CT-Image Segmentation using a Markov Chain Monte Carlo Approach

Lukas Valatka, Patrick Zumsteg

June 18, 2018

1 Introduction

The goal of this project was to create a 3D-Mesh to segment an image. We have been given five 3D CT-images of femur bones, as well as an Active Shape Model (ASM) learned from real femurs. We have solved this task using the Markov Chain Monte Carlo (MCMC) method. We have constructed several Markov Chains to fit both pose and shape of the given Model to our CT-images.

2 Methods

We have tried several implementations of an MCMC-pipeline. During our working process, we've come up with three major versions. First, we used two chains, one for pose- and one for shape fitting. We improved it later by using manually placed landmarks as guides. Lastly, we conducted shape fitting on some selected Principal Components of the Model. Throughout the process, we have also tuned the parameters for all the generators.

Proposal Generators

All perturbations are drawn from a multivariate normal distribution with zero mean and identity covariance matrix with configurable diagonal values.

- **Pose Update Proposal**. Returns a rotation and translation step update given the current coefficients. We have used this one for pose fitting.
- **Shape Update Proposal**. Returns an update to the Statistical Shape Model coefficients. In our case, this includes an update for all Principal Components.
- We have additionally used **Shape Update Proposals for Principal Components**. It returns an update for a singular Principal Component (i.e. for a single value of the model coefficients vector).

Evaluators

- **Correspondence evaluator**. We have used this one for landmark-based pose fitting. Higher probabilities are returned if all femur landmarks are closer to corresponding target landmarks.
- **Intensity Evaluator**. We have used this evaluator to evaluate the shape and the pose. It evaluates how likely a mesh fits a target image based on learned pixel intensities (provided by the instructors).
- **Shape Prior Evaluator**. We have used this evaluator to evaluate how likely an instance of model is.

2.1 The Initial Approach

Our initial chain combination looked like this:

1. **Pose fitting**. The chain used a rotation/translation mixture proposal (0.4 each and 0.1 each with double variance). A combination of Intensity and Shape Prior Evaluators were used.
2. After this, we placed a loop, where each iteration contains:
   - A **shape-fitting** chain, followed by
   - a short **pose-fitting** chain.

With each iteration of the loop, we decrease the variance of the proposals, as the acceptance rate for any given variance starts deteriorating after some time.

We chose to use the shape fitting followed by the pose fitting structure because there might not be necessarily an overall "best" pose: rather, the best pose fit depends on the bone's shape, which means that a previously good pose-fitting might have become worse after shape-fitting.

We noticed that this pose fitting fails to escape local minima, which is why we decided to use landmarks.

2.2 Landmark-based Pose Fitting

This approach differs from previous one, that instead of evaluating the proposals of the *initial* pose fitting using profile points, we evaluated them using the distance between landmarks, which we have placed manually.

Landmarks were placed on the head (the fovea capitis, specifically), the greater trochanter, and two on the condyles of the ASM and each target image.

The loop was kept unchanged; as our landmarks have not been created by the experts, they may not be very precise, thus we don't rely on them for the more fine-grained pose fitting.

2.3 Component-wise Shape Fitting

We still had problems fitting our Model to 10.nii, where the bone had an unusually small head. Thus, we inserted an additional chain into the pipeline, between the initial pose fitting and the loop, which predominantly proposes perturbations of the third to seventh component (i.e. the ones influencing the head's size and shape).

3 Results

We believe our results to be satisfactory. We significantly improved the performance of our initial implementation with the Landmark-based approach. Our final results evaluate between -7934 to -14383 (using prior and intensity evaluators combination). This is still worse than the average evaluations of the test data segmentations (-6377), however, to our untrained eyes, the results appear to be decent.

3.1 Experiment setup

While we have been running our programs on our own local machines, we quickly realized that we would need some kind of force multiplier, to test different parameter assignments and to obtain better results by running the same program several times for each target (as we never get the exact same result twice, doing multiple runs concurrently and choosing the best result is a valid strategy). For this, we used Amazon's *m4.2xlarge* instances as well as the university's *miniHPC*.

3.2 Initial approach

Despite having to run the program many times, with different parameters, this approach always delivered poor results. The initial pose fitting of the bone is bad, and the following shape fitting barely yields any improvement. Shown here is the result after running our initial approach with the following parameters:

- **Loop iterations:** 10, with one shape and pose fitting chain each.
- **Chain length:** Initial pose fitting: 1000, shape fitting: 1000/iteration, fine pose fitting: 250/iteration.
- **Variance:** Rotation/Translation: 0.001/0.005, Shape: 0.01. In the loop, variances decreased linearly.

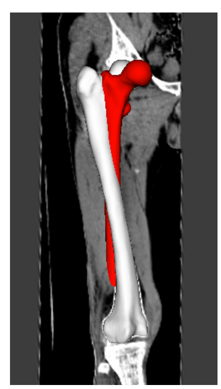

Figure 1: Red: final fit, white: ground truth. Done on image 4.nii

Acceptance rates were very low, fluctuating between 0.001 and zero. Using our likelihood based evaluator, the red mesh received a score of -348849. This approach is clearly unsatisfactory.

3.3 Landmark-based approach

Using landmarks, we achieved vastly improved results. Initial pose fitting converges quickly, and the acceptance rates improved as well, up to 10%. We ran the program five times for each target, taking the best result. We used the following parameters:

- **Loop iterations:** 3, with one shape and pose fitting chain each.
- **Chain length:** Initial pose fitting: 1500, shape fitting: 3000/iteration, fine pose fitting: 500/iteration.
- **Variance:** Rotation/Translation: 0.0001/0.001, Shape: 0.01. In the loop, variances decreased linearly.

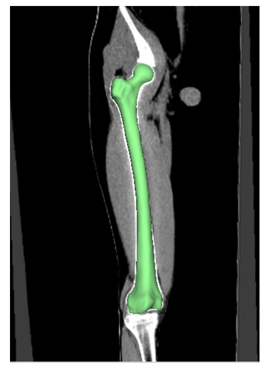

Figure 2: Final segmentation of 9.nii. Likelihood score: -7934.

We initially had trouble finding a fit for 10.nii using this approach, which was why we developed the component-wise fitting. It turned out, however, that 2/5 instances running with 10.nii managed to escape the local minimum in which the previous runs got stuck, and yielded a result of similar quality to the other segmentations. This scenario is likely in MCMC as the starting point in the distribution varied with each run, and this effect has an impact on the convergence rate [1].

3.4 Component-wise fitting

Adding a shape fitting chain, focusing on the third to seventh Principal Component has been surprisingly ineffective. We have not managed to obtain any better results than with the Landmark-based approach. It should be mentioned, though, that we have not had enough time to run the Component-wise fitting with the same chain sizes as with the Landmark-based approach. As to this, the landmark-based architecture was chosen as final one to do the fitting.

4 Discussion

Without landmarks, the initial pose fitting never seems to manage to obtain a good fit if the Model is not already well aligned to the image; instead, there seems to be a local minimum where the rotation is roughly correct, but only the outer wall of the bone and the image intersect. Moving the Model in either direction would then reduce the overlap, and it takes too long to reach a proper fit.

Introducing landmarks solves this problem. While placing these requires a human operator, there is no need for an expert: the landmarks we placed ourselves, without any medical expertise, performed quite decently.

Also, performing shape-fitting only on some selected Components did yield some increase in acceptance rates for short initial shape fitting periods, getting up to 20% in some cases, yet dropping later to normal and even worse acceptance rates.

Which leads us to the main problem of our implementations: The acceptance rates, especially for shape-fitting, were quite low. In the lecture, we have been encouraged to strive for acceptance rates of up to 60%, which we clearly never achieved.

4.1 Improvements

Parameter tuning: For example, we have not modified the uncertainty parameter for the Correspondence evaluator. Also, we had to stop tuning the variances for the Landmark-based pose fitting chain, and the mixture proposal's expression probabilities due to a lack of time.
Generating informed proposals: Currently, our generators propose Gaussian-distributed perturbations. We believe that it is possible to increase the proposals' acceptance rate by making them more informed, i.e. by generating the perturbations from a learned distribution. We have considered the Adaptive Metropolis Algorithm[2] as a candidate, but didn't manage to implement it due to time constraints.
Performance: AWS has yielded vast performance improvements, since we can run large numbers of instances simultaneously. In addition to that, the proposal evaluators could be parallelized (e.g., on GPUs) to improve performance, as they sum up independent likelihoods.

4.2 Lessons learned

While any Markov Chain eventually converges, the time it takes heavily depends on proper parameter choice. Acceptance rates quickly deteriorate for any given variance, which is why we iteratively decrease the variances.

Secondly, choosing meaningful proposal generators may yield quicker convergences than mere parameter tuning, as seen with the Landmark-based approach.

References

[1] John D. Cook. Markov Chains Don't Converge. `https://www.johndcook.com/blog/2011/08/10/markov-chains-dont-converge/`

[2] Haario, Saksman, Tamminen. An Adaptive Metropolis Algorithm. `https://projecteuclid.org/euclid.bj/1080222083`