CHAMP: Changepoint Detection Using Approximate Model Parameters

Scott Niekum[1,2] Sarah Osentoski[3] Christopher G. Atkeson[1]

Andrew G. Barto[2]

Abstract

We introduce CHAMP, an algorithm for online Bayesian changepoint detection in settings where it is difficult or undesirable to integrate over the parameters of candidate models. Rather than requiring integration of the parameters of candidate models as in several other Bayesian approaches, we require only the ability to fit model parameters to data segments. This approach greatly simplifies the use of Bayesian changepoint detection, allows it to be used with many more types of models, and improves performance when detecting parameter changes within a single model. Experimental analysis compares CHAMP to another state-of-the-art online Bayesian changepoint detection method.

1 Introduction

Many practical applications in statistics require detecting changes in the parameters and models that generate observed data. Commonly cited examples include detecting changes in stock market behavior [4], well drilling data [5], and DNA segmentation [3]. Bayesian changepoint detection methods offer notable advantages over their frequentist counterparts, including the ability to generate a full posterior distribution over changepoint locations and offering a natural way to incorporate prior knowledge. However, many Bayesian approaches to changepoint detection require parameters of the candidate models to be marginalized [1, 5]. This can be problematic in two ways. First, if the model is in a difficult form to analytically integrate, and the parameter space is too high-dimensional to numerically integrate, such methods are impractical. Second, in some cases, parameter integration can lead to an inability to detect changes in parameters within a single model.

We introduce an algorithm for online Bayesian changepoint detection in settings where it is difficult or undesirable to integrate over the parameters of candidate models. Building on the work of Fearnhead and Liu [5], we show that with some modifications, approximate online Bayesian changepoint detection can be performed using estimates of the maximum likelihood parameters

[1]Scott Niekum and Christopher G. Atkeson are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA sniekum@cmu.edu

[2]Scott Niekum and Andrew G. Barto are with the School of Computer Science, University of Massachusetts Amherst, Amherst, MA 01020, USA

[3]Sarah Osentoski is with Robert Bosch Research and Technology Center, Palo Alto, CA 94304, USA

for each segment—for example, via regression or a sample consensus method. Our modifications also remove a significant restriction on model definition when detecting parameter changes within a single model. We call this new algorithm CHAMP (**Ch**angepoint detection using **A**pproximate **M**odel **P**arameters). Finally, the capabilities of CHAMP are experimentally verified using artificially generated data and are compared to those of Fearnhead and Liu [5].

2 Related work

Hidden Markov Models (HMMs) are largely the *de facto* tool of choice when analyzing time series data, but the standard HMM formulation has several undesirable properties. The number of hidden states must be known ahead of time (or chosen using model selection), inference is often costly and subject to local minima when algorithms like Expectation-Maximization are used, and segment lengths are inherently geometrically distributed. Nonparametric Bayesian models like the HDP-HMM [6] relax some of these conditions, but incur a new set of challenges, including the need for MCMC-based inference. In settings where the primary objective is to identify model changes without considering shared hidden states across segments, changepoint detection methods can be a more appropriate algorithmic choice.

Frequentist approaches to changepoint detection and piecewise regression include methods such as PELT [7] that can perform exact inference in linear time over a wide range of cost functions. Alternately, Chopin [4] introduces a Bayesian changepoint detection algorithm that uses a recursive filtering approach, but requires MCMC steps for parameter inference. Building on this work, Fearnhead and Liu present an approximate Bayesian changepoint detection algorithm [5] that can perform online inference efficiently, finding the distribution of locations of the changepoints and the model parameters of each segment using computational time linear in the number of observations. However, this work requires that model parameters can be marginalized, as does a similar approach by Adams and MacKay [1]. Other approaches to multiple model fitting have been proposed, such as MultiRANSAC [8], but cannot take advantage of the time-series nature of our setting.

3 Changepoint Detection using Approximate Model Parameters

3.1 Online MAP Changepoint Detection

First, we describe the online MAP (maximum *a posteriori*) changepoint detection model of Fearnhead and Liu [5]. Assume we have time-series observations $\mathbf{y_{1:n}} = (y_1, y_2, \ldots, y_n)$ and a set of candidate models Q. Our goal is to infer the MAP set of changepoints times $\tau_1, \tau_2, \ldots, \tau_m$, with $\tau_0 = 0$ and $\tau_{m+1} = n$, giving us $m + 1$ segments. Thus, the i^{th} segment consists of observations $\mathbf{y}_{\tau_i+\mathbf{1}:\tau_{i+1}}$ and has an associated model $q_i \in Q$ with parameters θ_i.

We assume that data after a changepoint is independent of data prior to that changepoint, and we model the changepoint positions as a Markov chain in which the transition probabilities are defined by the time since the last changepoint:

$$p(\tau_{i+1} = t | \tau_i = s) = g(t - s), \tag{1}$$

where $g(\cdot)$ is a probability distribution over time and $G(\cdot)$ is its cumulative distribution function.

Given a segment from time s to t and a model q, define the model evidence for that segment as:

$$L(s,t,q) = p(\mathbf{y_{s+1:t}}|q) = \int p(\mathbf{y_{s+1:t}}|q,\theta)p(\theta)d\theta. \tag{2}$$

It can be shown how the standard Bayesian filtering recursions and an online Viterbi algorithm can be used to efficiently estimate C_t, the distribution over the position of the first changepoint prior to time t [5]. Define $\mathcal{E}_j$ as the event that given a changepoint at time j, the MAP choice of changepoints has occurred prior to time j and define:

$$P_t(j,q) = p(C_t = j, q, \mathcal{E}_j, \mathbf{y_{1:t}}) \tag{3}$$
$$P_t^{MAP} = p(\text{Changepoint at } t, \mathcal{E}_t, \mathbf{y_{1:t}}). \tag{4}$$

This results in the equations:

$$P_t(j,q) = (1 - G(t-j-1))L(j,t,q)p(q)P_j^{MAP} \tag{5}$$

$$P_t^{MAP} = \max_{j,q}\left[\frac{g(t-j)}{1-G(t-j-1)}P_t(j,q)\right]. \tag{6}$$

At any point, the Viterbi path can be recovered by finding the (j,q) values that maximize P_t^{MAP}. This process can then be repeated for the values that maximize P_j^{MAP}, until time zero is reached. A straightforward alternate formulation [5] allows for the simulation of the full posterior distribution of changepoint locations, though in this work, we focus only on the MAP changepoints.

The algorithm is fully online, but requires $O(n)$ computation at each time step, since $P_t(j,q)$ values must be calculated for all $j < t$. To reduce computation time to a constant, ideas from particle filtering can be leveraged to keep only a constant number of particles, M, at each time step, each of which represent a support point in the approximate density $p(C_t = j, \mathbf{y_{1:t}})$. At each time step, if the number of particles exceeds M, stratified optimal resampling [5] can be used to choose which particles to keep in a manner that minimizes the KL divergence from the true distribution in expectation.

3.2 CHAMP

The model evidence shown in Equation 2 requires that the parameters of the underlying model can be marginalized. This requires the use of either conjugate priors, allowing analytical integration, or a low dimensional parameter space that can be efficiently numerically integrated. However, many models do not fit into either of these categories, requiring an alternate solution for when only point-estimates of model parameters are available. Furthermore, marginalization of the model parameters can prevent the detection of changepoints in which the model stays the same, but the parameters of the model change. This can happen when the model being considered treats each data point as independent; since the likelihood can be factorized into a product and the model parameters are marginalized, the likelihood function shows no preference for multiple segments in the case of a parameter change within a model.

For example, imagine generating a set of independent data points under model q with parameters θ_{ab} for $\mathbf{y_{a:b}}$ and parameters θ_{bc} for $\mathbf{y_{b:c}}$. Despite the different underlying parameters for each

segment,

$$\int p(\mathbf{y_{a:c}}|q,\theta)p(\theta)d\theta$$
$$= \int p(\mathbf{y_{a:b}}|q,\theta)p(\theta)d\theta \int p(\mathbf{y_{b:c}}|q,\theta)p(\theta)d\theta.$$

Notice that this is not the case for some models, such as the autoregressive models originally used by Fearnhead and Liu [5].

We present CHAMP (**Ch**angepoint detection using **A**pproximate **M**odel **P**arameters)—a modified version of Fearnhead and Liu's changepoint algorithm that allows the use of models of any form (with independent emissions or otherwise), in which parameter estimates are available via means such as maximum likelihood fit, MCMC, or sample consensus methods. We propose three primary changes to best accommodate this new setting.

3.2.1 Approximate model evidence

The Bayesian Information Criterion (BIC) is a well-known approximation to integrated model evidence [2] that provides a principled penalty against more complex models by assuming a Gaussian posterior distribution of parameters around the estimated parameter value $\hat{\theta}$. Using the BIC, the model evidence can be approximated as:

$$\ln L(s,t,q) \approx \ln p(\mathbf{y_{s+1:t}}|q,\hat{\theta}) - \frac{1}{2}k_q \ln(t-s), \tag{7}$$

where k_q is the number of free parameters of model q. This approximation allows us to avoid directly evaluating the model evidence integral.

3.2.2 Minimum segment length

Since we are now assuming that parameter estimates come from some type of model fitting procedure, the quantity $L(s,t,q)$ is no longer well-defined for all $t > s$. Instead, each model q has a minimum value of $t-s$ for which the model is defined. For example, a line requires a minimum of two points to define, whereas a plane requires three. As a simplification, and to prevent overfitting, some sufficient minimum segment length α can be chosen for all models. This requires three changes: changepoints can only begin to be considered at time $t = 2\alpha$ (when a changepoint in the center would create two equal halves of length α), $P_t(j,q)$ must only be calculated for values of $t - j > \alpha$, and the choice of a segment length distribution $g(\cdot)$ must be reconsidered.

Fearnhead and Liu suggest the use of a geometric length distribution [5], as it arises naturally from a constant probability of seeing a changepoint at each time step. However, it is a monotonically decreasing distribution with a mode of 1 that favors shorter segments, which can lead to overfitting, especially in a setting with fitted model parameters. As an alternative, Chopin [4] suggests using a uniform prior over limited support to ensure it is well-defined. However, this artificially places a hard limit on segment lengths, regardless of the data. We propose the use of a truncated normal distribution, which enforces a minimum segment length naturally, has easily interpretable parameters, and is less prone to overfitting:

$$g(t) = \frac{\frac{1}{\sigma}\phi\left(\frac{t-\mu}{\sigma}\right)}{1-\Phi\left(\frac{\alpha-\mu}{\sigma}\right)} \tag{8}$$

$$G(t) = \Phi\left(\frac{t-\mu}{\sigma}\right) - \Phi\left(\frac{\alpha-\mu}{\sigma}\right), \tag{9}$$

where ϕ is the standard normal PDF, Φ is its CDF, and α is the minimum segment length. Since the mode of the distribution is close to the mean (or identical if no truncation occurs), segment lengths are pushed toward the mean, instead of being pushed toward 1. By using a broad value of σ, we can support a wide range of segments lengths, while leaving μ as a adjustable parameter that can be tuned if over-segmentation or under-segmentation is an issue. Alternatively, if specific prior knowledge about segment length is known, μ can be set accordingly with a more narrow value of σ to restrict segment length appropriately.

3.2.3 Particle definition

Finally, since model fitting can be an expensive procedure, we suggest a slight revision of the definition of a particle from that of Fearnhead and Liu. Previously, each particle represented a support point to approximate the joint distribution $p(C_t = j, \mathbf{y_{1:t}})$, marginalizing over models q. To potentially save on the number of required model fits, we suggest each particle also include the model, so that our approximated distribution is $p(C_t = j, q, \mathbf{y_{1:t}})$, allowing particular models to be selectively discarded at each time step. This also prevents us from overlooking the possibility of a changepoint at a given time step when only one model is a reasonable fit and the others are very poor.

Figure 1 provides pseudocode for CHAMP. Additionally, an open-source implementation of CHAMP as a ROS service is available online [1].

4 Experiments

4.1 1D Gaussian: zero mean, parameterized variance

First, we present an experiment to demonstrate the ability of CHAMP to reliably detect changepoints using maximum likelihood parameter estimates for models with independent emissions. Five segments of data were generated (of lengths 40, 60, 30, 50, and 70) by making draws from a zero-mean Gaussian distribution with parameterized variance ($\sigma = 2.0$, 1.0, 3.0, 1.5, and 2.5), shown in the left panel of Figure 2. CHAMP was then used to try to recover the locations of the changepoints with the following parameters: a truncated Gaussian length distribution with $\mu = 50$ and $\sigma = 10$, a minimum segment length of 2, and 100 maximum particles. We compared this analysis with that of the original Fearnhead and Liu algorithm under the same parameters (where applicable) by integrating the likelihood function with a conjugate Gamma prior on the precision τ, and setting hyperparameters $a = 4.0$, $b = 0.5$:

$$\begin{aligned} L(s,t,q) &= \int_0^\infty \prod_{i=s+1}^{t} \mathcal{N}(x_i|\mu=0,\tau^{-1}) Gam(\tau|a,b) d\tau \\ &= \Gamma(a+1/2)\frac{b^a}{\Gamma(a)}\frac{1}{\sqrt{2\pi}}\left(b+\frac{x^2}{2}\right)^{-a-1/2}. \end{aligned}$$

[1] `http://wiki.ros.org/changepoint`

Input: Observations $\mathbf{y_{1:n}}$, candidate models $q_1, \ldots, q_r$, prior distribution $\pi(q)$, minimum segment length α, and maximum number of particles M.
Output: Viterbi path of changepoint times and models

// Initialize data structures
max_path, prev_queue, particles = {}
prev_queue.push(1/r)
for $i = 1 : r$ **do**
  new_p = newParticle(pos = 0, model = q_i, prev_MAP = 1/r)
  particles.add(new_p)
end for

// Do for all incoming data, starting at time α
for $t = \alpha : n$ **do**

// Add new particles
  if $t >= 2\alpha$ **then**
    pref = prev_queue.pop() // P_t^{MAP} α steps ago
    for $i = 1 : r$ **do**
      new_p = newParticle(pos = t$-\alpha$, model = q_i, prev_MAP = prev)
      particles.add(new_p)
    end for
  end if

// Compute fit probabilities for all particles
  for $p \in$ *particles* **do**
    p_tjq = $L(\text{p.pos}, t, q) \cdot \pi(q) \cdot$ p.prev_MAP
    p.MAP = $g(t - \text{p.pos}) \cdot$ p_tjq
  end for

// Find max particle and update Viterbi path
  max_p = $\max_p$ p.MAP
  prev_queue.push(max_p.MAP)
  max_path.add(j = max_p.pos, q = max_p.model)

// Resample if too many particles
  if particles.length > M **then**
    particles = stratOptResample(particles, M)
  end if

end for

// Recover the Viterbi path
v_path = {}
curr_cp = n
while curr_cp > 0 **do**
  $\langle j, q \rangle$ = max_path[curr_cp - α]
  v_path.add(start = j, end = curr_cp, model = q)
  curr_cp = j
end while
return v_path

Figure 1: CHAMP

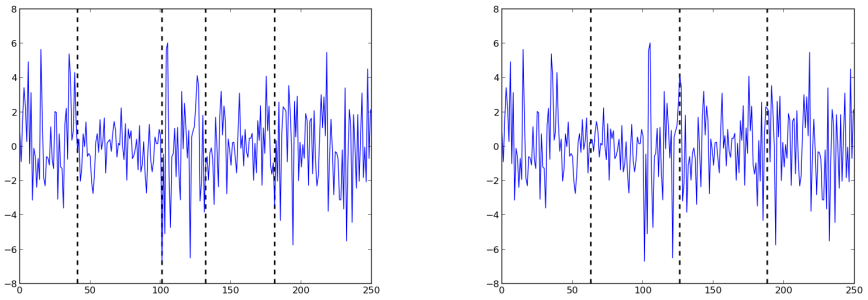

Figure 2: Five segments of mean-zero Gaussian data with changing variance: an accurate segmentation by CHAMP (left) and an inaccurate segmentation using Fearnhead and Liu's original algorithm (right).

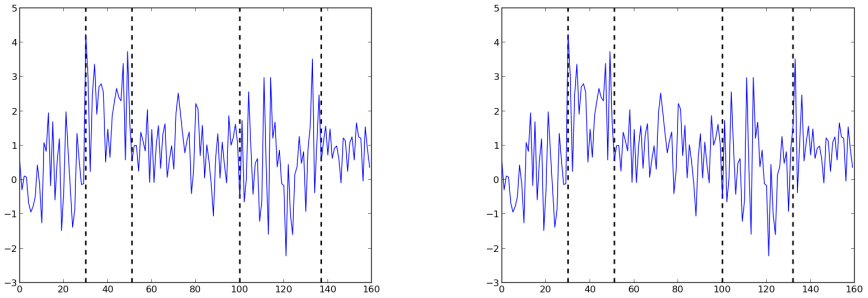

Figure 3: Five segments of discrete-mean Gaussian data with changing variance: an accurate segmentation by both CHAMP (left) and Fearnhead and Liu's original algorithm (right). Note that the mean changes with every segment.

The center panel of Figure 2 shows a segmentation of the data by CHAMP that correctly divides the data into 5 segments. Identical changepoint locations were found in all 100 runs of CHAMP (the stratified optimal resampling step can introduce stochasticity), and were all found to be within 2 data points of the true changepoint locations. The right panel of Figure 2 shows the failure of Fearnhead and Liu's algorithm to properly detect parameter switches within the single model, since the data emissions were independent, as discussed in Section 3.2. This result held across a wide range of parameter settings, as it is a fundamental deficit in the original algorithm.

4.2 1D Gaussian: discretized mean, parameterized variance

This experiment is identical to the previous example, with one important change—we now use 3 different models, each with a different static mean (0.0, 1.0, and 2.0). If the mean were instead added as another continuous parameter of a single model, the Fearnhead and Liu algorithm would have the same problem as before; changes would not be detected, since emissions are independent and parameters are integrated out. However, by using separate models with different, static means, we can compare CHAMP directly to the algorithm of Fearnhead and Liu, since it can detect changes between the models.

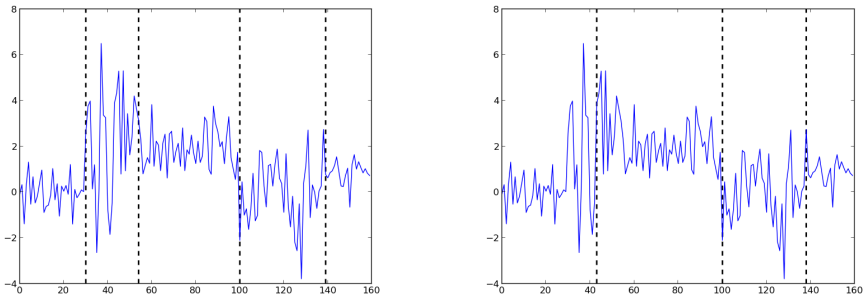

Figure 4: Five segments of discrete-mean Gaussian data with changing variance: an accurate segmentation by CHAMP (left) and an inaccurate segmentation using Fearnhead and Liu's original algorithm (right) that misses a change in variance without a change in mean.

Again, five segments of data were generated (of lengths 30, 20, 50, 40, and 20) in which both mean and variance changed each time (0.0, 1.0; 2.0, 1.8; 1.0, 0.7; 0.0, 1.2; and 1.0, 0.5). Figure 3 shows nearly identical accurate segmentations from both CHAMP and Fearnhead and Liu's algorithm. Both methods produced highly consistent segmentations and detected the correct number of changepoints every time during 100 runs. CHAMP was not only competitive with Fearnhead and Liu's algorithm despite not integrating out model parameters, but actually performed slightly better. CHAMP's changepoints were, on average, a distance of 1.215 time steps from the true changepoints, whereas Fearnhead and Liu's were a distance 2.5 on average.

Finally, we demonstrate how under this same model, a change in variance without a change in mean cannot be detected by Fearnhead and Liu's algorithm. Again, five segments of data were generated (of lengths 30, 30, 40, 40, and 20), but this time, there is one instance where the variance changes, but the mean stays the same (0.0, 0.7; 2.0, 2.0; 2.0, 0.7; 0.0, 1.2; and 1.0, 0.5). Figure 4 shows that CHAMP was able to accurately detect all the changes, while Fearnhead and Liu's algorithm misses the changepoint when only variance changed.

5 Conclusion

We introduced a general-purpose changepoint detection algorithm, CHAMP, that extends Bayesian changepoint detection to settings in which it is difficult or undesirable to integrate out the parameters of candidate models. Instead, our method uses estimates of the maximum likelihood parameters for each segment, removing the need for integration of the model evidence. This approach also allows for the detection of parameter changes within a single model, even when model emissions are independent. We evaluated CHAMP on an artificially generated data set, demonstrating the accuracy and consistency of the algorithm and it's improved performance relative to another state-of-the-art changepoint detection method.

Acknowledgements

Scott Niekum, Sarah Osentoski, and Andrew G. Barto were funded in part by the National Science Foundation under grant IIS-1208497. This material is also based upon work supported in part by the National Science Foundation (IIS-0964581) and the DARPA Robotics Challenge programs.

References

[1] R. P. Adams and D. J. MacKay. Bayesian online changepoint detection. *arXiv preprint arXiv:0710.3742*, 2007.

[2] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 2007.

[3] J. V. Braun, R. Braun, and H.-G. Müller. Multiple changepoint fitting via quasilikelihood, with application to dna sequence segmentation. *Biometrika*, 87(2):301–314, 2000.

[4] N. Chopin. Dynamic detection of change points in long time series. *Annals of the Institute of Statistical Mathematics*, 59(2):349–366, 2007.

[5] P. Fearnhead and Z. Liu. On-line inference for multiple changepoint problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4):589–605, 2007.

[6] E. Fox, E. Sudderth, M. Jordan, and A. Willsky. An HDP-HMM for systems with state persistence. In *Proceedings of the 25th International Conference on Machine Learning*, pages 312–319. ACM, 2008.

[7] R. Killick, P. Fearnhead, and I. Eckley. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598, 2012.

[8] M. Zuliani, C. S. Kenney, and B. Manjunath. The multiRANSAC algorithm and its application to detect planar homographies. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 3, pages III–153. IEEE, 2005.