UNCERTAINTIES 2012

Proceedings of the 1st International Symposium on Uncertainty Quantification and Stochastic Modeling
February 26th to March 2nd, 2012, Maresias, São Sebastião, SP, Brazil

Protein Structure Reconstruction with Data Uncertainties

Ivan Sendin[1], **Siome K. Goldenstein**[2], **Carlile Lavor**[3]

[1] Dept. of Computer Science-CAC, Federal University of Goias
[2] Institute of Computing, State University of Campinas
[3] Dept. of Applied Mathematics, State University of Campinas

Abstract. *Nuclear Magnetic Resonance (NMR) experiments produce imprecise data when applied to protein structure calculations. We propose a new hybrid method that uses affine arithmetic and particles to control the uncertainty propagation, while the protein structure is constructed. The method was successfully applied in some artificial instances that simulate NMR data*

Keywords. *Molecular Distance Geometry Problem, Interval Methods,Uncertainty Propagation,Affine Arithmetic,Particles*

1 INTRODUCTION

Finding the three-dimensional structure of a protein is one of the first fundamental steps in structural bioinformatics applications, such as drug design, protein comparison, and docking.

One important method to obtain the structure of a protein is Nuclear Magnetic Resonance(NMR). Using NMR we can obtain a set of imprecise distances related to pairs of atoms of a protein. Together with distances associated to covalent bonds and covalent angles and other chemical constraints, an optimization process is performed in order to obtain the protein structure (Güntert,1998;Wüthrich,1986).

1.1 Geometric Build-Up

Recently, Dong and Wu (Dong,2002) developed an linear-time algorithm to obtain a protein structure using sufficient exact inter-atomic distances. The algorithm is straightforward. First, it finds a 4 vertex clique and creates a geometric base to be used for all other atoms (see Algorithm 1), then it uses the fact that a position of point in space can be determined solving a linear system if we know the distance from this point to 4 other points whose positions are known(see Algorithm 2). The algorithm iteratively expands the base until all atomic positions are determined (Algorithm 3).

This algorithm works only with exact distances, so it can not be used with NMR data.

Input: Distances $d_{12}, d_{13}, d_{14}, d_{23}, d_{24}, d_{34}$
Output: Points (x_i, y_i, z_i) for $i \in \{1,2,3,4\}$
$(x_1, y_1, z_1) = (0,0,0)$
$(x_2, y_2, z_2) = (d_{12}, 0, 0)$
$x_3 = (d_{13}^2 * d_{23}^2)/(d_{12} * 2) + (d_{12}/2)$
$y_3 = \sqrt{d_{13}^2 - x_3^2}$
$z_3 = 0$
$x_4 = (d_{14}^2 - d_{24}^2)/(d_{12} * 2) + d_{12}/2$
$y_4 = d_{24}^2 - d_{34}^2 - (x_4 - d_{12})^2 + (x_4 - x_3)^2$
$y_4 = y_4/(y_3 * 2) + y_3/2$
$z_4 = \sqrt{d_{14}^2 - x_4^2 - y_4^2}$

Algorithm 1: Create Geometric Base

2 UNCERTAINTY REPRESENTATION

2.1 Range based models

A uncertain value can be represented as a range that contains this value. Interval Arithmetic(IA) (Moore,1966) represents a unknown value $\bar{x}$ in an interval of real numbers: $\bar{x} = [x.lo\_\_x.hi]$ with the "true" value x satisfying $x.lo \leq x \leq x.hi$. The **range** of the unknown portion of an interval $\bar{x}$ is given by $x.hi - x.lo$ For example, the constant π can be represented by $\bar{p} = [3.14, 3.15]$ with range 0.01.

A general IA operation $\otimes$ is defined as:

$$\bar{a} \otimes \bar{b} = \{a \otimes b | a \in \bar{a}, b \in \bar{b}\} \quad (1)$$

Input: Points $p_i = (x_i, y_i, z_i)$ and distances d_i for $i \in \{1,2,3,4\}$
Output: $p = (x, y, z)$

$$A = 2 \begin{bmatrix} x_1 - x_2 & y_1 - y_2 & z_1 - z_2 \\ x_1 - x_3 & y_1 - y_3 & z_1 - z_3 \\ x_1 - x_4 & y_1 - y_4 & z_1 - z_4 \end{bmatrix}$$

$$B = \begin{bmatrix} ||p_1||^2 - ||p_2||^2 - (d_1^2 - d_2^2) \\ ||p_1||^2 - ||p_3||^2 - (d_1^2 - d_3^2) \\ ||p_1||^2 - ||p_4||^2 - (d_1^2 - d_4^2) \end{bmatrix}$$

solve $Ap = B$

Algorithm 2: Calculate Point

Input: $G(V,E)$ with exact distances in edges
Output: $G(V,E)$ with position in vertices
D = findClique4(G)
create a base using the vertices from D using Algorithm 1
while $D \neq G(V)$ **do**
  Find a v in $G(V) - M$ and $u_i \in M | E(v,u_i) \in G(E) i \in \{1,2,3,4\}$
  Determine a position for v using u_i using Algorithm 2
  $D = D \cup \{v\}$
end

Algorithm 3: Geometric Build-Up with Exact Distances

This definition makes IA conservative meaning that intervals from arithmetic operations always contain the true value, but also contain unlikely values. After a chain of operations the interval obtained has a range too wide, causing **range explosion**. Also, "cancellation" does not take place in IA, so $\bar{a} - \bar{a} \neq 0$, contributing to wider ranges.

To address some of the IA problems, Stolfi and De Figueiredo (Stolfi,1997) developed a method similar to IA, called Affine Arithmetic(AA). The main difference is that each uncertain value is identified, so "cancellation" of uncertainties occurs.In AA, a partially known value $\hat{a}$ is defined as:

$$\hat{a} = x_0 + x_1\varepsilon_1 + x_2\varepsilon_2 + \cdots + x_n\varepsilon_n \tag{2}$$

with the coefficients $x_i \in \mathscr{R}$ and ε_i is the unknown part of the affine form, lying in the range $[-1,1]$. The range of one affine form with n unknown is calculated using this formula:

$$range(\hat{a}) = 2\sum_{i=1}^{n} |x_i| \tag{3}$$

So π can be represented by $\hat{p} = 3.145 + 0.005\varepsilon_1$ with range 0.01.

In AA, addition and subtraction of affine forms and also multiplication by real values are called affine operations whose implementations are straightforward, but some arithmetic operations are not affine operations. Those operations create an extra unknown value increasing the unknown range. So,

$$\hat{a} - \hat{a} = 0 \tag{4}$$

and

$$range(\hat{a}/\hat{a}) > 0 \tag{5}$$

Like IA, AA is also conservative, leading to similar problems of range explosion.

2.2 Particle based methods

Particles can represent non-parametric distributions and to estimate the function of a distribution we simply apply the transformation on the samples (Sanjevv at al.,2002;Serfozo,2009). These methods can be very efficient and contain a distribution-based representation of the uncertainty, which carries more information than intervals. Also, the spreading range of particles is smaller than affine or interval values for the same computation. One drawback is that particles do not keep history information, so inconsistent samples can be created(see Fig. 2).

A binary operation $\otimes$ over n-dimensional real values can be used to build an operation $\otimes_p$ over a set of particles using Algorithm 4. The range of a set of particles is defined by:

$$range(p) = max(p) - min(p), \tag{6}$$

where p is a set of particles.

Particles are based on probabilistic methods, so there is no guarantee that the correct value is properly represented by particles. In the same way, the "cancellation" property is not fully achieved.

Input: Sets A and B, target size n
Output: $A \otimes_p B$
$r = \emptyset$
for $i = 0; i < n; i++$ **do**
  take a sample a from A
  take a sample b from B
  $r = r \cup \{a \otimes b\}$
end

Algorithm 4: Operation $\otimes_p$ over particles

3 THE NEW PROPOSED METHOD

We present a new hybrid method to uncertainty modeling and propagation. The unknown values are modeled both by **particles** and **affine forms** and each model is used to improve the other.

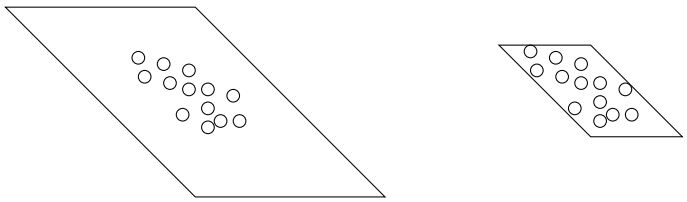
3.1 Particles Limit Affine Form

When the same unknown value is modeled by particles and affine forms, one can use the range of the particles to control the range of the affine form. There are a few ways to do this. In Algorithm 5, a unique value is used to change all dimensions of the affine form, without changing the first-order correlation (see Fig. 1).

The range of particles is usually smaller than the range of affine form for two main reasons:

Sampling Final extremal value is unlikely in a chain of samples and when the problem allows, one can use Gaussian distribution in order to produce smaller intervals

Filters Particles can be easily filtered and statistical filters can remove unlikely particles from the sample. Also, specific filters for the problem can easily remove bad particles.



a) b)

Figure 1: Affine Form range reduced by particles.

3.2 Affine Forms create Consistent Particles

When we have sets of particles modeling unknown values, sampling each set can create an incorrect final result. In Figure 2, we show two extreme particles (connected with a dashed line), where each one is correct, but together they create an incorrect pair - in this example the pair is too distant. Instead of sampling particles, we can sample in ε-space (see Algorithm 6) and use the same sampled values to create a consistent exact pair(solid lines).

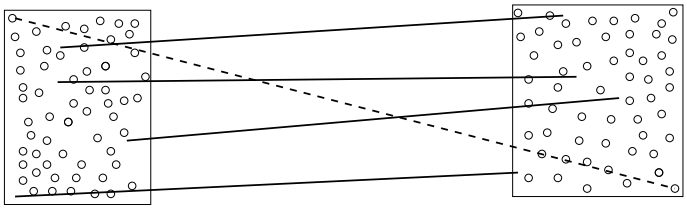


Figure 2: Consistent points created by sampling ε-space.

Input: n dimensional affine form $\hat{a}$ and particles P
Output: $\hat{a}$ modified
$k = 0$
for $i = 1; i \leq n; i++$ **do**
  $r_a = range(\hat{a}_i)$
  $r_p = range(P_i)$
  $k = max(k, r_a/r_p)$
end
for $i = 1; i < n; i++$ **do**
  $\hat{a}_i = \hat{a}_i.x_0 + k * (\hat{a}_i.x_1 + \hat{a}_i.x_2 + \dots)$
end

Algorithm 5: Control Affine

Input: Vector a of affine forms,n index of last unknown
Output: Vector r of floats
for $i = 1; i \leq n; i++$ **do**
  $s_i = uniform(-1, 1)$
end
for $i = 1; i \leq size(a); i++$ **do**
  $r_i = a_i.x_0 + \sum_{k=1}^{n} a_i.x_k * s_k$
end

Algorithm 6: Sample ε-space

4 COMPUTATIONAL EXPERIMENTS

We evaluated the applicability of our method for uncertainty propagation by applying it in the protein reconstruction problem. First, we modified algorithms 1, 2 and 3 in order to work with affine forms and particles. Each arithmetic operation was replaced from a *float type* to *affine form type* and to *particle type*. To control the uncertainty propagation, every affine point is limited (Algorithm 5) by particles and the particle resampling (Alg. 6) is performed. After the calculation of all atomic positions, a final structure is obtained by sampling the affine form points.

For each experiment we created an *artificial backbone* (Lavor,2006) in order to easily simulate NMR data:

- Covalent bonds and angles are constant, so for atoms separated by one or two covalent bonds we used the exact distance,
- For each pair of atoms with distance in 2Å to 6Å, we calculated the the real distance d and an affine form

$$\hat{d} = \lfloor d \rfloor + 0.5 + 0.5\varepsilon_k \tag{7}$$

  with a new k for each pair.

The algorithm was implemented using Python language and each experiment took about 1 hour in Linux system running on a Intel Core 2 Duo T6500 with 4GB of RAM. For each molecule with 10, 20, 40, 60, 80, and 100 atoms, we randomly generate 5 instances. The results were summarized in Tab. 1, where the root mean square deviation (RMSD) value, which compares the original structure with the solution given by our method, and the percentage (in parenthesis) of satisfied distance constraints are given. RMSD values less than 3Å mean that structures are very similar. Figure 3 shows in red the original backbone obtained by Alg. 7.

5 CONCLUSIONS

In this paper we presented a hybrid method that controls uncertainty propagation. We showed how to use particles to control affine forms and how to use a set of affine forms to create a set of consistent particles.

Using this method, we calculated artificial backbones with a small RMSD values, compared to the original ones. Future works include the use of other protein structural information to improve the quality of the solutions and expand the size of the protein calculated, and also the use of this method in other problems.

6 ACKNOWLEDGEMENTS

We thank financial support from CNPq, FAPESP and CAPES.

7 REFERENCES

Dong, Q and Wu,Z., 2002, "A linear-time algorithm for solving the molecular distance geometry problem with exact inter-atomic distances" Journal of Global Optimization, 22(1):365–375.

Input: G with imprecise distances in edges
Output: G with position in vertices
D = findClique4(G)
create particles and an affine form base using the vertices from D
while $D \neq G(V)$ **do**
  Find a v in $G(V) - M$ and $u_i \in M | E(v,u_i) \in G(E) i \in \{1,2,3,4\}$
  Determine a position for v using u_i both for particles and affine forms
  foreach $(v,w) \in G(E), w \in D, w \neq u_i, i \in \{1,2,3,4\}$ **do**
    Use the distance in (v,w) to filter inconsistent particles of v
  end
  Control affine form of v using particles
  $D = D \cup \{v\}$
  Resample particles using Algorithm 6
end

Algorithm 7: Geometric Build-Up with imprecise distances

Table 1: RMSD values and the percentages (in parenthesis) of satisfied distance constraints

Experiment	Number of Atoms					
	10	20	40	60	80	100
1	0.38 (100)	0.40 (93)	2.41 (89)	2.96 (72)	2.53 (90)	2.73 (76)
2	0.31 (97)	0.30 (93)	0.24 (94)	0.64 (90)	1.08 (85)	1.91 (81)
3	0.56 (98)	0.74 (82)	1.83 (84)	2.63 (89)	3.30 (98)	4.85 (96)
4	0.75 (97)	1.95 (97)	3.08 (80)	4.21 (80)	3.69 (71)	3.51 (74)
5	0.28 (100)	0.22 (93)	1.22 (95)	2.45 (84)	2.66 (89)	3.40 (79)

Güntert, P., 1998, "Structure calculation of biological macromolecules from nmr data" Quarterly Reviews of Biophysics, 31(02):145–237.

Lavor, C., 2006, "On generating instances for the molecular distance geometry problem" In L.Liberti et al.(ed), Global Optimization: Nonconvex Optimization and Its Applications, 84:405–414.

Moore, R.E., 1966, "Interval analysis. Prentice-Hall series in automatic computation" Prentice-Hall.

Serfozo, R., 2009, "Basics of Applied Stochastic Processes" Probability and Its Applications.

Stolfi, J. and De Figueiredo, L., 1997, "Self-validated numerical methods and applications" Brazilian Mathematics Colloquium monographs. IMPA/CNPq.

Wüthrich, K., 1986, "NMR of Proteins and Nucleic Acids" John Wiley & Sons.

RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.

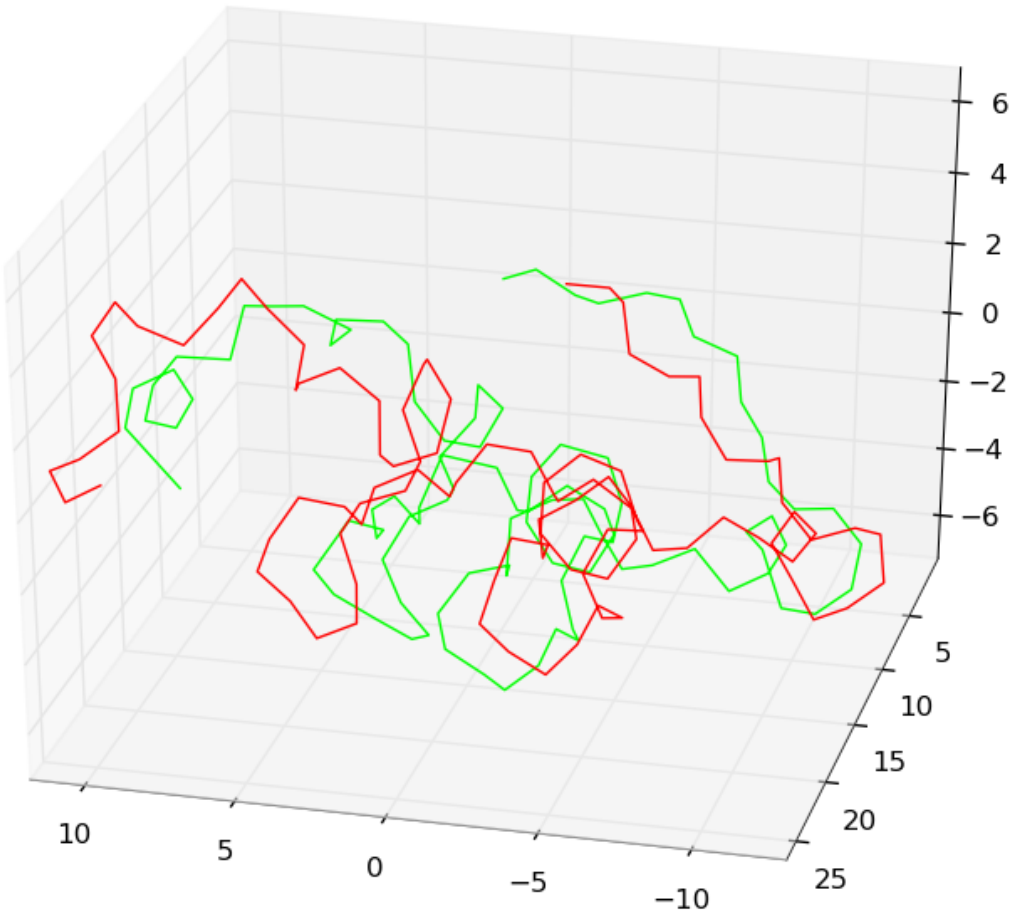


Figure 3: In red, original backbone and in green the final backbone calculated using Alg. 7