Forum Statistika dan Komputasi, April 2010 p : 1-7 Vol 15 No.1
ISSN : 0853-8115

METODE POHON GABUNGAN: SOLUSI PILIHAN UNTUK MENGATASI KELEMAHAN POHON REGRESI DAN KLASIFIKASI TUNGGAL

(*Ensemble Tree : An Alternative toward Simple Classification and Regression Tree*)

Bagus Sartono[1)], Utami Dyah Syafitri[2)]
Departemen Statistika IPB
E-mail : [1)] bagusco4@yahoo.com, [2)] utamids@ipb.ac.id

Abstract

Classification and regression tree has been a widely used tool in various applied fields due to its capability to give excellent predictive analysis. Later on, ensemble tree appeared to enhance simple tree analysis and deals with some of the weakness found in simple techniques. The ensemble tree basically combines predictions values of many simple trees into a single prediction value. This paper is intended as an introductory article to give a brief overview of the available ensemble tree methods which might be found in detail in a variety of reading materials.

Keywords: Ensemble tree, regression tree, classification tree

PENDAHULUAN

Penggunaan metode pohon regresi (*regression tree*) dan pohon klasifikasi (*classification tree*) saat ini dapat dengan mudah ditemui dalam berbagai penelitian. Kemampuan metode ini dalam memberikan dugaan dengan tingkat kesalahan yang kecil dan kemudahan memberikan interpretasi terhadap hasil analisis menjadikan pendorong populernya teknik ini. Kemajuan perangkat lunak yang ada juga menjadi salah satu pemicu tingginya penggunaan metode ini terutama di kalangan praktisi yang tidak memiliki landasan pengetahuan statistika yang mumpuni.

Metode pohon ini juga merupakan salah satu metode yang umum digunakan oleh para pengembang dan praktisi *data-mining*. Berdasarkan data yang ada, para analis melakukan pemodelan sehingga memperoleh pohon tunggal yang dianggap terbaik. Selanjutnya dari pohon tunggal tersebut dapat dengan mudah mengubahnya menjadi *if-then rule* untuk diaplikasikan ke data lain dalam proses menghasilkan prediksi dengan cepat.

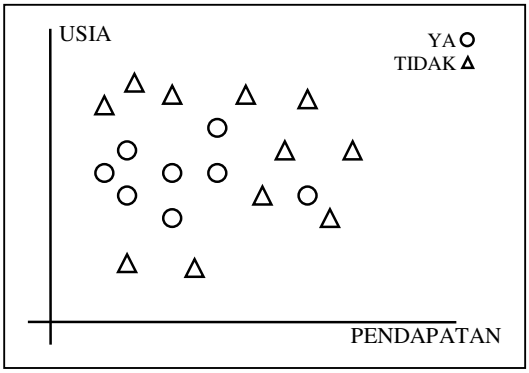
Selanjutnya metode pohon ini mengalami perkembangan yang cukup pesat. Salah satunya adalah pembuatan pohon gabungan (*ensemble-tree*) yang memanfaatkan beberapa pohon, bukan hanya satu pohon, untuk melakukan dugaan. Dengan kata lain dugaan dari suatu data tertentu, merupakan penggabungan dari dugaan-dugaan yang dihasilkan oleh beberapa pohon. Pada proses analisis, teknik ini menghasilkan beberapa pohon yang selanjutnya digunakan secara simultan untuk melakukan pendugaan. Proses lebih lanjut dari tehnik ini akan diuraikan secara lebih ringkas dan mudah dipahami bagi pembaca.

POHON REGRESI DAN KLASIFIKASI

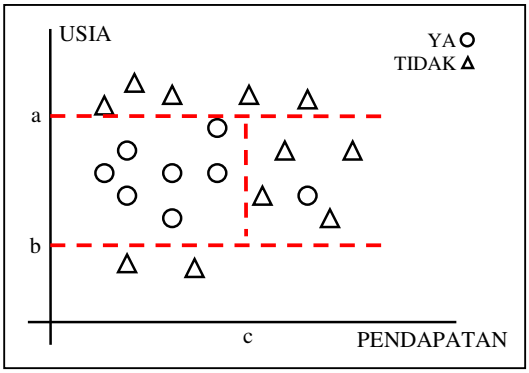
Kelinearan hubungan antara peubah prediktor dengan peubah respon sering kali menjadi kendala penggunaan metode regresi klasik. Hal yang relatif sama juga terjadi pada metode klasifikasi yang menggunakan model tertentu seperti regresi logistik dan diskriminan linear maupun kuadratik. Penambahan unsur tak linear dan upaya transformasi kadang-kadang ditempuh, namun tidak sepenuhnya berhasil. Kalaupun dapat dilakukan, sering menimbulkan kesulitan memberikan interpretasi.

Perhatikan ilustrasi rekaan berikut ini. Andaikan dua peubah numerik USIA dan PENDAPATAN digunakan untuk menduga apakah seseorang berminat atau tidak melakukan pembelian suatu produk. Dalam hal ini, peubah minat membeli menjadi peubah respon dengan dua kelas: YA dan TIDAK. Plot antara nilai-nilai peubah usia dan pendapatan pada dua kelas peubah respon tersebut disajikan pada Gambar 1.

Metode-metode yang berbasis pada model linear akan kesulitan memperoleh model pengklasifikasian yang mampu melakukan klasifikasi dengan kesalahan yang sangat rendah. Dengan menerapkan pohon klasifikasi, kita dapat mengharapkan memperoleh pemisahan (*splitting*) seperti yang ditunjukkan pada Gambar 2.



Gambar 1. Ilustrasi kasus klasifikasi kemungkinan pembelian berdasarkan peubah USIA dan PENDAPATAN



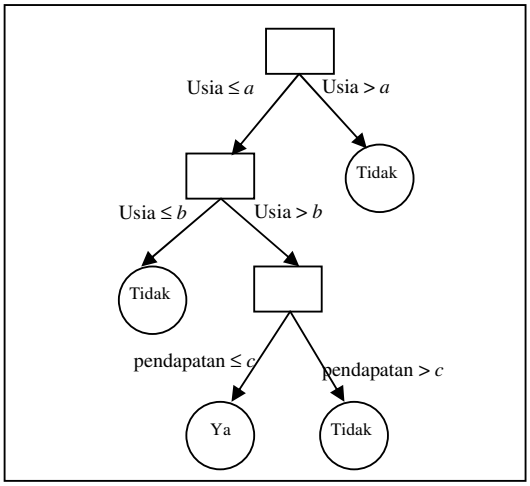
Gambar 2. Ilustrasi pemisahan tingkat kemungkinan pembelian berdasarkan peubah USIA dan PENDAPATAN

Proses pemisahan yang ditunjukkan pada Gambar 2, selanjutnya dapat diubah tampilannya dalam bentuk pohon seperti pada Gambar 3. Tampilan pohon pada Gambar 3 memberikan interpretasi yang lebih mudah bagi para pengguna analisis ini. Dengan mudah juga, pohon tersebut digunakan untuk menyusun sebuah gugus kaidah jika-maka (*if-then rule*) sebagai berikut:

1. Jika USIA > a maka orang tersebut TIDAK berminat membeli
2. Jika USIA ≤ a dan USIA ≤ b (bisa disingkat jika USIA ≤ b) maka orang tersebut TIDAK berminat membeli
3. Jika b < USIA ≤ a dan PENDAPATAN ≤ c maka orang tersebut tergolong kelas YA yang berarti berminat membeli
4. Jika b < USIA ≤ a dan PENDAPATAN > c maka orang tersebut TIDAK berminat membeli

Algoritma penyusunan pohon klasifikasi dan pohon regresi telah banyak diusulkan oleh banyak penulis. Beberapa yang banyak digunakan antara lain adalah ID3 (Quinlan 1986) yang selanjutnya dikembangkan menjadi algoritma C4.5 dan C5, CHAID, CART dan QUEST (Loh dan Shih, 1997). Andaikan S adalah gugus data yang digunakan untuk membangun pohon, pada prinsipnya algoritma-algoritma tersebut berjalan sebagai berikut:

1. identifikasi variabel penjelas dan nilainya (atau levelnya kalau itu adalah variabel kategorik) yang dapat digunakan sebagai pemisah data S menjadi dua atau lebih sub-set data.
2. lakukan iterasi terhadap proses nomor 1 terhadap subset-subset yang ada sampai ditemukan salah satu dari dua hal berikut:
   a. semua subset sudah homogen nilainya
   b. tidak ada lagi variabel penjelas yang bisa digunakan
   c. jumlah amatan di dalam subset sudah terlalu sedikit untuk menghasilkan pemisahan yang memuaskan
3. lakukan pemangkasan (*pruning*) jika pohon yang dihasilkan dinilai terlalu besar.



Gambar 3. Ilustrasi visualisasi pohon untuk kasus pada Gambar 2

Proses identifikasi peubah penjelas dan nilai yang menjadi batas pemisah dapat dilakukan dengan berbagai cara dan berbagai kriteria. Namun tujuan dari pemisahan ini pada berbagai metode adalah sama yaitu mendapatkan subset-subset yang memiliki nilai peubah respon yang lebih homogen dibandingkan sebelum dilakukan pemisahan. Ukuran kehomogenan untuk peubah respon numerik, pada kasus pohon regresi, antara lain adalah jumlah kuadrat. Sedangkan pada kasus pohon klasifikasi, yang umum digunakan adalah *entropy*, yang selanjutnya digunakan untuk menghitung *information gain* hasil pemisahan. Penjelasan tentang entropy dan kriteria lain dapat ditemukan di Bramer (2007).

Sebuah peubah dan nilai batas akan dipilih sebagai pemisah jika mampu memberikan subset-

subset dengan tingkat homogenitas yang paling besar. Umumnya tidak ada pengujian, dalam mengembangkan metode CHAID dan QUEST digunakan pengujian statistik formal yang sesuai, misalnya *Chi-Square test* dan ANOVA (lihat algoritma pemilihan peubah di Loh dan Shih (1997)).

Pemisahan suatu subset menjadi subset-subset yang lebih kecil dapat dilakukan dengan menghasilkan dua atau lebih bagian. Pemisahan yang menghasilkan hanya dua subset di setiap tahap sering disebut sebagai *binary-splitting*. Penentuan nilai peubah yang menjadi batas pemisah dikerjakan dengan banyak cara. Algoritma yang menerapkan *greedy search* pada peubah penjelas yang numerik melakukan dengan memeriksa setiap nilai tengah dua buah nilai berbeda untuk menjadi kandidat pemisah. Teknik ini tentu akan memakan banyak waktu. Cara lain yang bisa ditempuh adalah dengan menggunakan nilai persentil yang diawali dengan memeriksa nilai persentil 50 (median), kemudian persentil 25 (kuartil pertama), persentil 12.5%, dan seterusnya. Sedangkan untuk peubah kategorik, penentuan nilai pemisahnya jauh lebih sederhana.

Salah satu penghenti dari proses iterasi pemilihan peubah pemisah pada analisis pohon adalah kecilnya jumlah amatan dalam suatu subset. Beberapa software secara *default* menggunakan nilai antara 5 hingga 10 amatan. Dengan amatan sebanyak itu, sudah dianggap tidak cukup memadai untuk mendapatkan peubah pemisah dengan tingkat keandalan yang tinggi.

Tentu saja jika setiap subset sudah memiliki nilai yang homogen, tidak perlu lagi dilakukan proses pemisahan karena itu berarti telah diperoleh pemisahan sempurna. Namun demikian, meskipun kondisi ini menunjukkan bahwa pohon yang dihasilkan memiliki tingkat akurasi 100% dalam melakukan prediksi, tingkatan itu hanya berlaku pada gugus data yang digunakan saja (*training set*) dan umumnya memiliki hasil prediksi yang kurang memuaskan untuk gugus data lain meskipun dari populasi yang sama. Penggunaan gugus data lain untuk validasi sangat disarankan untuk memberikan penilaian terhadap kinerja pohon.

Turunnya kualitas prediksi suatu pohon yang memiliki kinerja yang prima pada gugus data *training* antara lain disebabkan karena pohon yang dihasilkan tumbuh terlalu besar karena banyaknya pemisahan-pemisahan yang mungkin tidak perlu seperti yang dijelaskan Quinlan (1993) dalam Salzberg (1994). Proses pemangkasan (*pruning*) terhadap pohon yang dihasilkan seringkali diperlukan untuk mendapatkan pohon yang memiliki kemampuan prediksi memuaskan pada berbagai gugus data dari populasi yang sama.

Dengan berkembangnya teknologi komputasi, proses mendapatkan pohon regresi maupun klasifikasi dapat dengan mudah dilakukan. Berbagai software analitik dan datamining menyediakan fasilitas atau prosedur tertentu dengan berbagai pilihan untuk mengubah parameter pohon.

Meskipun teknik analisis ini telah banyak memberikan hasil memuaskan di berbagai bidang terapan, Berk (2008) mencatat bahwa salah satu kelemahan pohon klasifikasi dan regresi adalah sifatnya yang tidak stabil. Dari populasi yang sama, jika diambil contoh yang berbeda maka sangat mungkin diperoleh pohon dengan bentuk yang berbeda. Ketidakstabilan ini oleh Berk (2008) dinyatakan kemungkinan disebabkan oleh: ukuran contoh yang kecil, terminal nodes yang heterogen, dan tingginya korelasi antar peubah penjelas.

METODE-METODE ENSEMBLE

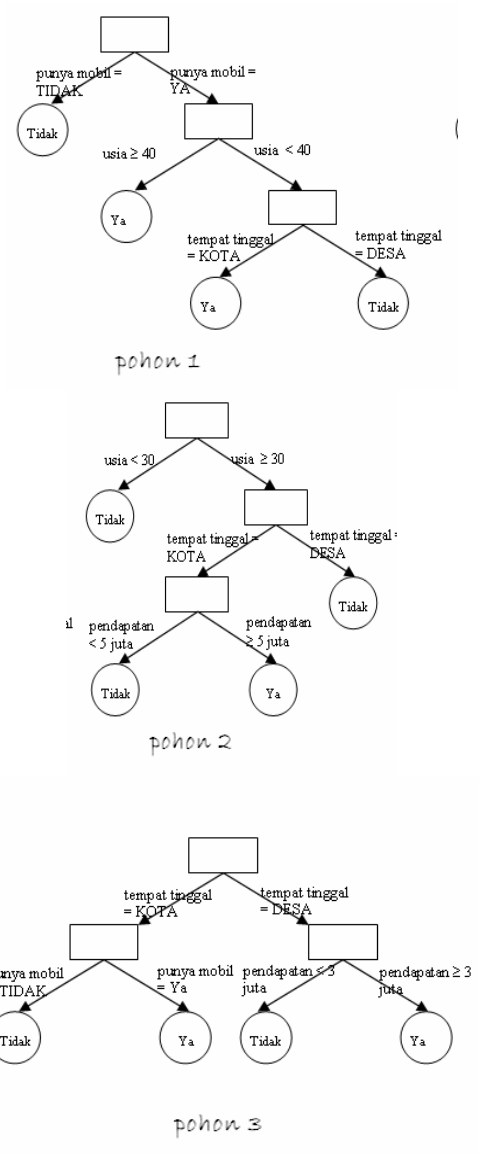
Bagging

Sebelum mendiskusikan lebih jauh tentang konsep dan penerapan *bagging*, ilustrasi berikut ini diharapkan dapat menjadi pengantar yang memadai. Dari sebuah gugus data yang dimiliki kita bisa melakukan pengambilan sebagian pengamatan saja secara acak (*resampling*) yang selanjutnya digunakan untuk menyusun pohon. Pengulangan proses ini akan menghasilkan sebagian data yang berbeda dan diikuti dengan pohon yang berbeda.

Hanya bertujuan sebagai ilustrasi, andaikan kita memperoleh tiga buah pohon berbeda dari tiga kali pengambilan sebagian data seperti yang digambarkan pada Gambar 4. Terdapat empat buah peubah penjelas yang terlibat yaitu usia, tempat tinggal, pendapatan, dan kepemilikan mobil, serta sebuah peubah respon biner (Ya dan Tidak) yang menggambarkan keinginan untuk membeli suatu produk.

Selanjutnya andaikan ada pengamatan baru tertentu yang memiliki karakteristik nilai usia = 28, tempat tinggal = KOTA, pendapatan = 6 juta, dan punya mobil = TIDAK. Maka, berdasarkan pohon yang pertama dan ketiga individu baru tersebut akan diduga memiliki nilai "Ya" untuk peubah keinginan membeli. Sedangkan jika menggunakan pohon kedua, dugaannya adalah "Tidak". Sekarang kita peroleh informasi bahwa dari tiga pohon yang kita miliki, dua pohon menduga "Ya" dan sebuah pohon menduga "Tidak". Prinsip penentuan berdasar suara terbanyak (*majority vote*) selanjutnya digunakan untuk menentukan dugaan akhir. Pada kasus ini dugaan "Ya" yang diambil karena mendapatkan suara lebih banyak. Kegiatan mengambil suara terbanyak merupakan salah satu proses menggabungkan (*ensemble*) nilai dugaan dari beberapa pohon menjadi satu dugaan akhir. Breiman (1996) menyebut proses ini sebagai

proses *aggregating* dalam analisis *bagging*. Nama *bagging* merupakan singkatan dari *bootstrap aggregrating*.



Gambar 4. Ilustrasi penerapan penggabungan dugaan pohon

Berdasarkan namanya, maka dapat diperkirakan ada dua tahapan utama dalam analisis ini, yaitu bootstrap yang tidak lain adalah pengambilan contoh dari data contoh yang dimiliki (*resampling*) dan aggregating yaitu menggabungkan banyak nilai dugaan menjadi satu nilai dugaan. Dengan demikian proses pembuatan dugaan secara bagging menggunakan pohon adalah sebagai berikut:

1. a. (tahapan *bootstrap*) tarik contoh acak dengan permulihan berukuran n dari gugus data training
   b. susun pohon terbaik berdasarkan data tersebut
   c. ulangi langkah a-b sebanyak *k* kali sehingga diperoleh *k* buah pohon acak
2. lakukan pendugaan gabungan berdasarkan *k* buah pohon tersebut (misal menggunakan *majority vote* untuk kasus klasifikasi, atau rata-rata untuk kasus regresi)

Rokach (2008) menyebutkan bahwa selain menggunakan konsep pengambilan suara terbanyak (majority vote) untuk menggabung dugaan dari banyak pohon, bisa juga digunakan penjumlahan dugaan peluang masing-masing kelas. Perlu diingat bahwa dugaan yang dihasilkan oleh suatu pohon dapat berupa nilai peluang. Pada ilustrasi di atas dapat dinyatakan berapa peluang terjadinya "Ya" dan berapa untuk "Tidak". Dari setiap pohon dapat dihitung peluang masing-masing kategori respon, dan selanjutnya dijumlahkan. Kategori yang memiliki jumlah peluang paling besar adalah kategori yang menjadi dugaan final. Rokach (2008) juga menyatakan bahwa penjumlahan dapat dilakukan dengan memberikan bobot tertentu pada setiap nilai peluang dari setiap pohon, misalkan menggunakan nilai entropi.

Penggunaan bagging ini sangat membantu terutama mengatasi sifat ketidakstabilan pohon klasifikasi dan regresi tunggal seperti yang telah disinggung sebelumnya. Hastie et al. (2008) menyatakan bahwa proses bagging dapat mengurangi galat baku dugaan yang dihasilkan oleh pohon tunggal. Hal ini dapat jelas terlihat karena dengan melakukan rata-rata misalnya maka ragam dugaan akan mengecil sedangkan tingkat bias dugaan tidak terpengaruh. Selain itu Breiman (1996) mencatat bahwa pada banyak gugus data yang dia coba, bagging mampu mengurangi tingkat kesalahan klasifikasi pada kasus klasifikasi.

Hal ini tentu tidak berlaku secara keseluruhan. Berk (2008) mencatat beberapa kasus yang mungkin menyebabkan dugaan bagging memiliki ragam dugaan yang lebih besar atau juga bias yang lebih besar pula. Ini terjadi antara lain pada kasus dengan kategori peubah respon yang sangat tidak seimbang, dan juga pada kondisi peubah penjelas yang distribusinya memiliki tingkat kemenjuluran yang tinggi.

Mengenai berapa banyak pengulangan bootstrap yang diperlukan, studi Breiman (1996) menunjukkan bahwa menggunakan 50 kali untuk kasus klasifikasi dan 25 kali untuk kasus regresi dapat memberikan hasil yang memuaskan. Buja dan Stuetzle (2006) dalam Berk (2008) mengungkapkan hasil studinya bahwa untuk memperoleh hasil bagging yang baik, tidak harus melalui pengambilan contoh dengan pemulihan

(*sampling with replacement*) secara berulang, namun dapat saja dilakukan tanpa pemulihan (*sampling without replacement*) jika ukuran contohnya cukup besar.

Random Forest (RF)

Metode RF mulai banyak diperbincangkan sejak tulisan Breiman (2001) muncul pada jurnal Machine Learning. Secara jeli, Breiman (2001) berupaya untuk memperbaiki proses pendugaan yang dilakukan menggunakan metode *bagging*. Perbedaan utama dari kedua metode ini terletak pada penambahan tahapan *random sub-setting* sebelum di setiap kali pembentukan pohon.

Secara sederhana, algoritma pembentukan RF dapat disebutkan sebagai berikut. Andaikan gugus data training yang kita miliki berukuran n dan terdiri atas d peubah penjelas (predictor). Tahapan penyusunan dan pendugaan menggunakan RF adalah:

1. a. (tahapan *bootstrap*) tarik contoh acak dengan permulihan berukuran n dari gugus data training
   b. (tahapan *random sub-setting*) susun pohon berdasarkan data tersebut, namun pada setiap proses pemisahan pilih secara acak $m < d$ peubah penjelas, dan lakukan pemisahan terbaik
   c. ulangi langkah a-b sebanyak k kali sehingga diperoleh k buah pohon acak
2. lakukan pendugaan gabungan berdasarkan k buah pohon tersebut (misal menggunakan *majority vote* untuk kasus klasifikasi, atau rata-rata untuk kasus regresi)

Proses penggabungan nilai dugaan dari banyak pohon yang dihasilkan serupa dengan yang dilakukan pada metode *bagging*.

Perhatikan bahwa pada setiap kali pembentukan pohon, kandidat peubah penjelas yang digunakan untuk melakukan pemisahan bukanlah seluruh peubah yang terlibat namun hanya sebagian saja hasil pemilihan secara acak. Bisa dibayangkan bahwa proses ini menghasilkan kumpulan pohon tunggal dengan ukuran dan bentuk yang berbeda-beda. Hasil yang diharapkan adalah kumpulan pohon tunggal memiliki korelasi yang kecil antar pohonnya. Korelasi kecil ini mengakibatkan ragam dugaan hasil RF menjadi kecil (Hastie *et al*, 2008) dan lebih kecil dibandingkan ragam dugaan hasil bagging (Zu, 2008).

Lebih jauh Zu (2008) menjelaskan bahwa dalam Breiman (2001) telah dibuktikan batasan besarnya kesalahan prediksi oleh RF adalah

$$\varepsilon_{RF} \le \overline{\rho}\left(\frac{1-s^2}{s^2}\right)$$

dengan $\overline{\rho}$ adalah rata-rata korelasi antar pasangan dugaan dari dua pohon tunggal dan s adalah adalah rata-rata ukuran kekuatan (*strength*) akurasi pohon tunggal. Nilai s yang semakin besar menunjukkan bahwa akurasi prediksinya semakin baik. Definisi formal mengenai s dapat dilihat di Breiman (2001).

Pertidaksamaan tersebut mengarahkan bahwa jika ingin memiliki RF yang memuaskan maka haruslah diperoleh banyak pohon tunggal dengan $\overline{\rho}$ yang kecil dan s yang besar. Apa yang dapat dilakukan selanjutnya?

Jika kita bahwa melihat secara seksama algoritma pembentukan RF, salah satu yang bisa kita ubah adalah nilai m, yaitu banyaknya peubah penjelas yang digunakan sebagai kandidat pemisah dalam pembentukan pohon. Nilai m yang semakin besar akan menyebabkan korelasi ($\overline{\rho}$) semakin besar. Contoh ekstrim adalah jika kita gunakan $m = d$ yang menyebabkan setiap kali pengulangan akan menghasilkan pohon yang sama sehingga nilai korelasi akan menjadi maksimum yaitu sebesar 1.

Namun, jika nilai m kita buat sekecil mungkin yaitu hanya 1 peubah penjelas saja yang dijadikan kandidat pemisah, maka pohon yang diperoleh akan menjadi pohon dengan akurasi yang sangat rendah atau nilai s yang kecil.

Dengan demikian jelas bahwa pemilihan m memegang peranan dalam menentukan kebaikan RF yang dihasilkan. Salah satu paket yang populer digunakan untuk menghasilkan RF adalah randomForest package di R. Dalam paket tersebut, untuk kasus klasifikasi besarnya nilai default bagi m adalah $\lfloor\sqrt{d}\rfloor$ dan untuk kasus regresi sebesar $\lfloor d/3\rfloor$ dengan $\lfloor x\rfloor$ adalah bilangan bulat terbesar yang lebih kecil dari x. Namun demikian Hastie et al. (2008) menyarankan pengguna untuk juga mencoba menggunakan nilai m karena ada kemungkinan memperoleh RF dengan prediksi yang lebih baik.

Boosting

Meskipun ide boosting sudah berkembang sebelumnya dan selanjutnya sudah mengalami banyak modifikasi, namun yang paling populer adalah metode yang disebut sebagai AdaBoost yang diperkenalkan oleh Freund (1995) dan selanjutnya diberi nama AdaBoost.M1 oleh Freund dan Schapire (1996).

Berbeda halnya dengan bagging dan random forest yang mendapatkan banyak pohon dari anak gugus data yang berbeda-beda hasil dari proses bootstrap, metode ini bekerja selalu dengan gugus data yang sama. Setiap kali pembuatan pohon, data yang digunakan tetap seperti semula tetapi memiliki sebaran bobot yang berbeda dalam tiap iterasi. Penggunaan bobot juga dilakukan pada saat proses penggabungan dugaan akhir dari

banyak pohon yang dihasilkan. Teknis pemberian bobot akan dijelaskan pada algoritma yang akan dijelaskan selanjutnya.

Freund (1995), Freund dan Schapire (1996) dan Hastie *et al.* (2008) memaparkan algoritma AdaBoost.M1 dengan cara penulisan yang agak berbeda. Andaikan gugus data yang kita miliki terdiri atas *n* pengamatan, dengan *y* sebagai peubah respon yang memiliki *k* kelas. Selanjutnya kita ingin membuat pohon gabungan menggunakan algoritma boosting dari sebanyak *M* iterasi. Secara ringkas, tahapan algoritma tersebut dapat dituliskan sebagai berikut:

1. Penentuan awal bobot setiap pengamatan, yaitu $w_i = 1/n$ untuk semua i = 1, 2, …, *n*
2. Andaikan *m* adalah nomor iterasi, maka untuk m = 1, 2, … *M*, lakukan proses berikut:
   a. susun pohon tunggal dengan memperhatikan bobot sebesar w_i
   b. hitung tingkat kesalahan klasifikasi (catatan: ***I*** adalah fungsi indikator bernilai 1 atau 0, dan diasumsikan $e_m < 0.5$ dan jika tidak maka proses berhenti)

$$e_m = \frac{\sum_{i=1}^{n} w_i I(y_i \neq \hat{y}_i)}{\sum_{i=1}^{n} w_i}$$

   c. hitung α_m sebagai

$$\alpha_m = \frac{1-e_m}{e_m}$$

   d. tentukan bobot yang baru untuk setiap pengamatan menjadi $w_i = w_i \alpha_m$ untuk pengamatan yang salah klasifikasi, sedangkan untuk pengamatan yang diduga dengan tepat maka bobotnya tetap

3 dugaan akhir adalah kelas *k* yang memiliki nilai terbesar dari

$$\sum_{m:\hat{y}_i^m = y_i} \log \alpha_m$$

Perhatikan bahwa karena nilai $e_m > 0.5$ maka α_m akan selalu lebih dari satu. Dengan demikian pada setiap penambahan iterasi, bobot dari pengamatan yang mengalami salah klasifikasi akan membesar. Ini adalah ide dasar dari boosting, yaitu bahwa pada setiap iterasi model pohon memberikan bobot yang lebih besar kepada data yang "lebih sulit". Dengan kata lain, pada iterasi berikutnya pohon yang dibentuk akan terkonsentrasi untuk menduga dengan lebih baik pada pengamatan yang gagal diklasifikasikan dengan benar pada pohon sebelumnya. Friedman *et al*. (2000) dalam Zu (2008) menjelaskan tentang kunci keberhasilan pemilihan peningkatan bobot menggunakan α_m.

Selanjutnya Hastie *et al.* (2008) menjelaskan bahwa untuk kasus peubah respon adalah peubah dengan hanya dua kelas dan tarafnya dinotasikan -1 dan +1, maka dugaan kelas akhir dapat dinyatakan sebagai fungsi:

$$sign\left[\sum_{m=1}^{M} \hat{y}_m \log \alpha_m\right]$$

Bentuk persamaan terakhir yang berupa total terboboti hasil dugaan dari setiap pohon ini memperjelas pengelompokan metode *boosting* sebagai salah satu metode penggabungan (*ensemble*).

Freund dan Schapire (1996) melaporkan hasil percobaannya yang membandingkan pohon tunggal dengan metode C4.5, pohon C4.5 gabungan dengan metode bagging, dan pohon gabungan dengan metode boosting. Ketiga metode diterapkan pada 27 gugus data berbeda. Dibandingkan dengan pohon tunggal, boosting mampu memberikan perbaikan tingkat salah klasifikasi sebesar 24.8% sedangkan perbaikan oleh metode bagging sebesar 20%.

KESIMPULAN

Berbagai literatur mencatat bahwa metode pohon gabungan mampu bekerja dengan baik dan memberikan dugaan yang lebih tinggi akurasinya dibandingkan pohon tunggal. Menimbang bahwa software-software *open-source* juga banyak yang menyediakan fungsi untuk menghasilkan analisis pohon gabungan ini, maka tidak berlebihan kiranya jika penulis manyebutkan bahwa metode ini layak untuk menjadi alternatif bagi pengguna pohon tunggal. Akurasi dugaan yang tinggi pada pohon gabungan relatif dapat diperoleh dengan mudah, dibanding menerapkan metode-metode klasifikasi yang lebih rumit seperti *support vector machine* (SVM) atau *kernel classification* (lihat diskusi di tulisan Zhu (2008)).

Salah satu yang mungkin akan terasa kurang menarik pada penggunaan metode ini dibandingkan pohon tunggal adalah visualisasi hasil dan interpretasi. Namun bagi mereka yang mementingkan hasil prediksi, hal tersebut tidak menjadi masalah berarti.

DAFTAR PUSTAKA

Berk RA. 2008. *Statistical Learning from a Regression Perspective.* New York : Springer Science + Business Media,

Bramer M. 2007. *Principles of Data Mining.* London.: Springer-Verlag,

Breiman L. 1996. Bagging Predictors. *Machine Learning* **24**: 123–140

Breiman, L. 2001. Random Forests. *Machine Learning* 45: 5–32.

Freund,Y, Schapire R E. 1996. Experiments with a new boosting algorithm. *Machine Learning*: Proceedings of the Thirteenth International Conference, Morgan Kauffman, San Francisco:. 148–156.

Freund Y, Schapire R E. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, **55**:119-139.

Gashler M, Giraud-Carrier C, Martinez T. 2008. "Decision Tree Ensemble: Small Heterogeneous Is Better Than Large Homogeneous". *Seventh International Conference on Machine Learning and Applications* (ICMLA 08). pp. 900–905

Hastie TJ, Tibshirani RJ, Friedman JH. 2008. *The Elements of Statistical Learning: Data-mining, Inference and Prediction*. Second Edition. New York: Springer-Verlag.

Loh WY, Shih, YS. 1997. Split Selection Methods for Classification Tree. *Statistica Sinica* **7**: 815 – 840

Quinlan JR. 1986. Induction of Decision Tree. *Machine Learning* 1: 81 – 106.

Rokach L. 2008. *Ensemble Methods For Classifiers* dalam *Data Mining and Knowledge Discovery Handbook* (editor Maimon O. and Rokach L.). New York : Springer Science+Business Media.

Salzberg SL. 1994. Book Review: C4.5: Program for Machine Learning by J. Ross Quinlan, Morgan Kaufmann Publishers, Inc. 1993. *Machine Learning* 16: 235 – 240.

Zhu M. 2008. Kernels and Ensembles: Perspectives on Statistical Learning. *The American Statistician* 62: 97 – 109.