



Metode Conjugate Gradient dan Broyden-Fletcher-Goldfarb-Shanno untuk Pembelajaran Parameter pada Conditional Random Fields

Intan Nurma Yulita^{1,2}

¹Departemen Ilmu Komputer, Universitas Padjadjaran

²Fakultas Ilmu Komputer, Universitas Indonesia

E-mail: intan.nurma@unpad.ac.id

Abstrak

Conditional Random Fields adalah metode yang banyak digunakan untuk pengenalan pola. Implementasi telah dilakukan di daerah penelitian dengan data secara berurutan misalnya gerak dan suara. Pengenalan dilakukan dengan melatih parameter melalui data pelatihan. Untuk mendapatkan parameter yang optimal, pelatihan dilakukan dengan teknik optimasi. Beberapa teknik optimasi yang sering digunakan untuk *Conditional Random Fields* dengan menggunakan *Conjugate Gradient* dan *Broyden-Fletcher-Goldfarb-Shanno*. Oleh karena itu, makalah ini membandingkan kedua metode optimasi ini untuk pengenalan gerakan. Hasilnya menunjukkan akurasi *Conjugate Gradient* dan *Broyden-Fletcher-Goldfarb-Shanno* adalah 68,28% dan 65,21%, sedangkan waktu eksekusi adalah 325,65 dan 379,05 detik. Dari hasil, dapat dilihat bahwa *Conjugate Gradient* lebih baik dari *Broyden-Fletcher-Goldfarb-Shanno* untuk pengenalan gerakan dalam makalah ini.

Kata Kunci: Conditional Random Fields, Pengenalan Gerakan, Conjugate Gradient, Broyden-Fletcher-Goldfarb-Shanno

1. Pendahuluan

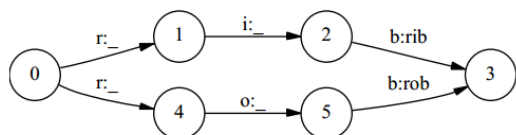
Kebutuhan untuk melakukan segmentasi dan pelabelan dari data berupa *sequence* muncul dalam beberapa bidang. Penggunaan dari model probabilistik seperti Hidden Markov Model (HMM) banyak digunakan untuk menyelesaikan permasalahan ini. Dalam biologi komputasi, HMM digunakan untuk menyelaraskan *biological sequence*, menemukan urutan homolog untuk diketahui kekerabatan evolusi, dan menganalisa struktur sekunder RNA (Elmezain dan AlHamadi, 2008; Spampinato dan Palazzo, 2012; Cooper dan Lipsitch, 1995; Yoon, 2009). Dalam komputasi linguistik dan ilmu komputer, HMM telah diterapkan untuk berbagai masalah dalam pengolahan teks dan suara termasuk segmentasi topic, dan ekstraksi informasi (Mozafari dan Charkari, 2012).

HMM merupakan model generatif yang menerapkan penggunaan *joint probability* untuk observasi berpasangan dan urutan label. Untuk menentukan *joint probability* ini, diperlukan perhitungan semua kemungkinan dari urutan yang diamati (Zhang, 2015). Hal inilah yang menjadi kekurangan dari HMM karena perhitungan ini menjadi tidak praktis untuk merepresentasikan data dengan beberapa fiturnya saling berinteraksi atau yang memiliki ketergantungan yang susah diamati. Untuk mengatasi hal ini, maka model bersyarat

(*conditional model*) menjadi alternatifnya (Lafferty dan McCallum, 2001).

Model bersyarat menspesifikkan kemungkinan dari urutan label yang diberikan. Namun model bersyarat ini memiliki keterbatasan yaitu fitur-fitur yang tidak memiliki ketergantungan (*independent feature*) dalam pembangunan modelnya sehingga probabilitas transisi antara label mungkin tidak hanya tergantung pada pengamatan saat ini, tetapi juga pada pengamatan masa lalu dan masa depan, jika tersedia. Oleh karena itu, model generatif harus membatasi asumsi ketidak-tergantungan pada observasi.

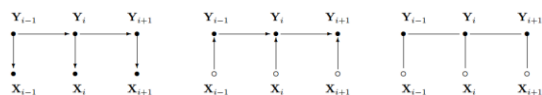
Maximum Entropy Markov Model (MEMM) memberikan batasan-batasan tersebut di dalam model probabilistic yang dibangun. Dalam MEMM, setiap sumber *state* memiliki model eksponensial yang mengambil fitur observasi sebagai masukan, dan output dari distribusi atas kemungkinan *state* berikutnya. Model-model eksponensial dilatih dengan metode skala berulang sesuai dalam kerangka entropi maksimum. Namun seperti model *non-generatif finite-state* lainnya, MEMM memiliki kekurangan dalam permasalahan *label bias* yaitu transisi yang meninggalkan *state* hanya berkompetisi satu sama lainnya bukan melawan semua transisi lainnya dalam model. Sebagai contoh dari masalah *label bias*, gambar dibawah merepresentasikan *finite-state-model* untuk membedakan antara dua kata *rib* dan *rob*.



Gambar 1 Contoh permasalahan label bias (Lafferty dan McCallum, 2001)

Diasumsikan bahwa urutan observasi adalah *rib*. Pada langkah pertama, *r* cocok untuk kedua transisi dari *state* awal, sehingga massa probabilitas akan didistribusikan merata di antara kedua transisi. Untuk langkah selanjutnya, kedua *state* hanya memiliki satu transisi keluar. *State* 1 merupakan pengamatan yang sering berada dalam pelatihan, sedangkan *state* 4 justru sebaliknya. Tetapi baik *state* 1 maupun 4, tidak memiliki pilihan selain untuk meluluskan semua input yang masuk dikarenakan hanya memiliki satu transisi keluar tunggal. Dengan demikian, *state* dengan transit keluar tunggal efektif mengabaikan pengamatan mereka.

Untuk mengatasi masalah ini, diperkenalkan Conditional Random Field (CRF) sebagai alternatifnya, sebuah framework yang melengkapi MEMM dengan mengatasi permasalahan *label bias*. Perbedaan penting antara CRF dan MEMM adalah MEMM menggunakan model- eksponensial setiap *state* untuk probabilitas bersyarat *state* berikutnya jika diberikan kondisi saat ini, sementara CRF memiliki model eksponensial tunggal untuk probabilitas gabungan dari seluruh urutan label jika diberikan urutan observasi. Untuk lebih jelas perbedaan antara HMM, MEMM, dan CRF dapat dilihat pada gambar dibawah ini:



Gambar 2 Struktur Grafik dari HMM (kiri), MEMM (tengah), dan CRF (kanan)

2. Conditional Random Fields

Dalam CRF, diasumsikan, *X* adalah variabel acak terhadap data *sequence* yang diberi label, dan *Y* adalah variabel random atas sesuai urutan label. Semua komponen Y_i bagian dari *Y*. Random variabel *X* dan *Y* secara bersama-sama didistribusikan, dan dibangun model bersyarat $p(Y | X)$ dari observasi dan urutan, dan tidak secara eksplisit memodelkan marginal $p(X)$ (Sutton dan McCallum, 2006).

Sehingga dapat didefinisikan bahwa:

$$X = X_1, X_2, X_3, X_4, \dots, X_n$$

$$Y = y_1, y_2, y_3, y_4, \dots, y_n$$

Kedua vector memiliki panjang yang sama. Peluang input *X* terhadap label *Y* didasarkan perhitungan sebagai berikut (Klinger dan Tomanek, 2007):

$$p(y|x) = (\exp \Delta) / \sum_{ij} (\exp \Delta)$$

Dimana

$$\Delta = \sum_{i=1}^n \sum_{j=1}^m \theta_i \phi_i(x_i, y_i, y_{i-1})$$

Dan $\phi(x, j, y_i, y_{i-1})$ adalah fungsi fitur dari posisi saat ini (*j*) dan θ adalah bobot dari fungsi fitur.

Fungsi fitur bisa tidak hanya berdasarkan dua posisi yang berdekatan tetapi jangkauannya bisa diperpanjang sesuai dengan ukuran jendela yang diatur. Pemodelan dari fungsi fitur ini diperoleh melalui proses pembelajaran dari data yang dimiliki. Untuk mencapai konvergensi, maka dua metode yang umumnya digunakan akan diuji pada penelitian ini yaitu *Conjugate Gradient* dan *Broyden-Fletcher-Goldfarb-Shanno* yang akan dijelaskan pada subbab berikut ini.

2.1 Metode Conjugate Gradient

Conjugate Gradient (CG) merupakan salah satu metode yang banyak digunakan untuk memecahkan permasalahan sistem linier dengan skala besar. Mekanisme dapat dilakukan secara langsung atau perulangan. Mekanisme langsung sering mengalami kendala untuk menanganani data yang besar sehingga banyak dilakukan mekanisme perulangan untuk mengatasinya (Hazewinkel, 2001).

Dimisalkan sistem persamaan linear yang dimiliki adalah sebagai berikut:

$$A x = b$$

A adalah matriks $n \times n$

Dengan menggunakan metode iterative CG, maka dilakukan inialisasi x_0 yang menjadi nilai perkiraan awal untuk mencari solusi sistem persamaan linear *x*. x_0 diinisialisasikan dengan nilai 0 dan kemudian dilakukan pengukuran seberapa dekat terhadap solusi yang didapatkan. Dengan menggunakan fungsi kuadrat Shewchuk yaitu $f(x) = 1/2 (x^T A x - b x)$ maka diperoleh gradient pada titik x_0 adalah d_0 .

Didefinisikan juga s_k sebagai sisa pada langkah ke-*k* yaitu:

$$s_k = b - A x_k$$

Dikarenakan s_k sama dengan gradient negative dari *f* pada x_k maka selanjutnya dapat diketahui bahwa:

$$d_0 = s_0 = b - A x_0$$

$$\alpha_i = (s_i^T s_i) / (d_i^T A d_i)$$

$$x_{i+1} = x_i + \alpha_i d_i$$

$$s_{i+1} = s_i - \alpha_i A d_i$$

$$\beta_{i+1} = (s_{i+1}^T s_{i+1}) / (s_i^T s_i)$$

$$d_{i+1} = s_{i+1} + \beta_{i+1} d_i$$

2.2 Metode Broyden-Fletcher-Goldfarb-Shanno

Metode ini dikembangkan oleh Broyden-Fletcher-Goldfarb-Shanno dan banyak diterapkan untuk optimasi numerik. Mekanismenya merupakan pengembangan dari quasi newton (Bonnans dkk, 2006) (Broyden, 1970). Persamaan umum quasi newton adalah sebagai berikut:

$$B^{k+1} s^k = y^k$$

Dimana
 $s^k = x^{k+1} - x^k$

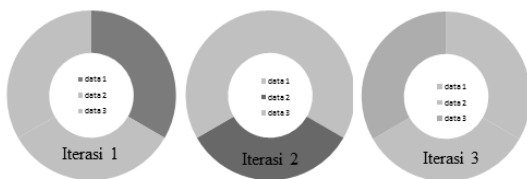
Prosedur BFGS adalah sebagai berikut:

- a. Tentukan nilai x^0 dan B^0 serta $e \in (0,1)$. B^0 diasumsikan I jika nilainya tidak diketahui
- b. Hitung $c^k = \Delta f(x^k)$ dengan $k=0$
- c. Hitung $\|c^k\|$
- d. Jika $\|c^k\| < e$ maka berhenti dan jika tidak maka lanjut ke iterasi selanjutnya
- e. Hitung d^k dari persamaan linear melalui persamaan $B^k d^k = -c^k$
- f. Hitung nilai $\lambda^k = \lambda$ dari fungsi $f(x^k + \lambda d^k) = \min_{\lambda > 0} f(x^k + \lambda d^k)$
- g. Hitung $x^{k+1} = x^k + \lambda d^k$
- h. Hitung $s^k = x^{k+1} - x^k = \lambda d^k$, c^{k+1} , $y^k = c^{k+1} - c^k$
- i. Update BFGS
- j. Kembali ke langkah c dengan $k=k+1$

3. Hasil dan Pembahasan

Analisis dilakukan dengan membandingkan CG dan BFGS untuk proses konvergensi pembentukan fungsi fitur di dalam CRF pada dataset yang berasal dari *School of Art, Sciences and Humanities University of Sao Paulo* dan dapat diperoleh melalui tautan https://archive.ics.uci.edu/ml/machine-learning-databases/00302/gesture_phase_dataset.zip (Wagner dkk, 2014).

Dataset tersebut memuat segmen fase gerak yang terdiri dari enam posisi titik tubuh dan masing-masing posisi tersebut memiliki tiga buah sumbu koordinat. Skema pengujian dilakukan melalui *3-cross validation* yang artinya pengujian dilakukan tiga kali seperti yang dijelaskan oleh Gambar 3.



Gambar 3. 3-cross-validation

Validasi dengan menggunakan *3-cross validation* akan menguji data sebanyak tiga kali ($K=3$). Pada iterasi pertama, data pertama menjadi data uji sedangkan data kedua dan ketiga menjadi data latih. Selanjutnya, pada iterasi kedua, data kedua menjadi data uji sementara yang pertama dan ketiga menjadi data latih. Demikian pula, pada iterasi ketiga, data ketiga adalah data tes, sedangkan sisanya adalah data latihnya. Kinerja masing-masing metode diuji untuk pengenalan gerak pada setiap iterasi. Kinerja ini didasarkan pada akurasi dan waktu eksekusi. Hasil setiap tes ditampilkan dalam sub-bab berikutnya.

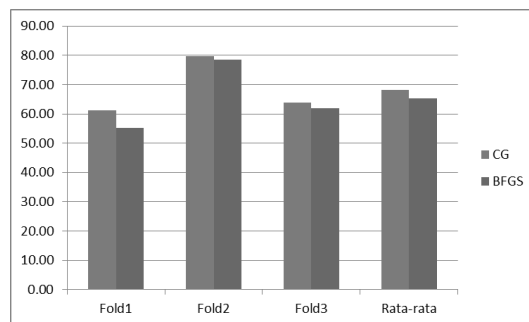
3.1 Akurasi

Performansi kedua metode diuji berdasarkan persentase akurasinya sebagaimana yang ditampilkan di tabel 1.

Tabel 1: Akurasi

Fold	CG	BFGS
1	61.27	55.21
2	79.64	78.43
3	63.95	62.01
Rata-rata	68.28	65.21

Dari tabel dapat diketahui bahwa persentase akurasi CG pada fold 1 lebih tinggi sebesar 6.06 % dibandingkan BFGS. Pada fold 2, akurasi CG lebih unggul dibanding BFGS dengan selisih yang kecil. Hal ini juga terjadi pada fold 3. Secara rata-rata, CG lebih unggul sebesar 3.07 % dibanding BFGS. Ilustrasi selisih akurasi antara CG dan BFGS ditampilkan di gambar 4 yang menunjukkan bahwa CG selalu lebih unggul di dalam penggunaan dataset pada penelitian ini.



Gambar 4. Persentase akurasi

Pada gambar 4, disajikan persentase akurasi yang direpresentasikan oleh sumbu y sedangkan metode disajikan dalam sumbu x. CG ditampilkan oleh batang berwarna abu-abu muda sedangkan BFGS oleh warna abu-abu tua.

3.2 Waktu Eksekusi

Analisis kedua metode juga dihitung berdasarkan waktu eksekusi di dalam menemukan konvergensi untuk memperoleh solusi. Waktu eksekusi yang disajikan di hitung dalam detik pada tabel 2 berikut ini:

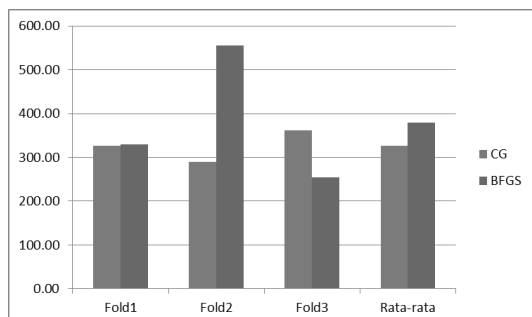
Tabel 2: Waktu eksekusi

Fold	CG	BFGS
1	326.47	329.00
2	289.04	554.59
3	361.43	253.56
Rata-rata	325.65	379.05

Dari tabel 2, dapat diketahui bahwa pada fold 1 yang menghasilkan selisih akurasi yang cukup besar ternyata CG membutuhkan waktu sebesar 326.47 detik sedangkan BFGS sebesar 329.00 detik sehingga meski akurasi CG lebih besar tetapi solusi ditemukan dalam waktu yang lebih sedikit dibandingkan BFGS. Pada fold 2, CG membutuhkan waktu lebih cepat sebesar 265.55 detik



dibanding BFGS. Sedangkan pada fold 3, CG lebih lama sebesar 107.87 detik dibanding BFGS. Namun secara rata-rata, CG membutuhkan waktu lebih cepat. Selisih waktu antara CG dan BFGS terlihat jelas pada gambar 5. Pada gambar tersebut, sumbu x merepresentasikan metode sedangkan sumbu y adalah waktu eksekusi (dalam detik).



Gambar 5. Persentase waktu eksekusi

4. Kesimpulan

Dari kedua hasil yang telah didapat pada penelitian ini maka diketahui bahwa akurasi CG lebih tinggi dibandingkan BFGS namun waktu eksekusi yang dibutuhkan lebih sedikit. Sehingga dapat disimpulkan bahwa CG lebih unggul dibanding BFGS.

Ucapan Terima Kasih

Penulis mengucapkan terima kasih kepada LPDP dan juga *Machine Learning and Computer Vision Laboratory*, Universitas Indonesia yang memberi dukungan terkait riset yang dipaparkan

Daftar Pustaka

- M. Elmezain, A. Al-hamadi. 2008 A hidden markov model-based isolated and meaningful hand gesture recognition, In Proceeding of World Academy of Science, Technology, Engineering and Technology 394–401.
- C. Spampinato, S. Palazzo. 2012. Hidden markov models for detecting anomalous fish trajectories in underwater footage, IEEE International

Workshop on Machine Learning for Signal Processing.

- B. Cooper, M. Lipsitch. 1995. The analysis of hospital infection data using hidden markov models, *Biostatistics* 5.2 223–237.
- B.-J. Yoon. 2009. Hidden markov models and their applications in biological sequence analysis, *Current Genomics* 10 402–415.
- Mozafari, K. & Charkari, N.M., 2012. A Novel Fuzzy HMM Approach for Human Action. *Knowledge Technology, Springer*, pp.184–193.
- Zhang, X., 2005. Human Motion Recognition through Fuzzy Hidden Markov Model. In *Proc. of the International Conf. on Computational Intelligence for Modelling, Control, and Automation*. pp. 450–456.
- Lafferty, J. & McCallum, A., 2001. Conditional Random Fields?: Probabilistic Models for Segmenting and Labeling Sequence Data, 2001(Icml), pp.282–289.
- Sutton, C., McCallum, A. 2006. An Introduction to Conditional Random Fields for Relational Learning. In "Introduction to Statistical Relational Learning". Edited by Lise Getoor and Ben Taskar. MIT Press. (2006) Online PDF
- Klinger, R., Tomanek, K. 2007. Classical Probabilistic Models and Conditional Random Fields. *Algorithm Engineering Report TR07-2-013*, Department of Computer Science, Dortmund University of Technology ISSN 1864-4503. Online PDF
- Hazewinkel, Michiel. 2001. Conjugate gradients, method, *Encyclopedia of Mathematics*, Springer, ISBN 978-1-55608-010-4
- Bonnans, J. Frédéric; dkk. 2006, Numerical optimization: Theoretical and practical aspects, Universitext (Second revised ed. of translation of 1997 French ed.), Berlin: Springer-Verlag, pp. xiv+490, doi:10.1007/978-3-540-35447-5, ISBN 3-540-35445-X, MR 2265882 French Original: Optimisation numérique: Aspects théoriques et pratiques ISBN 3-540-63183-6
- Broyden, C. G. 1970, The convergence of a class of double-rank minimization algorithms, *Journal of the Institute of Mathematics and Its Applications*, 6: 76–90, doi:10.1093/imamat/6.1.76
- Wagner, dkk. 2014. . Gesture Unit Segmentation Using Spatial-Temporal Information and Machine Learning. In: 27th Florida Artificial Intelligence Research Society Conference (FLAIRS), Pensacola Beach.