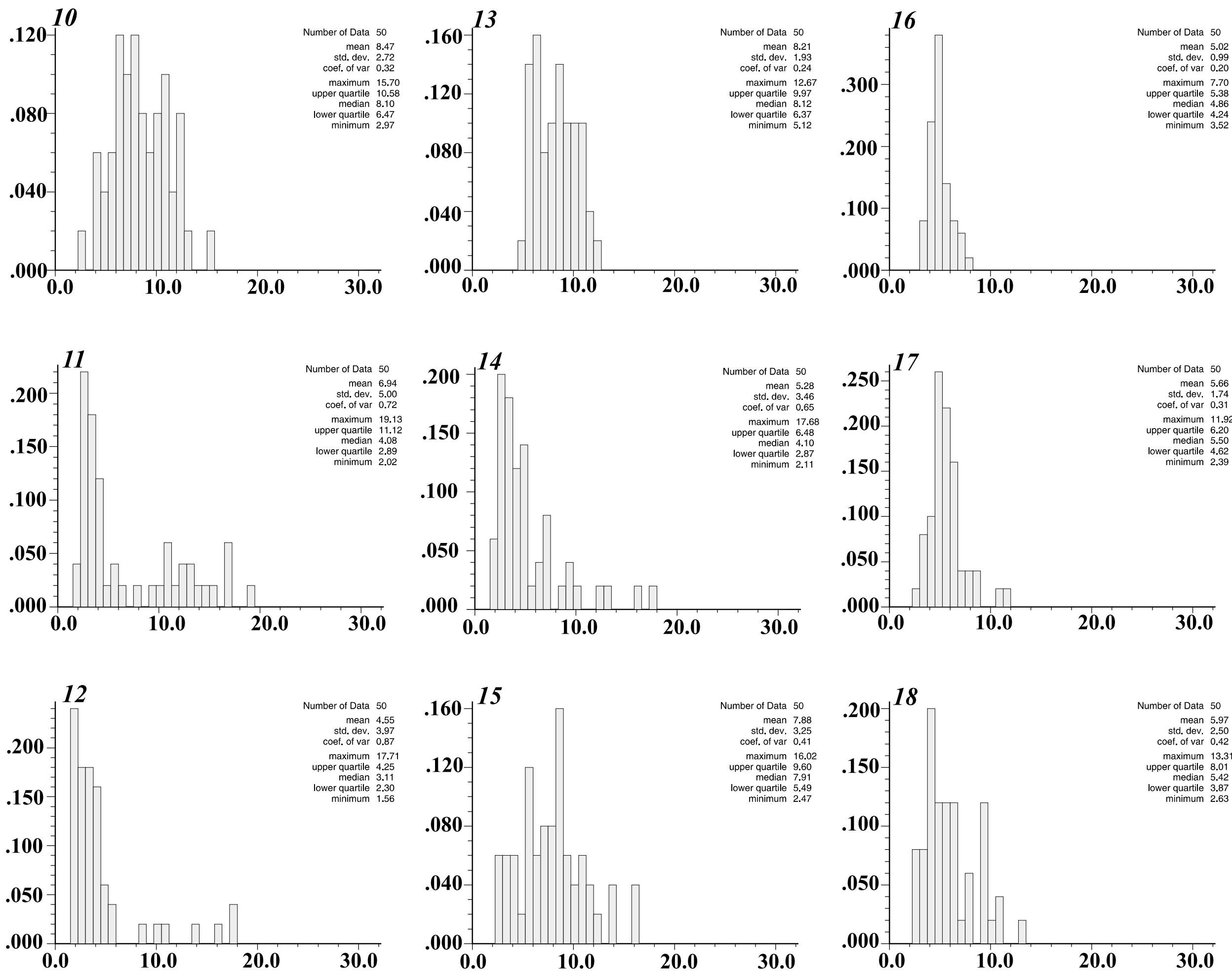

Figure 9: Histograms of effective permeability of 50 realizations for parameter distribution sets 10 to 18

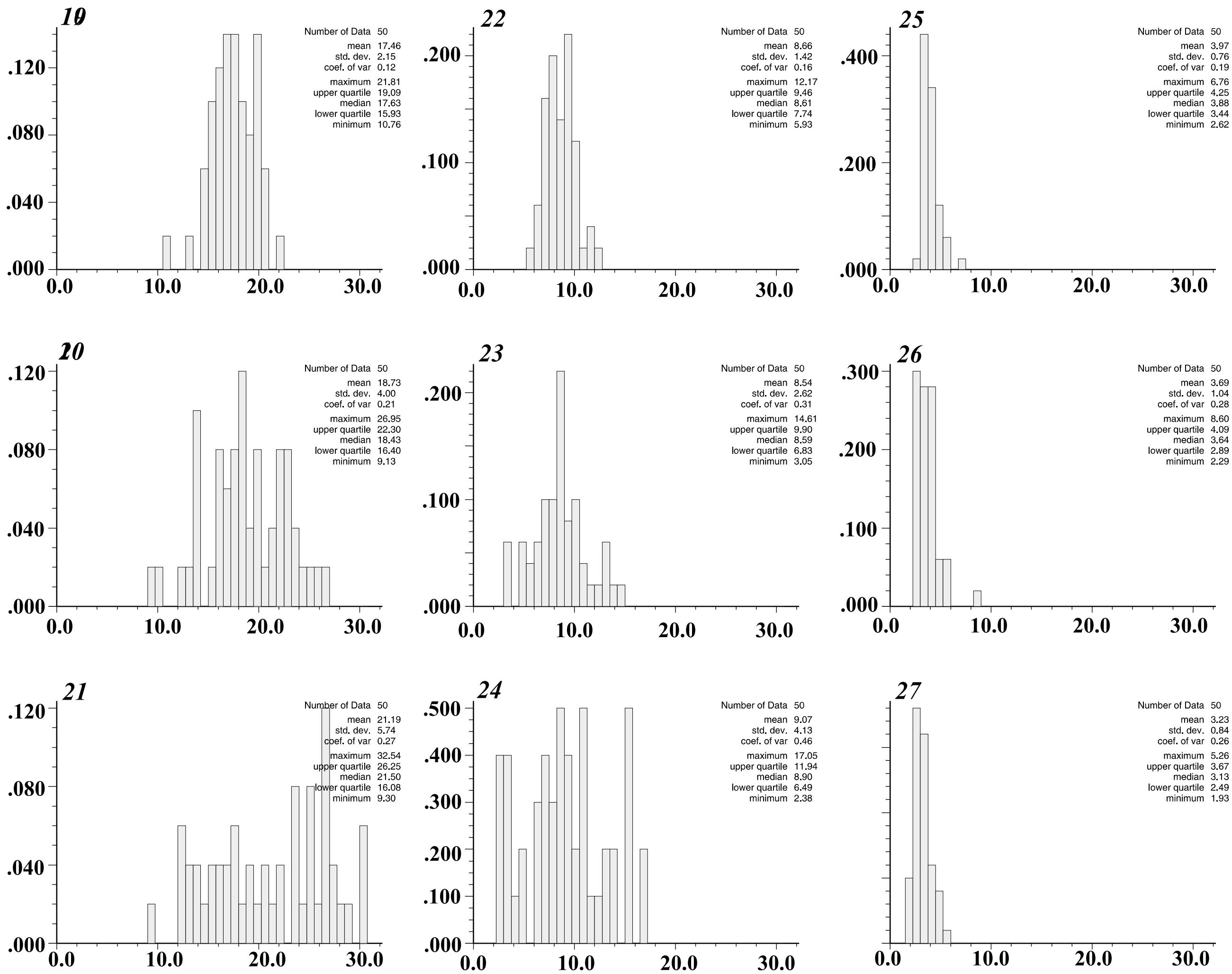

Figure 10: Histograms of effective permeability of 50 realizations for parameter distribution sets 19 to 27

```
                  Parameters for ruleind
                  **********************

START OF PARAMETERS:
fluvsim.sys                                 - input data file for rule induction
3                                           - no. of cond. attr.
1,2,3                                       - cols. of cond. attr.
3,3,3                                       - no. of levels of cond. attrs.
0,1,2                                       - levels of cond. attr. 1
0,1,2                                       - levels of cond. attr. 2
0,1,2                                       - levels of cond. attr. 3
1                                           - no. of decision attr.
4                                           - col of decision attr.
3                                           - no. of levels of decision attr.
0,1,2                                       - levels of decision attr.
fluv                                        - name of project
```

Figure 11: Parameter file of `ruleind`

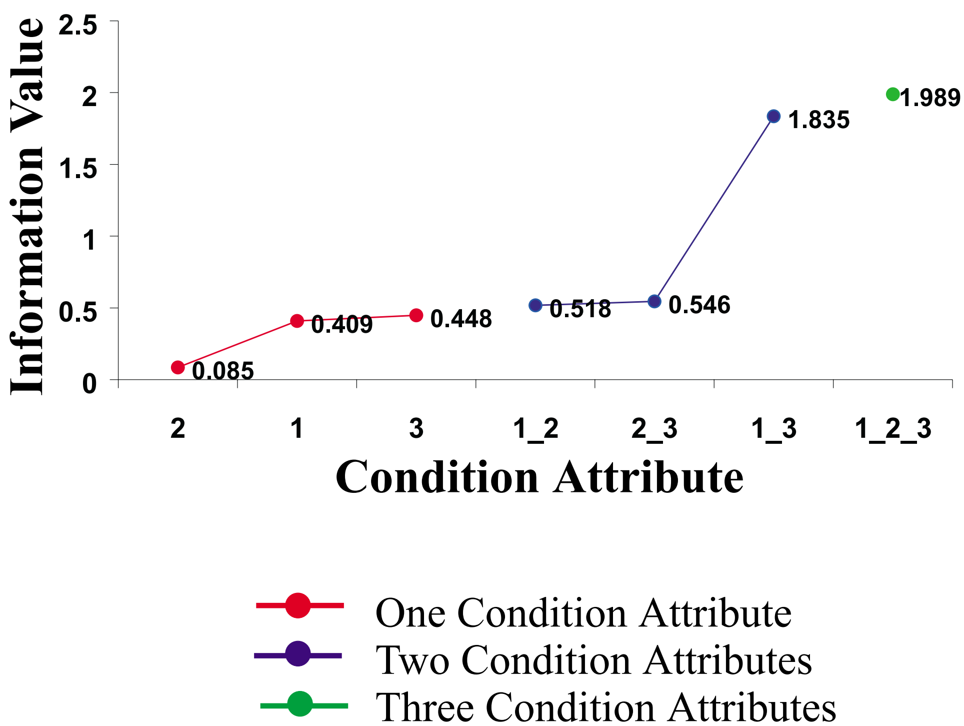

Figure 12: Information values for all subsets of condition attributes

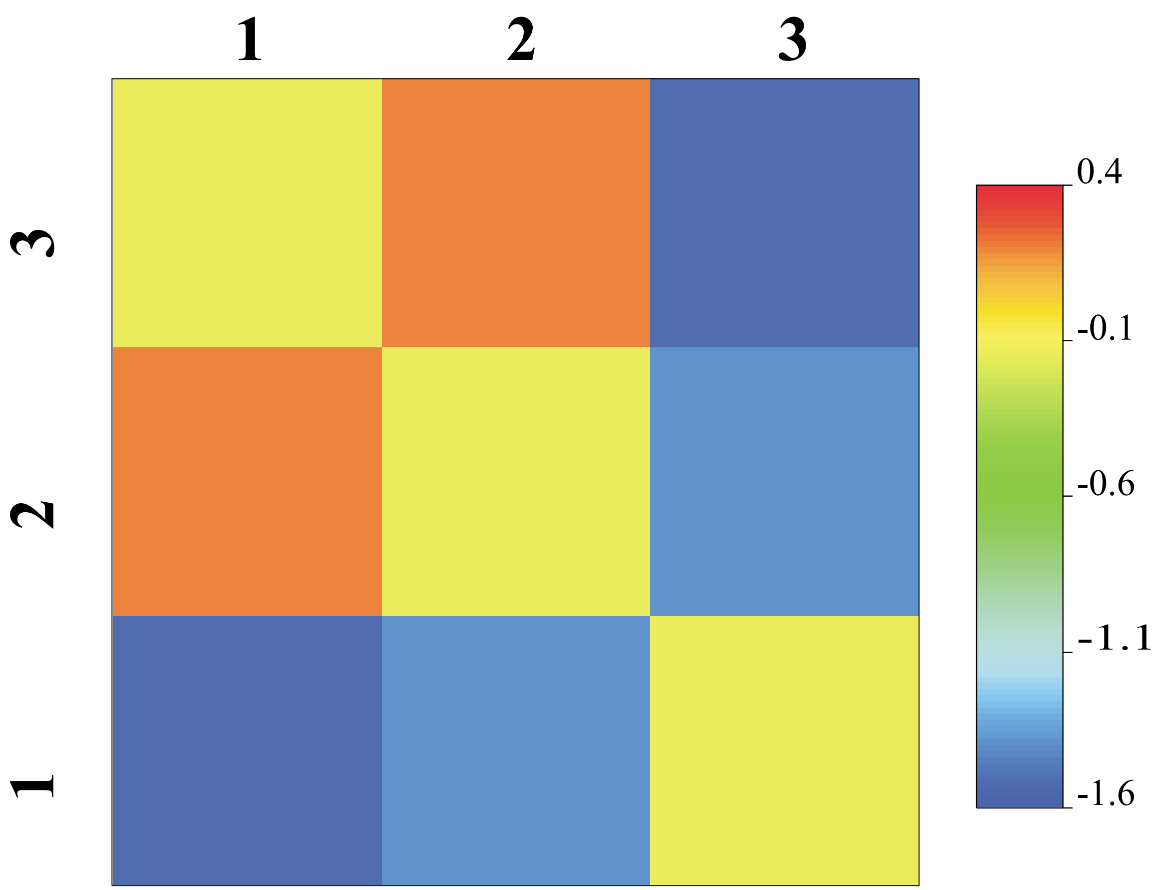

Figure 13: Changes in the information values when decision classes lumped pairwisely