

Chapter 2

Background and Theory

The focus of this book emphasizes the field of Granular Computing, where by nature is a vast area still under development. This section summarizes this matter as well as derived topics which aid Granular Computing, such as Fuzzy Sets and clustering algorithms.

2.1 Granular Computing

Granular Computing (GrC) is an area which was conceived by L.A. Zadeh in 1979 [1] which tries to adequate information granules to the information data that it tries to model. That is, an information granule is an atomic expression derived from an extracted model via a sample of data, where such granule can be small as to obtain a specific description of a part of a model, or coarser, denoting less precision and more generality. This being controlled by the requirements of the final granular model.

GrC is inspired by how the human brain processes information, in how it takes sets of factual data as to get a representation where a decision is required, then groups such data into different levels of resolution where each level can be ambiguous or very specific, suited to what can more effectively give an answer and finally make a correct decision based on collected data. With this in mind, GrC tries to reproduce this very behavior and take it into a computational algorithm which takes data, forms information granules and obtains a model more in affinity with reality.

It must be noted that GrC does nothing by itself; instead it works in conjunction with other algorithms where it applies its core theories and methodologies in order to obtain better information granules, and as such obtain better granular models.

2.2 Information Granule Representations

A key topic in GrC is information granule representation, where the simplest form of representation is by means of intervals, by which a delimitation of granule size exists thus having clear knowledge of what data is covered by such information granule. A clear example of this could be an age range, where the Universe of Discourse covers from 0 to 100 years old, but since a more specific age range is desired, such as 40–57 years old, this clearly eliminates all data outside this range, i.e. the ranges 0–39 years old and 58–100 years old is ignored. Although an interval is a simple representation, interval arithmetic [2] is not, and still requires more research to fully implement a viable solution to fuse GrC with interval arithmetic.

Another information granule representation schema which is also simple is Set Theory, represented by collection of items of the same type or categorical type. Similar to representation by intervals, an item either belongs or does not to a group, or set. Yet the main difference with intervals is that its mathematical operations are very mature. Although, as its representation schema is simple, more real world information granules cannot be properly represented.

The list of existing information granule representations is quite extensive, e.g. quotient space [3], fuzzy sets [4], rough sets [5], neural networks [6], shadowed sets [7], neutrosophic sets [8]; giving detailed attentions to each one would require much unnecessary attention which will not be given. Therefore, the chosen and used representation for information granules in this book is Fuzzy Sets (FS) since they are a representation which have much similarities with the core concept of GrC, which is to represent human cognition where imprecision is used throughout linguistic variables to represent granular models. With FSs an imprecise belongingness value can be represented where such imprecision is between the interval $[0, 1]$; zero, being that it does not belong at all to the set, one, being that it completely belongs to the set, and where any in-between value represents a certain degree of belonging to the set. With this in mind, FSs are an excellent representation for information granules, where in comparison with intervals or normal sets, where a datum either belongs or not to a set, with FSs a datum can simultaneously belong to multiple sets, although at different degrees, which is similar to human cognition.

2.3 Principle of Justifiable Granularity

This principle is a technique which purpose is to specify the adequate size of an information granule in such a way that it is not too small and has sufficient coverage of experimental data while at the same time it does not have too much coverage as to over generalize the granule. Figure 2.1 visually shows both these differences.

A double optimization must exist which takes into account both objectives:

1. The information granule must be as specific as possible.
2. The information granule must have sufficient numerical evidence.

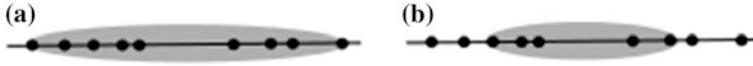


Fig. 2.1 Visual representation of two different objectives in data coverage, where **a** full experimental data coverage is achieved by the information granule, and **b** a more specific information granule covering only a portion of the experimental data

This double optimization is performed twice, as the length of the information granule has two sides which must be optimized, the left side interval from the Median of the data sample and the right side interval from the median of the data sample, as shown in Fig. 2.2. Both intervals, *a* and *b*, start from the $Med(D)$ of available data D which formed said information granule.

Each information granule's set, shown in Eq. (2.1), is important, since it is the basis for finding the required lengths used to obtain a meaningful information granule. Where x_k are the individual data which belong to the information granule Ω .

$$\text{set}(x_k \in \Omega) \quad (2.1)$$

The set of is separated into two sections, as shown in Eqs. (2.2) and (2.3), as to conform to each interval, *a* and *b* respectively.

$$\text{set}(x_k \in \Omega, > Med(D)) \quad (2.2)$$

$$\text{set}(x_k \in \Omega, < Med(D)) \quad (2.3)$$

As for the required double optimization, it is obtained by maximizing Eqs. (2.4) or (2.5), for *a* and *b* respectively. Which uses a user criterion $\alpha \in [0, \alpha_{max}]$, where α_{max} obtains the smallest possible length achieved by the principle of justifiable granularity, and can be obtained via Eqs. (2.6) and (2.7), for *a* and *b* respectively. Described as the natural logarithm of the cardinality of the chosen side (*a* or *b*) divided by the length of the closest datum x_1 to $Med(D)$.

$$V(a^*) = \max_{a < Med(D)} [V(a)] \quad (2.4)$$

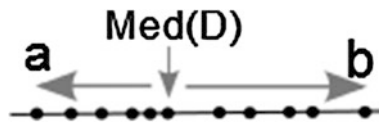


Fig. 2.2 Intervals *a* and *b* are separately optimized based on available numerical evidence from the formation of said information granule. Where both lengths start at the median of the information granule

$$V(b^*) = \max_{b > \text{Med}(D)} [V(b)] \quad (2.5)$$

$$\alpha_{\max}^a = \frac{\text{card}(x_k \in \Omega, < \text{Med}(D))}{|\text{Med}(D) - x_1|} \quad (2.6)$$

$$\alpha_{\max}^b = \frac{\text{card}(x_k \in \Omega, > \text{Med}(D))}{|\text{Med}(D) - x_1|} \quad (2.7)$$

Regarding the double optimization itself, shown in Eqs. (2.8) and (2.9), for a and b respectively. Which is an integration of the probability density function from $\text{Med}(D)$ to all prototypes of a , or b , multiplied by the user criterion for specificity α . In Appendix A, the demonstration of how Eqs. (2.8) and (2.9) are transformed into computational models is shown. And, in Appendix C.1, the code which computes the values for a and b is shown.

$$V(a) = e^{(-\alpha|\text{Med}(D)-a|)} \int_a^{\text{Med}(D)} p(x)dx \quad (2.8)$$

$$V(b) = e^{(-\alpha|b-\text{Med}(D)|)} \int_{\text{Med}(D)}^b p(x)dx \quad (2.9)$$

As an example of the behavior of $V(b)$ in respect to the optimal length in respect to the chosen value of α , Fig. 2.3 shows how the peak of each curve locates the optimum length.

Finally, another example is shown, in Fig. 2.4, that demonstrates how the curve optimization affects both intervals, a and b , of the information granule. Here, it can clearly be seen where the optimal length is located in respect the peak of the optimization curve from the principle of justifiable granularity.

A more detailed study of the behavior of the principle of justifiable granularity can be found in [9].

2.4 Data Granulation Algorithms

To obtain granular models, help is needed from algorithms which can create these models from experimental data. In GrC it is much more common to find machine learning techniques rather than traditional statistical methods. With these intelligent algorithms, the most common sub-group of algorithms are clustering algorithms, due to their focus on finding similarities between the data itself and for differentiating between these found groups. This action generates a model from the data.

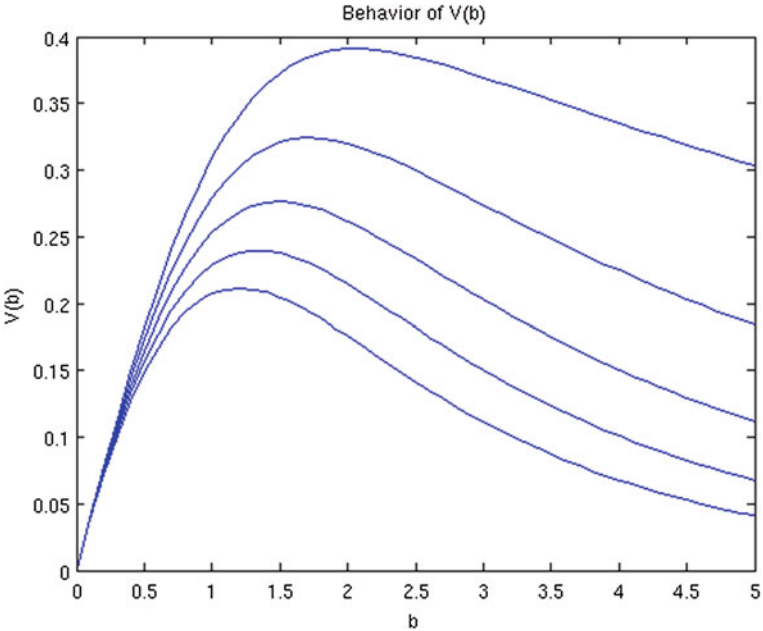


Fig. 2.3 Behavior of $V(b)$ in respect to b , with changing values of α

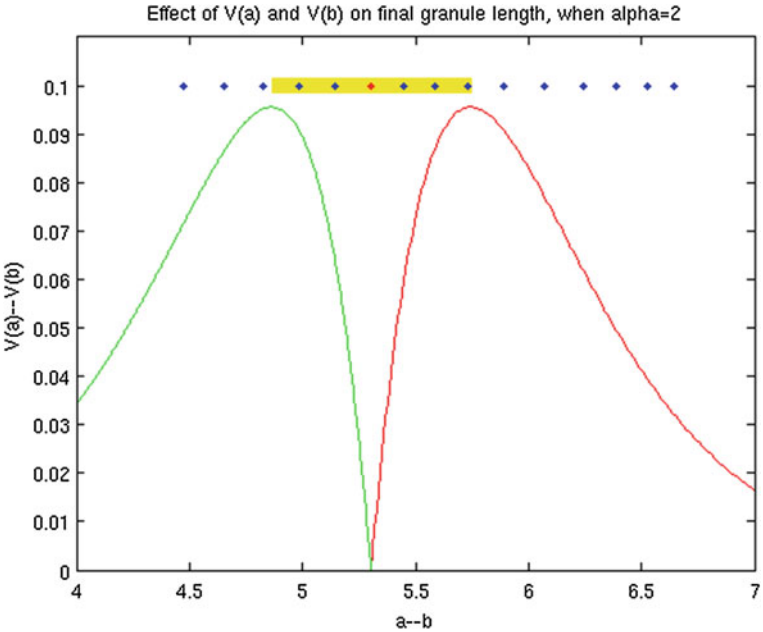


Fig. 2.4 Effect of $V(a)$ and $V(b)$ on the final size of the information granule, when $\alpha = 2$

Found groups of similar data are now called information granules which represent abstract models of a portion of relevant information as obtained from a given phenomenon. And the process of obtaining such information granules is called *data granulation*.

There is a great quantity of existing clustering algorithms, and among the most applied in GrC are K-Means [10] and Fuzzy C-Means (FCM) [11]. Where the K-Means algorithm contains a partition matrix that specifies which datum exactly belongs to which group, whereas the FCM algorithm specifies how much each datum belongs to each group. In K-means, each datum can only belong to one found group; while in FCM, each datum can belong to multiple groups, although in different degrees.

Many more clustering algorithms exist which perform the task of granulating data, such as the subtractive algorithm [12], or the fuzzy granular gravitational clustering algorithm [13] which will be seen in more detail further down in this document.

2.5 Fuzzy Logic

Fuzzy Logic (FL) can be seen as an advancement from bivariate logic. Where only two values can be expressed $\{0, 1\}$, whereas in FL values can be anything within the interval $[0, 1]$. This interval can express different degrees of perception, such as *not too much*, *very little*, or *more or less*. When used in a FS, aptly named a Type-1 Fuzzy Set (T1 FS), it can give better perceptual representations since ambiguities can now be modeled. Although many perceptual situations can be modeled using T1 FS, uncertainty is not one. For this, Interval Type-2 Fuzzy Sets (IT2 FSs) can be used, where its model directly handles, apart from imprecision, uncertainty. As for the complete model for Type-2 Fuzzy Sets, where T1 FSs and IT2 FSs are simplifications of Type-2 Fuzzy Sets. General Type-2 Fuzzy Sets (GT2 FSs) exist which in essence have a better handling of uncertainty than IT2 FS, this is due to how uncertainty is represented, instead of a 2D area, it is represented by a 3D volume.

2.5.1 Type-1 Fuzzy Sets

A T1 FS A , expressed by $\mu_A(x)$ where $x \in X$, described as $A = \{(x, \mu_A(x)) | x \in X\}$. A visual example is shown in Fig. 2.5, where a generic Gaussian membership function represents a T1 FS.

A Type-1 Fuzzy Logic System (T1 FLS) can be easily described by a block diagram, shown in Fig. 2.6. Where the Fuzzifier takes crisp inputs and maps them into FS; the Inference, based on Rules, maps Fuzzy Sets from the antecedents to FSs from the consequents; finally, the Output Processor defuzzifies and outputs a crisp value.

The rule set for T1 FLSs are in the format as shown in Eq. (2.10) where the relation between the input and output space is mapped. Where R^l is a specific rule, x_p is input p ,

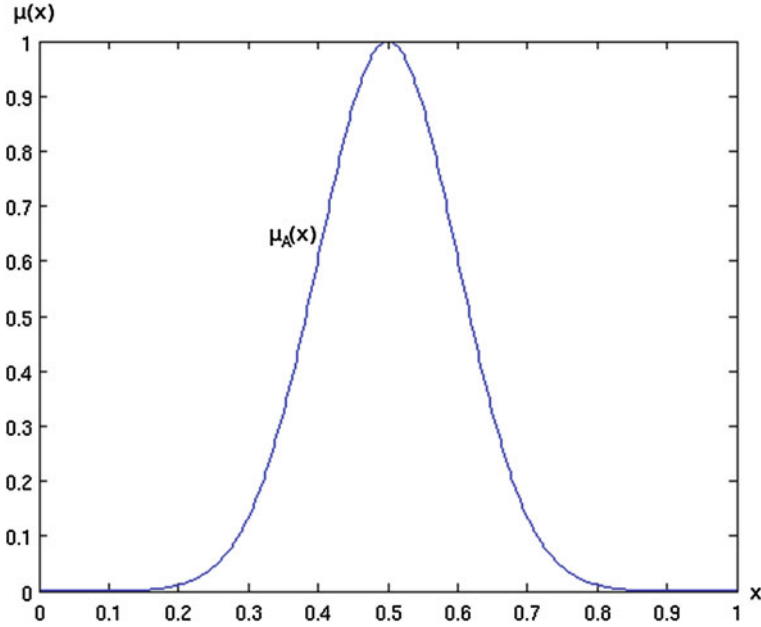


Fig. 2.5 Example of a generic T1 FS in the form of a Gaussian membership function

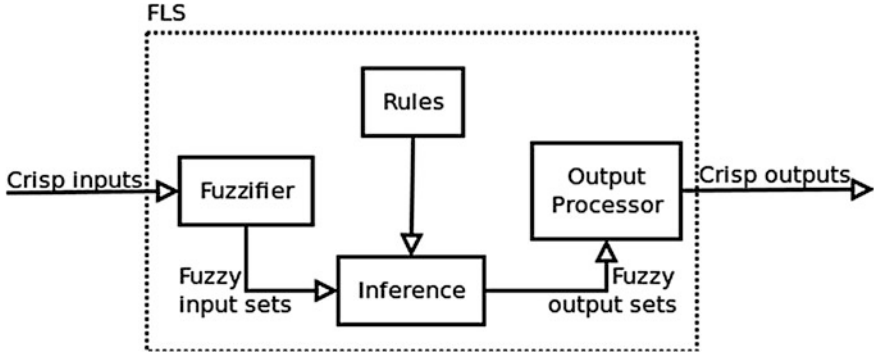


Fig. 2.6 Block diagram describing the main components of a T1 FLS

F_p^l is a membership function on rule l and input p , y is the output on membership function G^l . Both F and G are in the form of $\mu_F(x)$ and $\mu_G(y)$ respectively.

$$R^l : \text{IF } x_1 \text{ is } F_1^l \text{ and } \dots \text{ and } x_p \text{ is } F_p^l, \quad \text{THEN } y \text{ is } G^l, \quad \text{where } l = 1, \dots, M \quad (2.10)$$

As for the inference which calculates the compatibility between the antecedents and the consequents, using t-norms ($*$), Eq. (2.11) shows the basic methodology

required to process such t-norms. Where μ_{B^l} is the consequent membership function after being processed by the antecedents and Y is the domain space for the consequents.

$$\mu_{B^l}(y) = \mu_{G^l}(y) \tilde{*} \left\{ \left[\sup_{x_1 \in X_1} \mu_{x_1}(x_1) \tilde{*} \mu_{F_1^l}(x_1) \right] \tilde{*} \cdots \tilde{*} \left[\sup_{x_p \in X_p} \mu_{x_p}(x_p) \tilde{*} \mu_{F_p^l}(x_p) \right] \right\},$$

$$y \in Y \quad (2.11)$$

The defuzzification process can be achieved in many ways, each obtaining similar results. Examples of common defuzzifiers are centroid, shown in Eq. (2.12); center-of-sums, shown in Eq. (2.13); or heights, shown in Eq. (2.14). Where y_i is a discrete position from Y , $y_i \in Y$, $\mu_B(y)$ is a FS which has been mapped from the inputs, c_{B^l} denotes the centroid on the l th output, a_{B^l} is the area of the set, and \bar{y}^l is the point which has the maximum membership value in the l th output set.

$$y_c(x) = \frac{\sum_{i=1}^N y_i \mu_B(y_i)}{\sum_{i=1}^N \mu_B(y_i)} \quad (2.12)$$

$$y_a(x) = \frac{\sum_{l=1}^M c_{B^l} a_{B^l}}{\sum_{l=1}^M a_{B^l}} \quad (2.13)$$

$$y_h(x) = \frac{\sum_{l=1}^M \bar{y}^l \mu_{B^l}(\bar{y}^l)}{\sum_{l=1}^M \mu_{B^l}(\bar{y}^l)} \quad (2.14)$$

2.5.2 Type-2 Fuzzy Sets

Type-2 Fuzzy Sets can be used in two forms, by its simplified form, i.e. IT2 FSs; or its complete form, i.e. GT2 FSs. Both will be briefly described in this section.

An IT2 FS \tilde{A} is represented by $\underline{\mu}_{\tilde{A}}(x)$ and $\overline{\mu}_{\tilde{A}}(x)$ which are the lower and upper membership functions respectively of $\mu_{\tilde{A}}(x)$, described as $\tilde{A} = \{((x, u), \mu_{\tilde{A}}(x, u) = 1) \mid \forall x \in X, \forall u \in [0, 1]\}$, where x is a subset of the Universe Of Discourse X , and u is a mapping of X into $[0, 1]$. A visual example is shown in Fig. 2.7, where a generic Gaussian membership function with uncertain standard deviation represents an IT2 FS.

An Interval Type-2 Fuzzy Logic System (IT2 FLS) can be easily described by a block diagram, shown in Fig. 2.8. Where the Fuzzifier takes crisp inputs and maps them into FS; the Inference, based on the Rules, maps Fuzzy Sets from the antecedents to FS from the consequents; then, the output can be chosen between

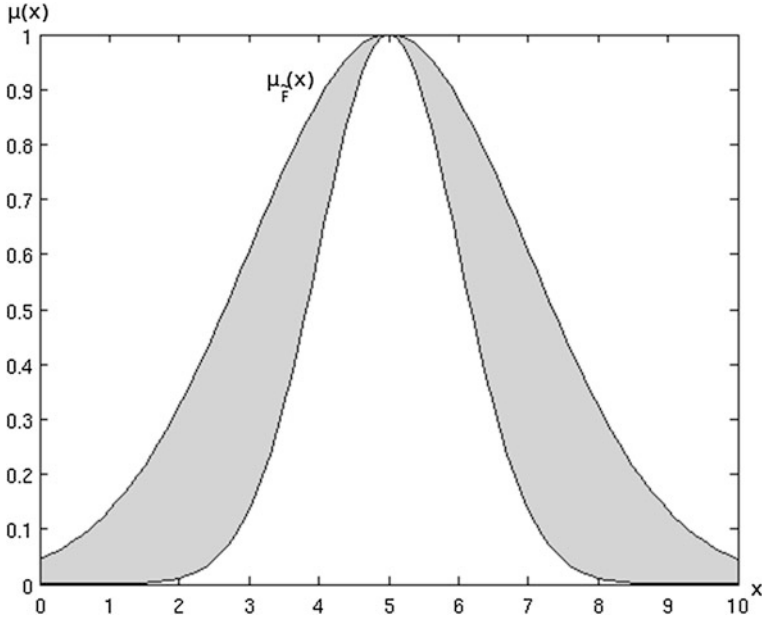


Fig. 2.7 Example of a generic IT2 FS in the form of a Gaussian membership function with uncertain standard deviation

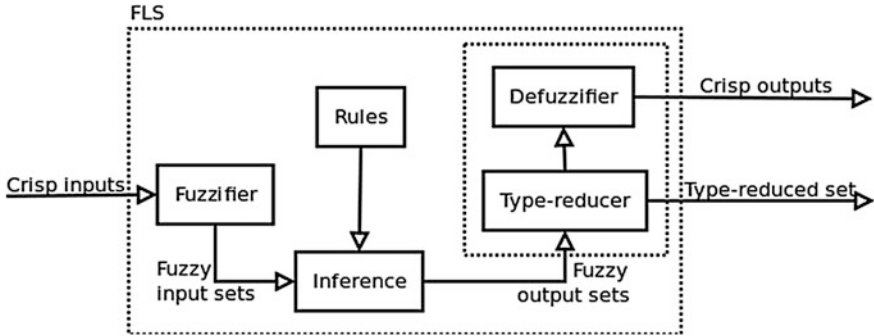


Fig. 2.8 Block diagram describing the main components of an IT2 FLS

obtaining a Type-reduced set (T1 FS) or Output Processor which defuzzifies and outputs a crisp value.

The rule set of an IT2 FLS maintains the same format as for the T1 FLS, but with a small notation difference, as shown in Eq. (2.15).

$$R^l : \text{IF } x_1 \text{ is } \tilde{F}_1^l \text{ and } \dots \text{ and } x_p \text{ is } \tilde{F}_p^l, \quad \text{THEN } y \text{ is } \tilde{G}^l, \quad \text{where } l = 1, \dots, M \quad (2.15)$$

The IT2 FLS inference can be summarized by Eq. (2.16). Where \underline{f}^l and \bar{f}^l represent the firing sets defined by Eqs. (2.16) and (2.17), and b is a interval defined by Eqs. (2.18) and (2.19).

$$\mu_{\bar{B}}(y) = \frac{1}{b}, \quad y \in Y$$

$$b \in \left[\left[\underline{f}^1 \tilde{\mu}_{\bar{G}^1}(y) \right]_{v \dots v} \left[\left[\underline{f}^N \tilde{\mu}_{\bar{G}^N}(y) \right] \right], \left[\bar{f}^1 \tilde{\mu}_{\bar{G}^1}(y) \right]_{v \dots v} \left[\left[\bar{f}^N \tilde{\mu}_{\bar{G}^N}(y) \right] \right] \right] \quad (2.15)$$

$$\underline{f}^l(x') = \underline{\mu}_{\bar{F}^1_l}(x'_1) \tilde{\mu} \cdots \tilde{\mu}_{\bar{F}^l_{p_l}}(x'_p) \quad (2.16)$$

$$\bar{f}^l(x') = \bar{\mu}_{\bar{F}^1_l}(x'_1) \tilde{\mu} \cdots \tilde{\mu}_{\bar{F}^l_{p_l}}(x'_p) \quad (2.17)$$

$$\underline{b}^l(y) = \underline{f}^l \tilde{\mu}_{\bar{G}^l}(y) \quad (2.18)$$

$$\bar{b}^l(y) = \bar{f}^l \tilde{\mu}_{\bar{G}^l}(y) \quad (2.19)$$

A common technique for type-reducing an IT2 FLS is center-of-sets (cos), shown in Eq. (2.20). Where Y_{\cos} is an interval set defined by two points $[y_l^i, y_r^i]$, which are obtained from Eqs. (2.21) and (2.22). Although other type-reducing techniques do exist, such as [14, 15].

$$Y_{\cos}(x) = \int_{y_l^1 \in [y_l^1, y_r^1]} \cdots \int_{y^M \in [y_l^M, y_r^M]} \int_{f^1 \in [\underline{f}^1, \bar{f}^1]} \cdots \int_{f^M \in [\underline{f}^M, \bar{f}^M]} 1 / \frac{\sum_{i=1}^M f^i y_l^i}{\sum_{i=1}^M f^i} \quad (2.20)$$

$$y_l = \frac{\sum_{i=1}^M f_l^i y_l^i}{\sum_{i=1}^M f_l^i} \quad (2.21)$$

$$y_r = \frac{\sum_{i=1}^M f_r^i y_r^i}{\sum_{i=1}^M f_r^i} \quad (2.22)$$

If a crisp output value is desired, a defuzzification of the type-reduced set y_l and y_r can be done, as shown in Eq. (2.23).

$$y(x) = \frac{y_l + y_r}{2} \quad (2.23)$$

A GT2 FS \tilde{A} described by $\tilde{A} = \left\{ \left((x, u), \mu_{\tilde{A}}(x, u) \right) \mid \forall x \in X, \forall u \in [0, 1] \right\}$. Where $\mu_{\tilde{A}}(x, u)$ is the set of secondary membership functions which form the uncertainty on the GT2 FS, x is the domain of the primary membership function,

and u is the domain of the secondary membership functions. Figure 2.9 shows a sample GT2 FS as seen from an orthographic top view for a generic Gaussian primary membership function with uncertain mean and a Gaussian secondary membership function. Figure 2.10 also shows the same GT2 FS but with an isometric view.

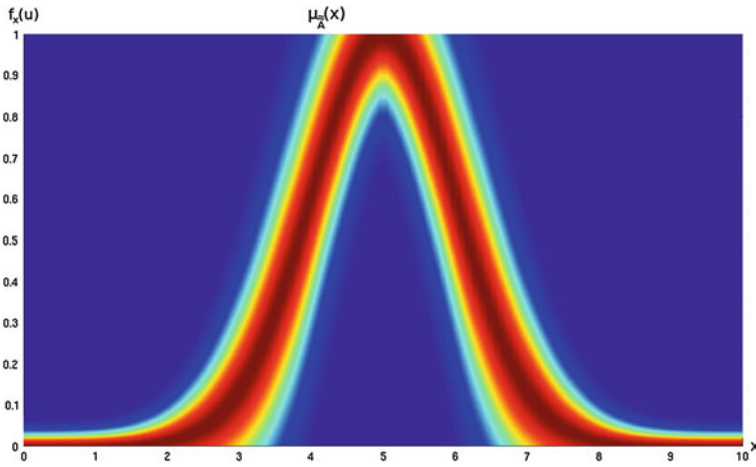


Fig. 2.9 Example of a GT2 FS in the form of a Gaussian primary membership function with uncertain mean and Gaussian secondary membership function, as seen from a orthographic top view

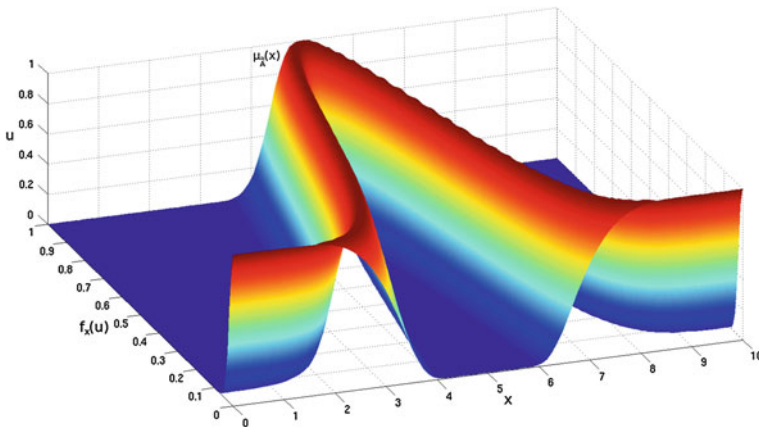


Fig. 2.10 Example of a GT2 FS in the form of a Gaussian primary membership function with uncertain mean and Gaussian secondary membership function, as seen from an isometric view

The rule set for a GT2 FLS maintains the format for T1 FLSs and IT2 FLSs, but with a slight notation difference, shown in Eq. (2.24).

$$R^l : \text{IF } x_1 \text{ is } \tilde{F}_1^l \text{ and } \dots \text{ and } x_p \text{ is } \tilde{F}_p^l, \quad \text{THEN } y \text{ is } \tilde{G}^l, \quad \text{where } l = 1, \dots, M \quad (2.24)$$

The inference of a GT2 FLS is quite different from the previous shown FLS (T1 and IT2), as the complete and original inference is very computational complex, hence the FLS simplifications of T1 and IT2. Yet it can be summed by two main operations, *meet* and *join*, shown in Eqs. (2.25) and (2.26) respectively. Analogous to the inference operations of either T1 FLS or IT2 FLS, they are used together in order to find the compatibility of the antecedents with the inputs, and accordingly their mapping into the consequents space.

$$\mu_{\tilde{A}}(x) \sqcup \mu_{\tilde{B}}(x) = \left\{ \left[\int_{u \in J_x^u} \int_{w \in J_x^w} f_x(u) \tilde{*} g_x(w) / (u \vee w) \right] \right\} \quad (2.25)$$

$$\mu_{\tilde{A}}(x) \sqcap \mu_{\tilde{B}}(x) = \left\{ \left[\int_{u \in J_x^u} \int_{w \in J_x^w} f_x(u) \tilde{*} g_x(w) / (u \wedge w) \right] \right\} \quad (2.26)$$

The defuzzification section of a GT2 FLS uses a centroid technique, shown in Eq. (2.27). Where $C_{\tilde{A}}$ defines the centroid of a GT2 FS, and θ_i is a combination associated to the secondary degree $f_{x_1}(\theta_1) \tilde{*} \dots \tilde{*} f_{x_N}(\theta_N)$. It must be noted that this defuzzification is of $O(n^n)$, which basically makes it uncomputational as the discretization increases. Although newer techniques exist which can highly reduce this into a more manageable and viable algorithm, via approximations.

$$C_{\tilde{A}} = \int_{\theta_1 \in J_{x_1}} \dots \int_{\theta_N \in J_{x_N}} [f_{x_1}(\theta_1) \tilde{*} \dots \tilde{*} f_{x_N}(\theta_N)] / \frac{\sum_{i=1}^N x_i \theta_i}{\sum_{i=1}^N \theta_i} \quad (2.27)$$

GT2 FLS approximations reduce the computational complexity of its original set operations, especially the defuzzification. These approximation techniques involve reducing the three-dimensional GT2 FS into multiple, and manageable, IT2 FSs.

One such approximation technique is done by obtaining an α -Plane [16] via the union of the α -cuts on all vertical slices on x_i . An α -Plane can be defined by Eq. (2.28) where \tilde{A}_α is an IT2 FS.

$$\tilde{A}_\alpha = \int_{\forall x \in X} \int_{\forall u \in [0,1]} \{ (x, u) | f_x(u) \geq \alpha \} \quad (2.28)$$

Another approximation technique is a z Slice [17], which obtains a plane by calculating the intervals for all vertical cuts on a given z . This z Slice is defined by Eq. (2.29). Here, the difference between an α -Plane and a z Slice is that an α -Plane performs its cuts on α/\tilde{A}_α , while a z Slice cuts on \tilde{Z}_α projected unto $X \times Y$.

$$\tilde{A}_z = \int_{\forall x \in X} \int_{\forall y \in [0,1]} z/(x, y) \quad (2.29)$$

When an α -Plane or a z Slice is calculated, the inference of an IT2 FLS is used. The union of all α -Planes or z Slices is defined by Eqs. (2.30) and (2.31) for α -Planes and z Slices respectively. Where $R_{\tilde{A}_\alpha}$ is one horizontal slice at level α .

$$\tilde{A} = \bigcup_{\forall \alpha \in [0,1]} R_{\tilde{A}_\alpha} \quad (2.30)$$

$$\tilde{A} = \bigcup_{\forall z \in [0,1]} \tilde{A}_z \quad (2.31)$$

2.6 Fuzzy Granular Computing

Having seen a description of what both GrC and Fuzzy Logic are, the term Fuzzy Granular Computing can be inferred from a combination of both. It is the direct application of the abstract concepts from GrC specifically when used with Fuzzy Logic. The concept of a FS is analogous to the concept of an Information Granule, where each represents an abstract model for a collection of data. And a collection of FSs, or Fuzzy Information Granules, becomes a Fuzzy Granular model.

References

1. Zadeh, L.A.: Fuzzy sets and information granularity. Adv. Fuzzy Set Theor. Appl. 3–18 (1979)
2. Hickey, T., Ju, Q., Van Emden, M.H.: Interval arithmetic: from principles to implementation. J. ACM **48**(5), 1038–1068 (2001)
3. Zhang, L.Z.L., Zhang, B.Z.B.: Quotient space based multi-granular computing. In: 2005 IEEE International Conference on Granular Computing, vol. 1, p. 98 (2005)
4. Pedrycz, W., Song, M.: Granular fuzzy models: a study in knowledge management in fuzzy modeling. Int. J. Approx. Reason. **53**(7), 1061–1079 (2012)
5. Sai, Y.S.Y., Nie, P.N.P., Chu, D.C.D.: A model of granular computing based on rough set theory. In: 2005 IEEE International Conference on Granular Computing, vol. 1, pp. 233–236 (2005)

6. Pedrycz, W., Vukovich, G.: Granular neural networks. *Neurocomputing* **36**(1–4), 205–224 (2001)
7. Pedrycz, W., Vukovich, G.: Granular computing with shadowed sets. *Int. J. Intell. Syst.* **17**(2), 173–197 (2002)
8. Solis, A.R., Panoutsos, G.: Granular computing neural-fuzzy modelling: a neutrosophic approach. *Appl. Soft Comput.* **13**(9), 4010–4021 (2013)
9. Sanchez, M.A., Castillo, O., Castro, J.R.: An analysis on the intrinsic implementation of the principle of justifiable granularity in clustering algorithms. In: O. Castillo, P. Melin, J. Kacprzyk (eds.) *Recent Advances on Hybrid Intelligent Systems*, vol. 451. Springer, Berlin, Heidelberg, pp. 121–134 (2013)
10. Lloyd, S.: Least squares quantization in PCM. *IEEE Trans. Inf. Theory* **28**(2), 129–137 (1982)
11. Bezdek, J.C., Ehrlich, R., Full, W.: FCM: the fuzzy c-means clustering algorithm. *Comput. Geosci.* **10**(2), 191–203 (1984)
12. Chen, M.-Y.: A hybrid ANFIS model for business failure prediction utilizing particle swarm optimization and subtractive clustering. *Inf. Sci. (Ny)* **220**, 180–195 (2013)
13. Sanchez, M.A., Castillo, O., Castro, J.R., Rodríguez-Díaz, A.: Fuzzy granular gravitational clustering algorithm. *North Am. Fuzzy Inf. Process. Soc.* **2012**, 1–6 (2012)
14. Yeh, C.-Y., Jeng, W.-H.R., Lee, S.-J.: An enhanced type-reduction algorithm for type-2 fuzzy sets. *IEEE Trans. Fuzzy Syst.* **19**(2), 227–240 (2011)
15. Mendel, J.M.: Enhanced Karnik-Mendel algorithms. *IEEE Trans. Fuzzy Syst.* **17**(4), 923–934 (2009)
16. Mendel, J., Liu, F., Zhai, D.: α -plane representation for type-2 fuzzy sets: theory and applications. *IEEE Trans. Fuzzy Syst.* **17**(5), 1189–1207 (2009)
17. Wagner, C., Hagens, H.: Toward general type-2 fuzzy logic systems based on zSlices. *IEEE Trans. Fuzzy Syst.* **18**(4), 637–660 (2010)

Type-2 Fuzzy Granular Models

Sanchez, M.A.; Castillo, O.; Castro, J.R.

2017, VIII, 93 p. 60 illus., 51 illus. in color., Softcover

ISBN: 978-3-319-41287-0