

## Bab III

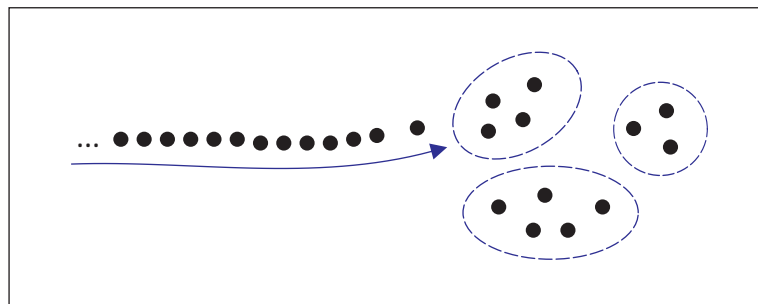
### *Clustering* pada Aliran Informasi

Pada bab ini, pembahasan ditekankan kepada algoritma *clustering* yang terkait dengan Tugas Akhir ini. Selain itu, pada bab ini dipaparkan pula kajian pengembangan algoritma *clustering* dengan penggunaan batasan (*constrained clustering*).

#### III.1 *Incremental Clustering*

Selama ini, *dataset* yang menjadi masukan pada analisis *cluster* biasanya diasumsikan bersifat statik. Maksudnya, *dataset* dianggap tidak berubah selama proses analisis dilakukan. Oleh karena itu, proses analisis dapat diterapkan secara langsung terhadap keseluruhan *dataset*. Namun, jika terjadi modifikasi terhadap *dataset* (penambahan, pengubahan, atau penghapusan), maka harus dilakukan analisis ulang (*reclustering*) terhadap keseluruhan data tersebut.

Pada kasus ukuran dataset sangat besar dan laju pertambahan ukurannya sangat cepat, proses *reclustering* tidaklah efisien. Diharapkan agar analisis *cluster* dapat dilakukan secara *incremental* atau bertahap. Sehingga proses *clustering* dapat dilakukan secara terus menerus —biasanya secara periodik— tanpa perlu memproses ulang seluruh *dataset*.



Gambar III-1: Ilustrasi *incremental clustering*

Berita merupakan aliran informasi (*information stream*) yang senantiasa mengalir. Saat ini, terdapat banyak sumber berita yang masing-masing mempublikasikan ratusan berita setiap harinya. Tentunya akan sangat tidak efisien jika proses *clustering* perlu dilakukan kembali terhadap seluruh berita yang sudah pernah diproses maupun berita yang baru datang. Oleh karena itu, dikembangkan metode *clustering* yang bersifat *incremental*.

## III.2 Ragam Algoritma *Clustering*

Pada bagian ini, akan dipaparkan beberapa algoritma *clustering*, terutama yang berkaitan erat dengan bahasan pokok Tugas Akhir ini. Pembahasan ditekankan pada algoritma *clustering* yang bersifat *incremental* dengan sebelumnya didahului dengan pemaparan singkat mengenai algoritma K-Means sebagai dasar algoritma lanjut.

**K-Means** K-Means [Mac67] adalah salah satu algoritma *clustering* yang paling sederhana. Algoritma ini membagi objek menjadi sejumlah  $k$  kelompok—nilai  $k$  sudah ditentukan sebelumnya. Sebuah *cluster* direpresentasikan dengan *centroid*, yaitu rata-rata dari semua objek yang terdapat pada *cluster* tersebut.

K-Means merupakan algoritma *clustering* yang menghasilkan solusi yang eksklusif dan non-hierarkis. Prosesnya dijalankan secara iteratif sampai susunan *cluster* dianggap sudah stabil (berdasarkan batasan tertentu). Algoritma K-Means secara runut dapat dilihat pada Algoritma III-1.

---

### Algoritma III-1 Algoritma K-Means

---

1. Tentukan titik-titik sebanyak  $k$  sebagai *centroid* awal
  2. Hubungkan setiap objek dengan *centroid* yang terdekat
  3. Hitung ulang posisi semua *centroid*
  4. Ulangi langkah 2 dan 3 sampai posisi *centroid* tidak berubah
- 

Untuk menentukan posisi *centroid* pada saat inisialisasi, terdapat beberapa alternatif cara. Salah satunya adalah dengan menempatkan titik-titik *centroid* secara acak. Kompleksitas waktu algoritma K-Means adalah  $O(I * K * n)$ , dengan  $I$  adalah banyaknya iterasi yang dilakukan dan  $n$  adalah banyaknya objek. Nilai  $I$  biasanya tidak terlalu besar dan dapat dibatasi, sehingga algoritma K-Means dapat dikatakan sangat efisien (hampir linear).

Walaupun sederhana, kualitas solusi *clustering* yang dihasilkan oleh K-Means relatif baik. Saat ini telah dikembangkan beberapa varian dan penyempurnaan terhadap K-Means, seperti Bisecting K-Means. Algoritma ini diklaim lebih baik dari algoritma K-Means asli berdasarkan pengujian konsistensi kualitas [SKK00].

***Single-Pass Incremental Clustering (INCR)*** Algoritma INCR pertama kali dipublikasikan pada [YCB<sup>+</sup>99] sebagai hasil riset pada bidang yang relatif baru, yaitu Topic Detection and Tracking (TDT)<sup>1</sup>. Fokus utama bidang ini adalah mendeteksi terjadinya suatu peristiwa baru pada aliran berita dan untuk melakukan pelacakan atas peristiwa-peristiwa yang sudah terjadi.

Secara umum, ada lima bidang penelitian utama dalam TDT, yaitu:

1. *Story segmentation*

Membagi sebuah aliran berita menjadi potongan-potongan berita yang *cohesive*.

Lazimnya metode ini diterapkan pada berita yang ada di TV atau radio, karena

---

<sup>1</sup><http://www.nist.gov/speech/tests/tdt/>

beritanya terus menerus dan tidak ada batas yang jelas antara satu berita dengan berita yang lain.

2. *Topic tracking*

Melacak berita-berita berdasarkan sampel berita yang sudah ada. Metode ini mengasosiasikan berita yang baru datang dengan berita terkait sebelumnya.

3. *Topic detection*

Membangun *cluster* berita-berita yang membahas topik yang sama.

4. *First story detection*

Mendeteksi apakah sebuah berita merupakan berita yang membahas suatu peristiwa yang baru terjadi.

5. *Link detection*

Mendeteksi apakah dua buah berita membahas topik yang terkait.

Algoritma INCR menggunakan konsep *centroid* seperti pada K-Means. Dokumen diproses secara berurutan satu demi satu dan *cluster* dibangun secara *incremental*. Sebuah dokumen baru dimasukkan ke dalam *cluster* yang paling dekat apabila *similarity* antara dokumen dan *cluster* tersebut melebihi suatu batasan tertentu (*clustering threshold*). Jika tidak, dokumen dianggap sebagai sebuah *cluster* baru. Algoritma INCR secara runut dapat dilihat pada Algoritma III-2.

---

**Algoritma III-2** Algoritma INCR

---

1. Pada saat awal, hanya terdapat sebuah berita yang membentuk *cluster* tunggal
  2. Untuk setiap berita baru  $d_*$ , hitung *similarity* antara  $d_*$  dengan semua *cluster* yang ada dalam *time window*  $T$
  3. Pilih *cluster*  $C_i$  yang paling dekat dengan  $d_*$
  4. Jika  $sim(d_*, C_i)$  melebihi nilai batas  $\epsilon$ , maka  $d_*$  dimasukkan ke dalam  $C_i$
  5. Jika tidak, maka buat *cluster* baru  $C_*$  untuk  $d_*$
  6. Ulangi langkah 2 sampai 4 untuk semua berita baru
- 

Yang [YCB<sup>+</sup>99] memperkenalkan karakteristik penting pada berita di mana berita-berita yang membahas suatu peristiwa cenderung datang dalam waktu yang berdekatan (*temporal locality*). Selain itu, rentang suatu peristiwa biasanya tidak lama (antara satu minggu sampai satu bulan). Oleh karena itu, daripada melakukan pengujian *similarity* dokumen baru terhadap seluruh *cluster*, pengujian cukup dilakukan terhadap *cluster* yang berada pada rentang waktu tertentu (*time window*).

**Chung-McLeod** Pada algoritma INCR (bagian III.2), urutan kedatangan dokumen sangat memengaruhi susunan *cluster* yang dihasilkan. Walaupun sebenarnya untuk dokumen yang berupa berita, urutan kedatangan dokumen sudah pasti (sesuai waktu publikasi berita), namun karakteristik tersebut sebaiknya dihindari. Oleh karena itu, Chung dan McLeod mengusulkan algoritma untuk menghindari hal tersebut [CM05].

Algoritma Chung-McLeod menggunakan konsep tetangga terdekat yang sama (*shared nearest neighbor*) seperti yang digunakan pada algoritma SNN [ESK03]. Algoritma ini dikembangkan berdasarkan pengamatan bahwa sifat sebuah objek dipengaruhi oleh sifat tetangga-tetangganya. Jika dilihat dari sudut pandang *clustering*, sebuah objek akan masuk dalam *cluster* yang tergantung *cluster* tetangganya.

Pada algoritma Chung-McLeod, diperkenalkan konsep  $\epsilon$ -*neighborhood*, yaitu himpunan dokumen terdekat yang memiliki *similarity* lebih dari nilai batasan  $\epsilon$ . Dokumen tetangga yang berada dalam  $\epsilon$ -*neighborhood*-lah yang menentukan keanggotaan *cluster* suatu dokumen.

Pada saat inisialisasi, hanya terdapat sebuah berita yang membentuk sebuah *cluster* tunggal. Kemudian, dokumen yang datang akan dikelompokkan berdasarkan *cluster* tetangganya. Algoritma Chung-McLeod dapat dilihat pada Algoritma III-3.

---

#### Algoritma III-3 Algoritma Chung-McLeod

---

**Langkah 1:** Inisialisasi

Dokumen  $d_0$  membentuk *cluster* tunggal  $C_0$

**Langkah 2:** Pencarian ketetanggaan ( $\epsilon$ -*neighborhood*)

Ketika ada dokumen baru  $d_*$ , cari seluruh dokumen  $d_i$  yang memiliki *similarity* lebih besar dari nilai batas ( $\text{sim}(d_*, C_i) \geq \epsilon$ ), dengan syarat  $d_i$  masih berada dalam *time window*

Tentukan  $C_{d_*}$ , yaitu himpunan semua *cluster* di mana tetangga  $d_*$  tersebut tergabung

**Langkah 3:** Identifikasi *cluster*

Jika  $C_{d_*}$  kosong, maka buat *cluster* baru  $C_*$  untuk  $d_*$

Jika tidak, hitung *similarity* antara ketetanggaan  $d_*$  dengan  $C_i \in C_{d_*}$

Pilih  $C_i \in C_{d_*}$  dengan *similarity* yang terbesar.

Masukkan  $d_*$  ke dalam  $C_i$ .

---

Algoritma Chung-McLeod tidak secara eksplisit membutuhkan parameter *time window*. Namun, parameter ini dapat bermanfaat untuk membatasi jumlah himpunan tetangga dokumen. Selain itu, penggunaan parameter *time window* juga dapat mencegah dokumen dimasukkan ke dalam sebuah *cluster* yang sudah sangat lama—mungkin sebuah *cluster* yang membahas peristiwa yang mirip dengan peristiwa kali ini.

### III.3 *Clustering* dengan Pembatasan (*Constrained Clustering*)

Analisis *cluster* lazimnya dilakukan dalam keadaan tak terbimbing (*unsupervised*). Algoritma menerima himpunan data yang akan dianalisis dan tidak diberikan informasi lain (misalnya label *cluster*) untuk “membimbing” algoritma dalam menganalisis data. Namun, dalam aplikasi dalam dunia nyata, seringkali terdapat informasi tambahan atau informasi latar belakang yang dapat digunakan untuk membantu proses analisis *cluster*. Algoritma *clustering* biasa tidak dapat menangani informasi tambahan seperti demikian.

Salah satu informasi tambahan yang dapat digunakan untuk membantu proses analisis *cluster* adalah masukan atau umpan balik dari pengguna (*user feedback*). Biasanya masukan seperti ini digunakan pada aplikasi *clustering* yang sifatnya interaktif. Pengguna dapat menginformasikan kepada sistem bahwa sistem membuat kesalahan. Informasi ini dapat digunakan dalam proses *clustering* iterasi berikutnya untuk memperbaiki kualitas *clustering*.

Informasi tambahan seperti demikian dapat dianggap sebagai batasan (*constraint*) untuk membatasi proses *clustering* sehingga hasilnya lebih baik dibandingkan tanpa adanya batasan. Pendekatan seperti ini dapat disebut sebagai *semi-supervised clustering*.

### III.3.1 Representasi Batasan (*Constraint*)

Salah satu cara yang dapat digunakan untuk merepresentasikan batasan secara generik tanpa terkait dengan bidang persoalan adalah dengan menggunakan batasan antar objek (*pairwise constraint*) [WC00, WCRS01]. Terdapat dua jenis batasan yang terdefinisi, yaitu:

1. Batasan *must-link*, digunakan untuk menyatakan bahwa dua buah objek  $d_i$  dan  $d_j$  harus dimasukkan ke dalam *cluster* yang sama
2. Batasan *cannot-link*, digunakan untuk menyatakan bahwa dua buah objek  $d_i$  dan  $d_j$  tidak boleh dimasukkan ke dalam *cluster* yang sama

Batasan-batasan tersebut bersifat *hard constraint*, artinya solusi *clustering* harus memenuhi semua batasan yang ada dan tidak diperkenankan sama sekali untuk melanggar batasan tersebut. Berbeda dengan batasan yang bersifat *soft constraint* yang hanya menyatakan preferensi pada solusi *clustering* dan sifatnya lebih fleksibel karena dapat dilanggar oleh algoritma. Setiap batasan dapat diatur “kekuatan”-nya sehingga batasan paling lemahlah yang kemungkinannya paling besar untuk dilanggar.

### III.3.2 Akomodasi *Constraint* pada Algoritma

Pembahasan pada bagian sebelumnya mengenai penggunaan batasan pada *clustering* tidak terbatas hanya pada algoritma *clustering* tertentu saja. Karena sifatnya yang generik, algoritma *clustering* harus dimodifikasi agar dapat mengakomodasi penggunaan *constraint*.

Penggunaan batasan ini difokuskan terhadap algoritma *clustering* yang bersifat *partitioning*. Pada algoritma yang berjenis demikian, umumnya terdapat tahapan yang disebut tahap pemasukan objek ke dalam *cluster* (*cluster assignment step*) [Wag02]. Pada tahap ini, objek dimasukkan ke dalam *cluster* terbaik yang sesuai. Oleh karena itu, pada tahap inilah modifikasi algoritma dapat dilakukan untuk dapat mengakomodasi *constraint*. Penjelasan modifikasi algoritma yang dilakukan pada Tugas Akhir ini dapat dilihat pada bagian IV.6.