Qu'est ce qu'un algorithme en boîte noire ? Tractatus des décisions algorithmiques

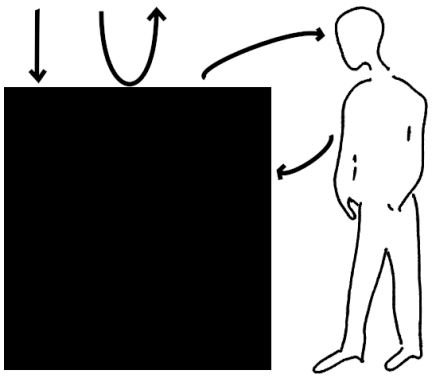
Erwan Le Merrer, Gilles Trédan

Qu'est ce qu'un algorithme en boîte noire ?

Tractatus des décisions algorithmiques

v1.0

Erwan Le Merrer, Inria
Gilles Trédan, CNRS



Les entreprises ou institutions (Divinités) désintermédient leurs relations avec les utilisateurs via des algorithmes décisionnels (Pythies) : les usagers (mortels) se voient proposer des décisions arbitraires (oracles) lors de leurs interactions. Comment appréhender ces algorithmes en boîte noire, du point de vue d'un usager ou regulateur ?

Algorithmes

1. Une donnée est la représentation numérique d'un objet, ou le produit d'un traitement sur des variables numériques.
2. Un algorithme décrit un ensemble d'opérations à exécuter, pour résoudre un problème déterminé.
3. Un algorithme dit de traitement prend une ou plusieurs données (ses entrées), et effectue des opérations sur celles-ci, pour produire d'autres données (ses sorties).
4. Un algorithme est implémenté (*i.e.*, mis en oeuvre) via une description par un langage de haut niveau, appelé code source, intelligible (*i.e.*, *compilable*, puis exécutable par une machine).
5. Un même algorithme peut être implémenté de plusieurs manières différentes mais équivalentes (*i.e.*, les variantes produisent les mêmes sorties).
6. La divulgation (ouverture) du code source implémentant un algorithme revient à exposer l'intégralité des opérations effectuées par celui ci.
7. Les algorithmes peuvent recourir à des variables. Ces variables sont fixées au début de son exécution, ou au cours de son exécution.
8. Les algorithmes peuvent recourir à de l'aléa; ils sont alors dits non déterministes ou aléatoires. Ils sont déterministes dans le cas contraire.

(a) Un algorithme déterministe produit les mêmes sorties pour les mêmes entrées, toutes variables étant égales par ailleurs.

9. Les algorithmes dits non explicites effectuent des opérations complexes sur les données, c'est-à-dire non descriptibles par un langage de haut niveau.
10. Un algorithme ou une implémentation est dit biaisé s'il produit une sortie qui diverge de l'attente. On dit qu'il est *buggé* si ce biais est involontaire.
11. L'interaction entre algorithmes via leurs entrées/sorties compose un système complexe.

(a) La dynamique de cette interaction et sa sortie sont dites émergeantes.

Décisions algorithmiques

12. L'espace des entrées (resp. sorties) est l'ensemble des combinaisons de valeurs que les données peuvent prendre en entrée (resp. sortie).

(a) La taille de l'espace des entrées (resp. sorties) est le nombre de combinaisons possibles de celles-ci.

13. Exécuter un algorithme déterministe sur l'ensemble de ses entrées (avec variables et aléas fixés), permet de caractériser intégralement cet algorithme (table de correspondance).

(a) Cette méthode n'indique toutefois en rien comment retrouver (*i.e.*, retro-engénierer) cet algorithme.

14. Ouvrir le code d'un algorithme permet de définir l'espace de ses entrées et sorties.

(a) Un algorithme dont le code source est ouvert ne suffit pas à décrire un traitement précis, en l'absence de ses données d'entrée, variables et sources d'aléa.

15. Un algorithme dit décisionnel est comme tout autre algorithme de traitement: sa sortie prend une valeur parmi un ensemble prédéfini ; mais cette valeur est alors appelée décision et proposée à l'usager.

(a) Un algorithme décisionnel peut alors tenter d'approcher au plus près (*i.e.*, d'encoder) un jugement humain rationnel débouchant sur une décision.
(b) Un algorithme décisionnel peut très bien être purement aléatoire et ne prendre aucune donnée en entrée; c'est alors un générateur aléatoire de décisions (binaires par exemple).

16. Une décision est dite biaisée si elle est produite causalement par des données qui ne devraient pas conduire à celle-ci.

Explication des décisions algorithmiques

17. La possibilité de manipuler librement un algorithme, et ainsi de connaître chacune de ses opérations, et d'interagir via ses entrées et d'observer ses sorties, permet de le qualifier de boîte blanche.
18. L'implémentation d'un algorithme ne peut être "transparente" que par sa mise à disposition en tant que boîte blanche. En effet, le code source seul n'explique pas les décisions mais les contraint.
19. L'explication d'une décision est une projection sur le langage des instructions de haut niveau d'un algorithme ayant amené à cette décision particulière.
20. Une décision est dite non explicite, si les opérations menant des données d'entrées à cette décision ne peuvent être décrites par un langage de haut niveau.
21. Une explication est sous déterminée, si elle ne décrit qu'une partie des opérations ayant mené à la décision.
22. Faire varier les entrées afin de faire varier les décisions permet de quantifier l'impact relatif de ces entrées.
23. Les décisions sont identiques à l'intérieur de frontières dont chacune donne sur une autre décision.

(a) Un point sur une frontière de décision est défini comme un endroit précis ou une modification infime de l'entrée fait basculer la décision.

(b) Les frontières décisionnelles d'un algorithme décisionnel sont constituées de l'ensemble des points précédemment définis.

Algorithmes en boîte noire

24. Un algorithme en boîte noire est dit observable si tout changement de ses variables transparaît dans ses sorties.
25. Un algorithme décisionnel est dit opaque pour un individu si ce dernier ne peut pas accéder à un seul des composants suivants : code source, entrées ou variables.
26. L'exécution d'un algorithme sur une machine contrôlée par un observateur n'est en rien la garantie de la nature sa boîte blanche pour ce même observateur (code fermé, enclaves, chiffrement).

(a) Cela est d'autant plus évident pour une exécution sur une machine tierce (cf "*There is no cloud, just someone else's computer*").

27. Un algorithme en boîte noire est un algorithme pour lequel un utilisateur ne peut simplement que proposer des données en entrée et observer les sorties qui résultent *à priori* causalement de ces entrées.

(a) *A priori*, car un algorithme peut tout à fait répondre par une valeur fixe ou aléatoire quelles que soient ses entrées.
(b) En d'autres termes, le voile qui fait passer un algorithme en boîte blanche à noire est l'exécution par un tiers, ou les multiples techniques d'obfuscation du code. Autrement dit, même un algorithme connu (boîte blanche) lorsque exécuté par un tiers devient une boîte noire.

28. Une explication par un algorithme en boîte noire est falsifiable, car il existe toujours une explication validant la sortie, qui n'est pas l'explication des opérations réellement effectuées sur les entrées (problème du videur).
29. Appelons hypothèse de stabilité le fait qu'un algorithme en boîte noire et ses variables restent fixes durant une période d'interaction.
30. Sous hypothèse de stabilité, deux algorithmes en boîte noire sont indistinguables si toutes leurs sorties sont égales pour toutes leurs entrées respectives.

(a) Deux objets en boîte noire, et qui sont indistinguables, peuvent être le même algorithme (leur code source peut en effet diverger), ou deux algorithmes différents (*e.g.*, un algorithme de tri à bulles et un algorithme de tri fusion); leurs sorties sont attendues identiques.
(b) Deux algorithmes sont distinguables s'il existe deux sorties différentes pour au moins une entrée donnée.
(c) La distinguabilité de deux algorithmes peut ne tenir alors qu'à une seule entrée particulière.

31. Tout comme dans le domaine quantique, une simple interaction avec un algorithme en boîte noire afin d'observer sa sortie peut tout à fait modifier l'état des variables de celui-ci. Ne serait ce que par ré-apprentissage de cet algorithme sur l'entrée proposée.

(a) Tout espoir de réitération d'une observation est ainsi potentiellement vain. Dans le pire cas, l'observation est "à un coup".
(b) Le cas plus favorable de boîtes noires stables car inchangées par l'observation est également envisageable.
(c) L'acte de fournir volontairement des données particulières en entrée, de manière à modifier dans une certaine direction l'état interne de l'algorithme en boîte noire –et en conséquence, ses sorties– , est appelé empoisonnement des données.

32. La taille de l'espace des entrées est en relation directe avec la difficulté d'observer des propriétés d'un algorithme en boîte noire. Plus cette taille augmente, plus le fléau de la dimension est prégnant, et plus il est complexe d'observer un algorithme en boîte noire.

33. L'apprentissage du résultat de plusieurs couples d'entrées/sorties peut permettre une extra/interpolation permettant des prédictions sur les sorties de la boîte noire.

34. Un algorithme peut très bien spécialiser ses sorties (*e.g.*, créer un biais volontairement) en ne tenant compte que d'une seule information en entrée ("mise en bac à sable", *i.e.*, *sandboxing*).

35. Les réseaux de neurones, piliers des avancées modernes en apprentissage automatique, sont des boîtes noires car ils produisent des décisions non explicites.

(a) Un processus d'apprentissage procède par des milliards d'essais-erreurs, et se conclue par le positionnement des variables de l'algorithme aux valeurs qui ont minimisé l'erreur finale de celui-ci sur des cas connus.
(b) Ces variables étant en quantité gigantesque, et en l'absence d'étapes algorithmiques, il n'y a donc pas d'explication directe possible des décisions qui résultent.
(c) Il en résulte une compréhension à tâtons des éléments numériques de la fonction apprise qui ont amené à un certain résultat sur un ensemble donné d'entrées-sorties (interprétation).

L'algorithme (en boîte blanche) côté tiers

36. Nous appelons tierce l'entité qui exécute un algorithme en boîte noire pour des utilisateurs.

(a) La relative nouveauté et la généralisation de la dichotomie courante boîte blanche/boîte noire découle de la mise en production d'algorithmes décisionnels en contact direct avec les utilisateurs. L'usage antérieur voyait plutôt les algorithmes à des fins de conseil d'une

institution, qui elle faisait l'intermédiaire entre cet algorithme et les utilisateurs. La désintermédiation incite ainsi à la volonté de compréhension de ce qui est faisable ou intelligible dans ce nouveau type d'interaction.

(b) Le placement d'un code source par un tiers sur un serveur public pour son inspection, ne peut impliquer aucune conclusion de transparence. En effet, et trivialement, rien en prouve que c'est bien de près ou de loin l'algorithme qui s'exécute réellement dans le service en question.

Audit algorithmique

37. Une classe d'audit algorithmique cherche à établir la validité ou non dans un algorithme en boîte noire de prédicats définis au préalable ; *e.g.*, pour telle entrée, telle sortie ne doit pas être observée.

38. Sous hypothèse de stabilité, une autre classe cherche à établir des tendances à l'oeuvre dans la relation entrées-sorties (modèle substitut).

(a) Plus la quantité de relations entrées-sorties observée est grande, plus il est possible de construire une image précise de l'algorithme (*i.e.*, l'approximer finement). Ainsi, pour un budget d'entrées tendant vers la taille de l'espace d'entrée, la fonction peut être approximée avec une erreur arbitrairement faible.

39. La maxime de Shannon (voir également Kerckhoffs), qui précise qu'il faut faire l'hypothèse que "l'adversaire (*i.e.*, l'observateur) connaît le système (*i.e.*, l'algorithme ici)", semble peu à l'oeuvre en pratique dans le domaine : les tiers s'appuient largement sur de la sécurité par l'obscurité.

Raccourcis

40. Qu'est un algorithme en boîte noire pour lequel je peux prédire toutes les sorties ? Un algorithme dont je possède un substitut parfait, au sens de son indistingabilité totale sur toutes les entrées possibles (30a) d'avec ce substitut.

41. Un algorithme en boîte noire tombe sous la définition d'un logiciel propriétaire, car il ne permet pas ne serait-ce qu'un seul des quatre points définissant le logiciel libre.

(a) Soit l'impossibilité d'exécuter, de distribuer, d'étudier, ou de modifier ces logiciels.

Implications et distorsions sociétales

42. Un algorithme est également dit biaisé s'il produit des décisions considérées comme "injustes" par un groupe d'individus.

(a) La notion d'équité pour un algorithme décisionnel voudrait mesurer l'équilibre des décisions pour des groupes d'individus représentés par des entrées spécifiques.
(b) Le principe d'un algorithme de classification est de discriminer des groupes d'individus à partir des entrées.
(c) Un algorithme de classification sera considéré comme juste ou injuste selon l'intersection de ses frontières avec un ensemble de frontières socialement construites.

43. Le doute que fait planer la possibilité de la mise en "bac à sable", sur toute tentative d'audit, s'illustre de façon criante.

(a) "Diesel gate": un auditeur habilité était mis face à une réalité alternative (discrimination sur entrée "auditeur"), c'est à dire à une spécialisation des sorties en vue de l'obtention d'une certification, alors que le reste des usagers faisaient l'expérience d'une toute autre réalité.
(b) Les pratiques de bannissement furtif (shadow banning) proposent l'illusion à un utilisateur d'être normalement traité, alors que le tiers a lui décidé un abaissement de la visibilité de celui-ci (sur un réseau social par exemple).
(c) Il en va de même pour les propositions de tiers d'ouverture de leurs interfaces (APIs) pour audit leurs algorithmes. Aucune garantie en pratique que les sorties soient celles qu'un utilisateur lambda pourrait observer.

44. Un algorithme biaisé n'est pas fautif en tant que tel. Il a été placé face aux utilisateurs sur décision du tiers, qui l'a commandité. Ainsi la question de la faute incombe transitivement et naturellement à ce dernier.

45. Les pratiques des lanceurs d'alerte peuvent trouver une légitimité sociétale simplement dans l'exhibition des sorties jugées impropres par un algorithme en boîte noire (*e.g.*, Tay bot). Même si certains avaient volontairement et abusivement fabriqué les entrées qui allaient amener à ce comportement impropre.

(a) La question n'est alors pas ici un problème dans la correspondance entrées-sorties, mais plutôt dans l'absence de régulation/censure sur certaines sorties par l'algorithme (*accountability*, *i.e.*, responsabilité du tiers).
(b) Autrement dit, certaines sorties ne devraient jamais être proposées, par construction/contrainte.

46. Les systèmes de classement ou de recommandation sont des algorithmes décisionnels qui filtrent et classent des objets d'un catalogue

de grande dimension à destination d'un utilisateur (personnalisation).
47. Une bulle de filtre est le résultat de la sur-personnalisation d'un algorithme de recommandation quant à ses sorties à destination d'un utilisateur en particulier.

(a) Un corollaire est l'effondrement de la diversité des propositions faites par l'algorithme à cet utilisateur.
(b) Un cas particulier (*rabbit-hole*) est le fruit de l'interaction croissante d'un utilisateur avec la personnalisation qui lui est faite par un algorithme, créant ainsi un phénomène d'amplification de la réaction algorithmique.

48. Les algorithmes dits prédictifs regroupent les techniques dont le but est de maximiser une métrique (telle que l'appréciation des utilisateurs de ce qui leur est proposé) sur le comportement futur des utilisateurs face à leurs propositions (*i.e.*, sorties).

(a) Un algorithme ne sait pas ce que veut un utilisateur. Pour produire cette illusion, il peut exploiter ce que d'autres utilisateurs –qu'il juge semblables– ont choisi par le passé. Autrement dit : les algorithmes dits prédictifs ne prévoient pas le futur, ils encodent le passé pour le rejouer.
(b) Connaissant les nombreux biais qui peuvent affliger les algorithmes, un travers commun est de prendre la décision algorithmique comme une prédiction fiable et de l'imposer à tous en la justifiant comme indiscutable scientifiquement.

49. Les tenants de l'existence d'une vie privée en ligne suggèrent qu'il est possible d'interagir avec des boîtes noires via une partie des données personnelles des utilisateurs, mais que l'autre partie restera secrète, voire non inférable.

(a) De par les masses de données disponibles pour l'apprentissage des algorithmes et la précision de leurs inférences, la possibilité de fournir des entrées qui ne permettent pas une inférence correcte du reste des données "cachées" devient négligeable.