International Conference of Agricultural Engineering
AgEng 2014 Zurich 6–10 July

Ref: C0276

Automatic body condition scoring - improvements using fuzzy logic

Roii Spoliansky, Dept. of Industrial Engineering and Management, Ben-Gurion University of the Negev

Yael Edan, Dept. of Industrial Engineering and Management, Ben-Gurion University of the Negev

Yisrael Parmet, Dept. of Industrial Engineering and Management, Ben-Gurion University of the Negev

Uzi Moallem, Agricultural Research Organization

Aaron Antler, Agricultural Research Organization

Ilan Halachmi, Agricultural Research Organization

Abstract

Developing an automatic body condition scoring (BCS) system is an important management tool for dairy farms, currently it is determined manually. In this research, a classification system using fuzzy logic was built to improve classification of existing computer vision algorithms. The existing computer vision system matches a body condition score to a set of Fourier descriptors, which represent the outline of the cow. Data of 151 dairy cows were collected in the ARO research farm in Bet-Dagan during 2011-2012. Model performance was evaluated by comparing BCS references given by an expert, using the average and median error of the model's evaluation and the number of correctly classified cows. The model was developed using statistical and visual analysis of the data to derive the number and type of variables, membership functions, rules and optimization methods, which will be presented in the paper. Minimizing the objective function, with suitable constraints, was performed in Matlab's constrained nonlinear multivariable minimization algorithm. The fuzzy logic classification systems resulted in significant improvement in the training data as compared to a system with simple rules, although some cases resulted with decreased performance for the testing data. The best system resulted in 51.6% correct classification for the testing data set, with an average error of 0.389 and median error of 0.288, as compared to previous results, which resulted in 46.8% correct classification (average error – 0.343; median error – 0.315). Dividing the data into 4 groups according to BCS shows the algorithm ability to cope with extreme values. For obese cows with BCS classes of 4-5, the classifier detected 100% of the images correctly compared to 0% in previous studies. For cows with BCS classes between 3 and 4, the classifier detected 19% compared to 33.3%. For BCS classes between 2 and 3, 62.2% were correctly classified compared to 56.8% (with non-fuzzy logic classification), and for cows with low BCS (classes 1-2) both algorithms detected 100%. Classification of BCS using fuzzy logic has proved itself as an effective method and as less sensitive to extreme values.

Keywords: Cow BCS, Fuzzy-logic, Computer-vision

1. Introduction

Body Condition Scoring (BCS) is an important management tool in dairy farming. BCS evaluates the cow's energy reserves in a 5 point scale, where 1 is an emaciated cow and 5 is an obese cow (Ferguson et al., 1994). A cow's Body Condition Score (BCS) and the change in a cow's BCS were found to be a good forecasting tool for problems with the cow's health, production and reproduction (Wildman et al., 1982). BCS is currently performed manually by an experienced farmer using tactile or visual methods; the farmer has no computerized decision support tools to help with his decisions in the farm (Hady et al., 1994). A manually performed BCS requires experienced personnel, it is time consuming and the result is a subjective measure that cannot be automatically collaborated in a computer report (Halachmi et al., 2008). Several attempts have been made to automate the BCS process, but so far the process is still manual. Past research had promoted the automatic BCS system a great deal, especially in the direction of a computer vision based system (Azzaro et al., 2011, Bercovich, 2012, Bewley et al., 2008, Halachmi et al., 2008). Most of the systems developed still require some level of manual classification and others are not fully operational (Bercovich, 2012). A common way of solving classification problems when the output is vaguely defined is Fuzzy Logic, which is used in many control systems and classifiers (Yu& Fowler, 1994). The main objective of this study is to further develop an automatic BCS by applying Fuzzy Logic to overcome the insufficient system classification.

2. Materials and methods

A fuzzy logic algorithm was developed to improve classification. The research was based on an existing algorithm (Bercovich, 2012), based on data collected at the ARO research farm at Bet Dagan between October 2011 and February, 2012. The classifier was developed using Matlab R2012a and the appropriate toolboxes, Fuzzy Logic Toolbox and Optimization Toolbox. The number and type of the variables, the rules and parameters were determined using statistical and visual analysis of the data and optimized using heuristics and optimization algorithms.

2.1 Data

2.2.1 Raw data

The raw data in this research were 289 images that were taken in previous studies (Bercovich, 2012) with BCS defined manually by an expert for each image. The images were taken at the ARO research farm at Bet Dagan between October 2011 and February, 2012. A Nikon D 7000 DSLR camera was placed on entrance to a milking parlor at 2.5 meter height. All images were taken before noon milking. Each cow that entered through the parlor gate activated the camera, and six consecutive images were acquired with a resolution of 1632X2464. All images were downloaded to the PC and processed off-line.

2.2.2 Data analysis

The database was split in to three main parts:

- X - The input matrix includes all Fourier descriptors. The images were classified according to this data.
- Y – The output vector includes the real BCS that was evaluated by trained personnel.
- $\hat{y}$– The predicted output includes the basic algorithm's output and the output of this research classification.

In order to better understand the data before starting to build the fuzzy classification model graphical and mathematical tools such as cross correlation matrix, scatter plots, observing the max and min values of each input value and their impact on the output value and more were applied.

2.2 Classifier modeling

According to the information gathered in the data analysis phase, the fuzzy classifier's properties were derived. The properties determined were: the number and identity of the inputs, the input membership functions, the output membership functions, the fuzzy rules, the weights of the fuzzy rules, the Logical operations types and the defuzzification type. In Table 1, all the points defining the membership functions (3 membership functions for each variable – low, mid and high) for input values of the initial system. Table 2 shows the points that define the membership functions (5 membership functions – emaciated, thin, average, fat and obese) of the BCS output value for the initial system. Table 3 shows the initial rules for the fuzzy classification system constructed and their weights.

Table 1 –Values of input membership functions for initial system

	Low 1	**Low 2**	**Low 3**	**Low 4**	**Mid 1**	**Mid 2**	**Mid 3**	**Mid 4**	**High 1**	**High 2**	**High 3**	**High 4**
Fft 1	-45	-5	5	45	5	45	55	95	55	95	105	145
Fft 2	-22.5	-2.5	2.5	22.5	2.5	22.5	27.5	47.5	27.5	47.5	52.5	72.5
Fft 3	-22.5	-2.5	2.5	22.5	2.5	22.5	27.5	47.5	27.5	47.5	52.5	72.5
Fft 4	-13.5	-1.5	1.5	13.5	1.5	13.5	16.5	28.5	16.5	28.5	31.5	43.5
Fft 5	-9	-1	1	9	1	9	11	19	11	19	21	29
Fft 6	-6.75	-0.75	0.75	6.75	0.75	6.75	8.25	14.25	8.25	14.25	15.75	21.75

Table 2 - Values of output membership functions for initial system

	Point 1	**Point 2**	**Point 3**	**Point 4**
emaciated	0.1	0.9	1.1	1.9
thin	1.1	1.9	2.1	2.9
average	2.1	2.9	3.1	3.9
fat	3.1	3.9	4.1	4.9
obese	4.1	4.9	5.1	5.9

Table 3–Rules of the initial fuzzy classification system

Rule	**If Fft**	**Is**	**Then BCS is**	**Weight**	Rule	**If Fft**	**Is**	**Then BCS is**	**weight**
1	1	Low	obese	0.027	**10**	4	Low	obese	0.081
2	1	Mid	average	0.054	**11**	4	Mid	average	0.162
3	1	High	emaciated	0.027	**12**	4	High	emaciated	0.081
4	2	Low	obese	0.024	**13**	5	Low	obese	0.087
5	2	Mid	average	0.048	**14**	5	Mid	average	0.174
6	2	High	emaciated	0.024	**15**	5	High	emaciated	0.087
7	3	Low	obese	0.011	**16**	6	Low	obese	0.072
8	3	Mid	average	0.022	**17**	6	Mid	average	0.144
9	3	High	emaciated	0.011	**18**	6	High	emaciated	0.072
19	Fft 1 is high or Fft 2 is high or Fft 3 is high or Fft 4 is high or Fft 5 is high or Fft 6 is high							emaciated	1

2.3 Performance Analysis

The Classifier's performance was tested using testing data, randomly separated from the training data at the beginning of the research. The classifier model was analyzed by its capability to distinguish between BCS classes. The numerical values after defuzzification were used to evaluate the model's performance. The fuzzy logic classification model was also compared to the basic algorithm classification (Bercovich, 2012), using the mean error between the model's output and the manual BCS.

$$Mean\_BSC_{error} = \frac{1}{n}\sum_{i=1}^{n} |BCS\_model_i - BCS\_manual_i|$$

In order to better understand the results and fully grasp the classifier's ability the median error and the number of correct classifications (with errors less than 0.3) were also recorded.

$$Median\_BSC_{error} = Median_{i=1}^{n} |BCS\_model_i - BCS\_manual_i|$$

2.3.1 Sensitivity analysis

After the Initial modeling of the classifier, the classifier properties were tested and changed in order to see the influences of the change and to determine the most accurate value for each property. The sensitivity analysis focused on slight changes around the following initial values: adding and removing input variables, changing the rules weights, changing the membership function type and size and testing defuzzification methods.

To ensure that the model training is performed on data, which accurately represents the population, additional sensitivity analyses tests were performed: K-fold cross validation (k=10), leave one out and random selection of training data.

3. Algorithm

3.1 Work process

In the initial construction phase, the core features of the fuzzy classifier were determined: the number of input variables was set to 6 out of the 10 Fourier descriptors chosen using stepwise regression, number of membership function in each variable was set to start as 3 for the input variable in order to properly distinguish between large, medium and small values and 5 for the output acording to the BCS, type of the membership functions, trapezoid that can be a triangularr function with some values and rules were set acording to the correlation between the input and the output (for example if X is high then Y is low) and their weights were set acording to the regression coefficients. Initial values were provided to all numerical features and the influence of all feature change was tested. The numerical features that define the system, rule weights and the defining values of all membership functions were changed and the mean error was retested. Two main algorithms were used in this repeated process: a heuristic that changes the defining values as long as the mean error does not increase and an optimization of a constrained nonlinear multivariable function using Matlab optimization toolbox. The values defining the membership functions are 4 values for each function, which together create a trapezoid function and can became a triangular function if the two middle values are equal. The rule weights are numbers between 0 and 1 that define the level of influence of each rule on the system; the rule weights can dismiss rules by having the value 0.

3.2 Heuristic

The heuristic was created in order to maximize improvevement in the system classification. The first heuristic implementation was a Matlab code that changed the defining numerical values of the fuzzy classifier to the direction with the maximumimprovement to the mean error and as long as the mean error improved the value continued changing in the same direction. The first heuristic did not achieve very good results due to the complexity of the system, Lack of change in the mean error in many cases and the fact that the values were changed only once and could not re-change to adapt to the other values in the system. Several changes were made in the heuristic code in order to improve its results: the code changes the values only in one direction and then the other instead of just one direction for each value. The values does not stop changing until the mean error grows instead of changing as long as the mean error decreases and the code is a repetitive code for long cycles or as long as there is a change between cycles. In Table 4, all the points defining the membership functions (3 membership functions for each variable – low, mid and high) for input values of the heuristic based system. Table 5 lists the detailed points that define the membership functions (5 membership functions – emaciated, thin, average, fat and obese) of the BCS output value for the heuristic based system. Table 6 shows the rules for the fuzzy classification system constructed and their weights.

Table 4–Values of input membership functions for heuristic based system

	Low 1	Low 2	Low 3	Low 4	Mid 1	Mid 2	Mid 3	Mid 4	High 1	High 2	High 3	High 4
Fft 1	-37.64	2.36	12.36	52.99	12.36	52.36	62.36	102.4	52.13	146.9	147.9	152.4

Fft 2	-15.14	4.86	9.86	29.86	9.86	29.86	34.86	54.86	27.39	34.98	72.36	79.86
Fft 3	-15.14	4.86	9.86	29.86	9.86	29.86	34.86	54.86	29.45	39.94	77.36	79.86
Fft 4	-6.14	2.86	4.58	24.96	4.75	11.19	17.9	20.04	12.26	31.21	46.87	50.37
Fft 5	-2.13	-0.86	0.1	9.32	7.87	15.42	17.87	25.87	10.48	24.69	33.42	35.87
Fft 6	-1.62	-0.62	2.4	6.5	2.898	7.998	10.4	16.4	8.12	12.12	26.17	28.62

Table 5- Values of output membership functions for heuristic based system

	Point 1	**Point 2**	**Point 3**	**Point 4**
Emaciated	-1.34	0.05	0.76	1.39
Thin	1.07	1.99	2.83	2.84
Average	1.84	2.04	2.65	3.83
Fat	2.54	3.04	3.54	4.34
Obese	4.07	4.46	5.28	5.82

Table 6–Rules weights of the heuristic based fuzzy classification system

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
0	0	0	0	0	0.5415	0	0	0	0.0265
11	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	
0.1985	1	0.228	0	0.019	0.073	0	1	1	

3.3 Optimization

Optimizing the fuzzy classification system using regular methods was a difficult task due to the inability to formulate a clear objective function. The Matlab optimization toolbox was applied; it allows working on a programmed Matlab function without defining the mathematical relations of the input variables and the output variable. A Matlab function was written, receiving an input of the defining numerical values of the fuzzy system, inserting them into the system and performing an evaluation of the training data. The functions output was the mean error of the classifier on the training data. In addition to the Matlab function, a set of constraints were created in order to keep the membership functions a legal trapeze function and the rule weights between one and zero. The function, constraints and an initial input vector were entered in to the Matlab "fmincon" algorithm – an algorithm to find minimum of constrained nonlinear multivariable function. The algorithm solves a quadratic programming sub-problem at each iteration by estimating the Hessian of the lagrangian. Foillowing the optimization, the optimized values were inserted into a fuzzy system and the training data's and the testing data's mean error. Median error and the number of correct classifications were evaluated. In Table 7, all the points defining the membership functions (3 membership functions for each variable – low, mid and high) for input values of the optimization based system. The points that define the membership functions (5 membership functions – emaciated, thin, average, fat and obese) of the BCS output value for the optimization based system are detailed in Table 8. Table 9 shows the rules for the fuzzy classification system constructed and their weights.

Table 7–Values of input membership functions for optimization based system

	Low 1	**Low 2**	**Low 3**	**Low 4**	**Mid 1**	**Mid 2**	**Mid 3**	**Mid 4**	**High 1**	**High 2**	**High 3**	**High 4**
Fft 1	-45	-5	4.991	45.01	5.019	45.04	55.03	95	55.48	95.28	105	145
Fft 2	-22.5	-2.5	2.512	22.31	2.506	22.52	27.5	47.5	25.22	47.07	52.5	72.5
Fft 3	-22.5	-2.5	2.497	22.5	2.605	22.5	27.5	47.5	27.81	47.42	52.5	72.5

Fft 4	-13.5	-1.5	1.482	13.54	4.145	13.57	16.5	28.5	13.84	27.84	31.5	43.5
Fft 5	-9	-1	0.9542	7.564	1.033	9.052	11.01	19	10.7	19.85	21	29
Fft 6	-6.75	-0.75	1.83	6.662	0.7557	6.767	8.25	14.25	8.644	13.93	15.75	21.75

Table 8 - Values of output membership functions for optimization based system

	Point 1	**Point 2**	**Point 3**	**Point 4**
Emaciated	0.1	0.9	1.1	1.9
Thin	0.3702	2.555	2.644	3.3
Average	0.928	2.434	2.894	3.299
Fat	3.1	3.9	4.1	4.9
Obese	4.689	4.724	5.1	5.9

Table 9–Rules of the optimization based fuzzy classification system

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
0.20237	0.017844	0.021534	0.6228	0.010265	0.03954	0.064417	0.015174	0.093792	0.078537
11	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	
0.21654	0.11023	0.83777	0.0028153	0.013283	0.36922	0.015464	0.077507	0.99903	

4. Results

4.1 Model performance

The results of the initial system did not show any improvement compared to previous studies (Table 10). However, they were sufficiently accurate to start the system training. The first algorithm tested was the training heuristic, which showed an improvement in the training data for all the performance measures and improved some of the performance measures of the testing data. Using the optimizing method, the training data performance measures improved even more but the results for the testing set were less compelling with a slight increase in some measures and a decrease in others (Table 10).

Table 10 - Results of different classification systems

	Training			**Testing**		
	Average	**Median**	**Correct classification**	**Average**	**Median**	**Correct classification**
Past study	0.286	0.238	50/87	0.343	0.315	30/64
Initial system	0.3961	0.3314	38/87	0.4245	0.3615	28/64
Optimization	0.2104	0.0978	64/87	0.4192	0.3264	31/64
Heuristic	0.226	0.123	63/87	0.389	0.288	33/64

Results indicate (Table 11) that both the optimization and the heuristic improved the classification for cows with abnormal BCS, especially for the cows with high BCS. However, the ability to classify the cows with the normal BCS was slightly damaged. For obese cows with BCS classes of 4-5, the classifier detected 100% of the images correctly compared to 0% in previous studies. For cows with BCS classes between 3 and 4 the classifier detected 19% compared to 33.3%, for BCS classes between 2 and 3, 62.2% were correctly classified compared to 56.8% (with non-fuzzy logic classification) and for cows with low BCS (classes 1-2) both algorithms detected 100%. In all cases the maximum misclassification was by one level. The overall classification performance of the two suggested systems was higher than the basic algorithm’s performance and can better deal with extreme cases.

Table 11- Classifications of different BCS

		BCS 1-2	BCS 2-3	BCS 3-4	BCS 4-5
Number of samples	**Training**	0	58	18	11
	Testing	2	37	21	4
Basic algorithm	**Training**	0	36 (62.1%)	9 (50%)	5 (45.5%)
	Testing	2 (100%)	21 (56.8%)	7 (33.3%)	0 (0%)
Correct classification optimization	**Training**	0	44 (75.9%)	10 (55.6%)	10 (90.9%)
	Testing	2 (100%)	21 (56.8%)	4 (19%)	4 (100%)
Correct classification Heuristic	**Training**	0	45 (77.6%)	8 (44.4%)	10 (90.9%)
	Testing	2 (100%)	23 (62.2%)	4 (19%)	4 (100%)

4.2 Model repeatability

In tables 12 and 13, the computed BCS, from the heuristic based system and the optimization based system of three different cows, is presented for 5 images each. The average standard deviation per cow (0.057 and 0.065) and the largest standard deviation (0.082 and 0.088) are both significantly smaller from the standard deviation of all samples (1.3 and 0.6).

Tables12 and 13- Model repeatability for heuristic and optimization based system

Number	2894	3186	3071
1	2.509176	2.973161	4.295906
2	2.59442	2.824187	4.220976
3	2.566709	2.802334	4.09974
4	2.60912	2.775606	4.197844
5	2.590761	2.768343	4.107791
Std	0.039328	0.053032	0.082141

Number	2894	3186	3071
1	2.438876	2.973161	4.285979
2	2.477088	2.824187	4.059411
3	2.467803	2.802334	4.114092
4	2.503183	2.775606	4.199902
5	2.473468	2.768343	4.119699
Std	0.022996	0.083735	0.088353

4.3 Sensitivity analyses

Analyses of the influence of the training data set were tested using three approaches: K-fold cross validation (K=10), leave one out and random selection of training set, which presented in Table 14. As the results show the classifiers constructed were within reasonable error range, the training and the testing results were close to errors computed in the training set sensitivity analyses – the error is larger but not two times bigger. The last test, random training set, showed results that were not as good as the other tests as expected. The high standard deviation in the testing data results indicate that there were a small number of samples that are different in their relationship between the Fourier descriptors and the BCS. These samples can be defined as exceptions and a likely reason for that is background dependency problem with the computer vision algorithm.

Table 14- Training set sensitivity analyses

	Training		Testing	
	Average	**Std**	**Average**	**Std**
K-Fold	0.2803	0.0112	0.2803	0.1132
Leave one out	0.2561	0.0219	0.2631	0.2866
Random training set	0.2308	0.0461	0.405	0.068

5. Discussion and Conclusion

A fuzzy logic classifier seems to be suitable for the purpose of matching a BCS to a cow. This being said, in classification systems using fuzzy logic the number of influential values are much larger than the number of variables. As with any classification, we should keep in mind that the system was built and trained on data from a specific computer vision algorithm and a specific farm; if changes were to be made in the computer vision algorithm, the system should be rebuilt and retrained and if the system were to be installed in a new farm, it would need to be retrained on images from that specific farm. In the system constructed in this research, for each variable there are three membership functions and for each membership function there are 4 numbers that define the position, size and shape. In addition, the rule weights are also numerical values that influence the system's clasification ability. The large number of influential values affects the training's ability to match all possible data with a relatively small sample size. It is recommended that the system will be retrained on a larger data set in order to confirm the results obtained in this research. In the heuristic based system, most rules weights were set to 0, canceling the rule and casing the middle membership functions in all input variables to be unimportant. In the optimization-based system, the position of the input membership function only slightly changed, showing the relatively small value of the input membership function position to the goal value. In both systems, the most important feature for discovering high BCS was the 6th Fourier descriptor. The classification systems developed in this study show a good ability to connect between Fourier descriptors and a BCS provided by a trained expert. The systems use fuzzy logic, a rule set based on the idea that a member does not have be a part of only one group and the level of membership to that a group does not have to be 1 or 0. The systems training was performed by two techniques; the first, a heuristic changing the key values of the system as long as the system's performance did not decrease, and the second, a built-in minimization tool of the Matlab optimization toolbox used to minimize the mean error of the system within the training data set. Both systems showed an improvement in the system's ability to classify, especially in the high BCS. In the sensitivity analysis, a small number of samples were incompatible with the majority of the samples in this classification method. This was most likely due to the background dependency of the computer vision part of the algorithm. We should remember that the fuzzy logic output is not a crisp number and in this study we used defuzzification methods in order to compare the result with previous studies.

6. Acknowledgements

We would like to thank Dr. Victor Alchanatis and Dr. Ephraim Maltz for their valuable contributions and Mr. Shamai Yaacobi for the technical assistance. This work was partially supported by the Ben-Gurion University of the Negev, Rabbi W. Gunther Plaut Chair in Manufacturing Engineering.

7. References

Azzaro G, Caccamo M, Ferguson J, Battiato S, Farinella G, Guarnera G, Puglisi G, Petriglieri R, &Licitra G (2011). Objective estimation of body condition score by modeling cow body shape from digital images. *Journal of dairy science*, 94(4), 2126-2137

Bercovich A (2012). AUTOMATIC COW'S BODY CONDITION SCORING. Ben-Gurion University of the Negev faculty of engineering sciences department of industrial engineering and management

Bewley J, Peacock A, Lewis O, Boyce R, Roberts D, Coffey M, Kenyon S, &Schutz M (2008). Potential for estimation of body condition scores in dairy cattle from digital images. *Journal of dairy science*, 91(9), 3439-3453

Ferguson J D, Galligan D T, & Thomsen N (1994). Principal descriptors of body condition score in Holstein cows. *Journal of dairy science*, 77(9), 2695-2703

Hady P, Domecq J, &Kaneene J (1994). Frequency and precision of body condition scoring in dairy cattle. *Journal of dairy science*, 77(6), 1543-1547

Halachmi I, Polak P, Roberts D, &Klopcic M (2008).Cow body shape and automation of condition scoring. *Journal of dairy science*, 91(11), 4444-4451

Wildman E, Jones G, Wagner P, Boman R, Troutt H, &Lesch T (1982). A dairy cow body condition scoring system and its relationship to selected production characteristics. *Journal of dairy science*, 65(3), 495-501

Yu Y,& Fowler E (1994). Rule based fuzzy logic inferencing. *IEEE International Conference on Information and Technology*, 1, 465-470