Protein Structure Reconstruction with Data Uncertainties

Ivan Sendin¹, Siome K. Goldenstein², Carlile Lavor³

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 a 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.

```
Algorithm 1: Create Geometric Base

Input: Distances d₁2, d₁₃, d₁₄, d₂₃, d₂₄, d₃₄
Output: Points (xᵢ, yᵢ, zᵢ) for i ∈ {1, 2, 3, 4}

(x₁, y₁, z₁) = (0, 0, 0)
(x₂, y₂, z₂) = (d₁₂, 0, 0)
x₃ = (d₁₃² + d₁₄²) / (d₁₂ + 2) + (d₁₂ / 2)
y₃ = √d₁₃² - x₃²
z₃ = 0
x₄ = (d₁₄² - d₂₄²) / (d₁₂ + 2) + d₁₂ / 2
y₄ = d₂₄² - d₃₄² - (x₄ - d₁₂)² + (x₄ - x₃)²
z₄ = d₁₄² - x₄² - y₄²
```

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 \( \pi \) can be represented by \( \bar{\pi} = [3.14, 3.15] \) with range 0.01.

A general IA operation \( \otimes \) is defined as:

\[
\bar{a} \otimes \bar{b} = \{a \otimes b | a \in \bar{a}, b \in \bar{b}\}
\]
Algorithm 4. The range of a set of particles is defined by:

Keep history information, so inconsistent samples can be created (see Fig. 2).

A distribution-based representation of the uncertainty, which carries more information than intervals. Also, the spreading transformation on the samples (Sanjeev at al., 2002; Serfozo, 2009). These methods can be very efficient and contain a

2.2 Particle based methods

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

Particles can represent non-parametric distributions and to estimate the function of a distribution we simply apply the transformation on the samples (Sanjeev 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:

\[
\text{range}(p) = \max(p) - \min(p),
\]
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.

\[
\text{Input: Sets } A \text{ and } B, \text{ target size } n \\
\text{Output: } A \otimes_p B \\

r = \emptyset \\
\text{for } i = 0; i < n; i++ \text{ do} \\
\quad \text{take a sample } a \text{ from } A \\
\quad \text{take a sample } b \text{ from } B \\
\quad r = r \cup \{a \otimes b\} \\
\text{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.

\[\text{Figure 1: Affine Form range reduced by particles.}\]

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 \( \varepsilon \)-space (see Algorithm 6) and use the same sampled values to create a consistent exact pair (solid lines).

\[\text{Figure 2: Consistent points created by sampling } \varepsilon\text{-space.}\]
Input: $n$ dimensional affine form $\hat{a}$ and particles $P$
Output: $\hat{a}$ modified

$k = 0$

for $i = 1; i \leq n; i++$
do

$\hat{r}_a = \text{range}(\hat{a}_i)$
$\hat{r}_p = \text{range}(P_i)$
$k = \max(k, \hat{r}_a / \hat{r}_p)$

end

for $i = 1; i < n; i++$
do

$\hat{a}_i = \hat{a}_i.x_0 + k \times (\hat{a}_i.x_1 + \hat{a}_i.x_2 + \ldots)$

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

$s_i = \text{uniform}(-1, 1)$

end

for $i = 1; i \leq \text{size}(a); i++$
do

$r_i = a_i.x_0 + \sum_{k=1}^{n} a_i.x_k \times s_k$

end

Algorithm 6: Sample $\epsilon$-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 2 Å to 6 Å, we calculated the real distance $d$ and an affine form

\[ \hat{d} = \lfloor d \rfloor + 0.5 + 0.5 \epsilon_k \]  \hspace{1cm} (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 an 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 3 Å 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

\textbf{Input:} $G$ with imprecise distances in edges  
\textbf{Output:} $G$ with position in vertices  

\[ D = \text{findClique4}(G) \]

create particles and an affine form base using the vertices from $D$

\textbf{while} $D \neq G(V)$ \textbf{do}

\hspace{1em} Find a $v$ in $G(V) - M$ and $u_i \in M | E(v, u_i) \in G(E) i \in \{1, 2, 3, 4\}$

\hspace{1em} Determine a position for $v$ using $u_i$ both for particles and affine forms

\hspace{1em} \textbf{foreach} $(v, w) \in G(E), w \in D, w \neq u_i, i \in \{1, 2, 3, 4\}$ \textbf{do}

\hspace{2em} Use the distance in $(v, w)$ to filter inconsistent particles of $v$

\hspace{1em} \textbf{end}

\hspace{1em} Control affine form of $v$ using particles

\hspace{1em} $D = D \cup \{v\}$

\hspace{1em} Resample particles using Algorithm 6

\hspace{1em} \textbf{end}

\textbf{Algorithm 7:} Geometric Build-Up with imprecise distances

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


\textbf{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