

3.4 La résolution des problèmes non-linéaires par les techniques itératives

Les méthodes itératives pour l'optimisation non linéaire sont des approches algorithmiques qui visent à trouver la solution optimale d'un problème non linéaire en utilisant un processus itératif. Le principe de ces méthodes est le suivant :

Initialisation : Choisissez un point de départ initial, généralement noté X_0 , qui peut être n'importe quelle valeur admissible. Plusieurs méthodes itératives nécessitent également une valeur initiale pour les paramètres de l'algorithme.

Itérations : À partir du point de départ, effectuez des itérations successives pour améliorer la solution. Le but est de se rapprocher de la solution optimale.

Direction de Recherche : À chaque itération, une direction de recherche est déterminée pour se déplacer vers une solution potentielle meilleure. Cette direction est souvent calculée à l'aide du gradient, de la dérivée seconde (Hessien), ou d'autres informations sur la fonction objectif.

Pas de Mise à Jour : Une fois que la direction de recherche est déterminée, un pas est effectué le long de cette direction pour mettre à jour la solution courante. Le choix de la taille du pas peut être crucial pour la convergence de l'algorithme.

Critère d'Arrêt : Les itérations se poursuivent jusqu'à ce qu'un certain critère d'arrêt soit satisfait. Ce critère peut être basé sur la convergence du gradient, la stabilité de la solution, le nombre d'itérations, ou d'autres mesures.

Évaluation de la Fonction Objectif : À chaque itération, la valeur de la fonction objectif est évaluée pour déterminer si l'algorithme se rapproche d'une solution optimale.

Actualisation de la Solution : La solution courante est mise à jour à chaque itération si une solution meilleure est trouvée.

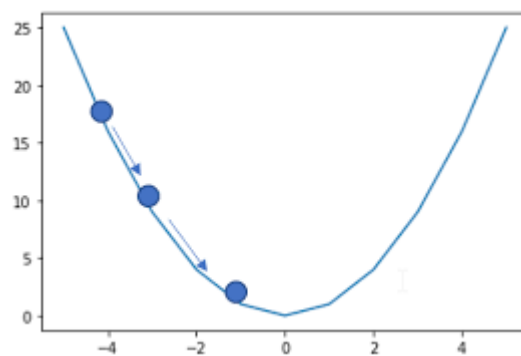
Convergence : L'algorithme continue de s'itérer jusqu'à ce qu'il converge vers une solution qui satisfait les critères d'arrêt. L'algorithme peut converger vers un minimum local ou global en fonction des méthodes et des propriétés de la fonction objectif.

Les méthodes itératives couramment utilisées pour l'optimisation non linéaire comprennent la méthode de Newton-Raphson, la descente de gradient (ou méthode de gradient), les méthodes de quasi-Newton, les méthodes de recherche linéaire, et bien d'autres. Chacune de ces méthodes utilise un principe similaire d'itération pour se rapprocher de la solution optimale, mais elles diffèrent dans leur manière de déterminer la direction de recherche et la taille du pas.

L'efficacité et la convergence de ces méthodes dépendent souvent de la qualité de l'initialisation, de la nature de la fonction objectif, et du choix approprié des paramètres. Les méthodes itératives sont couramment utilisées pour résoudre des problèmes d'optimisation non linéaire dans de nombreux domaines, tels que l'ingénierie, la finance, la science des données, et bien d'autres.

3.4.1 l'algorithme de la descente du gradient

L'algorithme de la descente du gradient est une méthode itérative largement utilisée pour l'optimisation non linéaire multivariable. Il est principalement utilisé pour trouver un minimum (ou maximum) local d'une fonction objective. Voici le principe de l'algorithme de la descente du gradient pour l'optimisation non linéaire multivariable :



Principe de la Descente du Gradient :

1. **Initialisation** : Sélectionnez un point initial x_0 comme point de départ pour l'optimisation.
2. **Itérations** : Répétez les étapes suivantes jusqu'à ce qu'un critère d'arrêt soit satisfait (par exemple, jusqu'à ce que la variation de la fonction objective devienne suffisamment petite) :
 - a. **Calcul du Gradient** : Calculez le gradient de la fonction objectif par rapport aux variables de décision au point courant x_n . Le gradient est un vecteur de dérivées partielles, noté $\nabla f(x_n)$
 - b. **Direction de Recherche** : Choisissez une direction de recherche, généralement opposée au gradient pour minimiser la fonction. La direction est souvent définie comme $-\nabla f(x_n)$, mais d'autres méthodes, telles que la méthode du gradient conjugué, peuvent être utilisées.
 - c. **Taille du Pas** : Déterminez la taille du pas (α_n) le long de la direction de recherche. Il s'agit d'un facteur d'échelle qui affecte la distance parcourue à chaque itération. Il peut être déterminé par différentes méthodes, telles que la recherche linéaire ou des règles heuristiques.

d. **Mise à Jour de la Solution** : Mettez à jour la solution courante en déplaçant x_n le long de la direction de recherche avec la taille du pas calculée :

$$x_{n+1} = x_n - \alpha_n \nabla f(x_n)$$

3. **Critère d'Arrêt** : Arrêtez les itérations lorsque l'un des critères d'arrêt est satisfait, tels que la convergence du gradient, le nombre maximal d'itérations, ou une tolérance sur la variation de la fonction objectif.
4. **Sortie** : La solution x_n à la fin des itérations est une approximation du minimum local de la fonction objective.

L'algorithme de la descente du gradient

Input : min F(x)

Initialisation

Set $x_0 = (,,,,)$

Eps = 10^{-6}

Repeat

- Compute the gradient of F in x_n Grad(F(x_n))
- Set the step a *exactly or approximatively*
- $x_{n+1} = x_n - a * \text{Grad}(F(x_n))$

Until convergence ($\|\text{Grad}(F(x_n))\| \leq \text{eps}$)

Calcul du pas dans l'algorithme de la descente du gradient

Le choix du pas, souvent noté par (γ) ou (α), dans l'algorithme de descente de gradient est une étape cruciale. Il existe plusieurs méthodes pour le faire :

- **Pas constant** : On peut choisir un pas constant pour toutes les itérations. Cependant, cela peut ne pas être optimal car le pas idéal peut varier à chaque étape.
- **Recherche linéaire** : On peut utiliser une recherche linéaire pour trouver le pas qui minimise la fonction objectif le long de la direction du gradient¹. Cela implique généralement de résoudre un problème d'optimisation à une dimension à chaque étape, ce qui peut être coûteux en calcul.

- **Condition d'Armijo-Goldstein** : Une autre approche consiste à utiliser la condition d'Armijo-Goldstein1. On commence par un pas initial, puis on le réduit jusqu'à ce que la condition soit satisfaite. La condition d'Armijo-Goldstein est une condition sur la diminution de la fonction objectif .

$$(f(x + \gamma v) \leq f(x) + c\gamma \nabla f(x)^T v)$$

où:

(f) est la fonction objectif,

(x) est le point courant,

(v) est la direction de recherche (généralement la direction du gradient négatif),

(γ) est le pas,

(c) est un petit nombre positif (par exemple, 0.1),

($\nabla f(x)$) est le gradient de la fonction objectif au point courant.

Exemple :

$$\text{Min } f(a, b) = a^2 + 3b^2$$

$$\text{grad } f(a, b) = \left(\frac{\partial f}{\partial a}(a, b), \frac{\partial f}{\partial b}(a, b) \right) = (2a, 6b).$$

k	(a_k, b_k)	$\text{grad } f(a_k, b_k)$	$f(a_k, b_k)$
0	(2, 1)	(4, 6)	7
1	(1.2, -0.2)	(2.4, -1.20)	1.56
2	(0.72, 0.04)	(1.44, 0.24)	0.523
3	(0.432, -0.008)	(0.864, -0.048)	0.186
4	(0.2592, 0.0016)	(0.5184, 0.0096)	0.067
5	(0.15552, -0.00032)	(0.31104, -0.00192)	0.024
...			
10	(0.012, $1.02 \cdot 10^{-7}$)	(0.024, $6.14 \cdot 10^{-7}$)	0.00014
...			
20	($7.31 \cdot 10^{-5}$, $1.04 \cdot 10^{-14}$)	($1.46 \cdot 10^{-4}$, $6.29 \cdot 10^{-14}$)	$5.34 \cdot 10^{-9}$

Les avantages et les inconvénients de la descente du gradient

L'algorithme de la descente du gradient est largement utilisé pour l'optimisation, mais il présente à la fois des avantages et des inconvénients.

Avantages de l'algorithme de la descente du gradient :

1. **Efficacité éprouvée** : La descente du gradient est une méthode d'optimisation éprouvée qui a été largement utilisée avec succès dans de nombreux domaines, y compris la recherche opérationnelle, l'apprentissage automatique et l'optimisation mathématique.
2. **Applicabilité** : Elle peut être appliquée à un large éventail de problèmes d'optimisation, qu'ils soient continus, non linéaires, ou en présence de contraintes.
3. **Convergence garantie** : L'algorithme est garanti de converger vers un minimum local de la fonction objective si le taux d'apprentissage est choisi de manière appropriée et si les conditions de régularité sont satisfaites.
4. **Facilité d'implémentation** : La descente du gradient est relativement simple à mettre en œuvre et ne nécessite pas de calculs mathématiques complexes.
5. **Parallélisme** : Il est possible de paralléliser l'algorithme de la descente du gradient, ce qui peut accélérer la convergence sur des systèmes informatiques multicœurs ou distribués.

Inconvénients de l'algorithme de la descente du gradient :

1. **Sensibilité au choix du pas (taux d'apprentissage)**
2. **Minimum local** : La descente du gradient peut converger vers un minimum local de la fonction objective, ce qui signifie qu'elle ne garantit pas la découverte du minimum global, à moins que la fonction soit convexe.
3. **Sensibilité aux conditions initiales** : Les performances de la descente du gradient peuvent dépendre des conditions initiales. Le choix du point de départ peut influencer la convergence.
4. **Convergence lente dans certains cas** : Dans certaines situations, la descente du gradient peut converger lentement, en particulier lorsque la fonction a des régions plates ou des crêtes étroites.
5. **Sensibilité au bruit** : Si la fonction objective est affectée par du bruit ou des erreurs de mesure, la descente du gradient peut être sensible à ces perturbations et peut converger vers des valeurs sub-optimales.
6. **Nécessité de calculer le gradient** : La descente du gradient nécessite le calcul du gradient de la fonction, ce qui peut être coûteux en termes de calcul dans certains cas.

3.4.2. La méthode de newton-Raphson

La méthode de Newton-Raphson, également connue sous le nom de méthode de Newton, est une technique itérative utilisée pour trouver les zéros (ou les solutions) d'une fonction. Elle est couramment appliquée pour résoudre des équations non linéaires, c'est-à-dire des équations dans lesquelles la variable que vous cherchez à déterminer n'est pas nécessairement linéairement liée aux autres variables. Voici l'algorithme de la méthode de Newton-Raphson :

Entrées :

- Fonction objectif $f(x)$ à minimiser.
- Gradient de la fonction objectif $\nabla f(x)$.
- Hessien de la fonction objectif $H(x)$.
- Critère d'arrêt (par exemple, tolérance sur le gradient, nombre maximal d'itérations, etc.).
- Point initial x_0 .

Sortie :

- Solution approximative x qui minimise la fonction.

Algorithme de la méthode de Newton-Raphson :

1. **Initialisation :**

- Définir $k = 0$.
- Sélectionner un point de départ initial x_0 .

2. **Itérations :**

Répéter les étapes suivantes jusqu'à ce qu'un critère d'arrêt soit satisfait :

- Calcul du Gradient :** Calculer le gradient de la fonction au point courant : $\nabla f(x_k)$.
- Calcul du Hessien :** Calculer le hessien de la fonction au point courant : $H(x_k)$.
- Direction de Recherche :** Calculer la direction de recherche en résolvant le système d'équations linéaires $H(x_k)\Delta x = -\nabla f(x_k)$ pour Δx .
- Mise à Jour de la Solution :** Mettre à jour la solution courante : $x_{k+1} = x_k + \Delta x$.
- Critère d'Arrêt :** Vérifier si un critère d'arrêt est satisfait, par exemple, en vérifiant la convergence du gradient, le nombre maximal d'itérations ou une tolérance sur la variation de la fonction.
- Incrémenter :** Augmenter k de 1 : $k = k + 1$.

3. **Sortie :** La solution x_k est une approximation du minimum local de la fonction.

Les avantages et les inconvénients de newton raphson

La méthode de Newton-Raphson est une technique itérative d'optimisation et de recherche de racines de fonctions, largement utilisée dans divers domaines des mathématiques et de la science. Elle présente des avantages et des inconvénients distincts :

Avantages de la méthode de Newton-Raphson :

1. **Convergence rapide** : La méthode de Newton-Raphson est connue pour sa convergence rapide, en particulier lorsque vous êtes proche de la solution. Dans de nombreux cas, elle atteint une précision élevée en quelques itérations seulement.
2. **Adaptabilité à diverses fonctions** : Elle peut être utilisée pour résoudre une variété de problèmes, y compris la recherche de racines de fonctions non linéaires, l'optimisation de fonctions, la résolution de systèmes d'équations non linéaires, et bien plus encore.
3. **Efficacité en dimension élevée** : La méthode de Newton-Raphson peut être adaptée pour résoudre des problèmes en dimension élevée, ce qui en fait un outil précieux pour la recherche opérationnelle et l'optimisation mathématique.
4. **Utilisation du gradient** : Elle utilise non seulement la valeur de la fonction, mais aussi la dérivée (ou le gradient) de la fonction. Cela peut aider à accélérer la convergence, en particulier lorsque le gradient est disponible.

Inconvénients de la méthode de Newton-Raphson :

1. **Sensibilité aux conditions initiales** : La méthode de Newton-Raphson peut être sensible aux conditions initiales. Le choix d'un point de départ inapproprié peut entraîner une divergence ou la convergence vers une racine incorrecte.
2. **Non-convergence** : Dans certaines situations, la méthode de Newton-Raphson peut ne pas converger, en particulier lorsque la fonction est mal conditionnée ou lorsque le point de départ est trop éloigné de la solution.
3. **Calcul de la dérivée seconde** : Pour appliquer la méthode de Newton-Raphson, vous devez calculer la dérivée seconde de la fonction, ce qui peut être complexe ou coûteux en termes de calcul dans certains cas.
4. **Problèmes de singularités** : La méthode de Newton-Raphson peut rencontrer des problèmes lorsque la fonction présente des singularités ou des points de non-différentiable.
5. **Nécessite une bonne approximation initiale** : Pour obtenir de bons résultats, il est souvent nécessaire de fournir une approximation initiale raisonnablement proche de la solution.

Les méthodes de Quasi-Newton

Les méthodes de Quasi-Newton sont des techniques d'optimisation numérique utilisées pour résoudre des problèmes d'optimisation sans avoir besoin de calculer explicitement la matrice hessienne (dérivée seconde) de la fonction objectif. Ces méthodes sont conçues pour être plus efficaces que la méthode de Newton-Raphson classique dans le cas où le calcul de la matrice hessienne est coûteux ou difficile.

Les méthodes de Quasi-Newton tirent leur nom de la matrice d'approximation de la hessienne, appelée la "matrice quasi-Newtonienne". Cette matrice est mise à jour itérativement au fur et à mesure que l'algorithme progresse, ce qui permet de capturer la structure de la fonction objectif localement. Les deux méthodes de Quasi-Newton les plus couramment utilisées sont la méthode de Broyden-Fletcher-Goldfarb-Shanno (BFGS) et la méthode des différences finies limitées (DFP) :

1. **Méthode de Broyden-Fletcher-Goldfarb-Shanno (BFGS) :** La méthode BFGS est l'une des méthodes de Quasi-Newton les plus populaires. Elle est basée sur une approximation de la hessienne inversée. L'idée est de mettre à jour itérativement une estimation de la matrice hessienne en utilisant les informations de la fonction objective et des itérations précédentes. La méthode BFGS est bien connue pour sa rapidité de convergence et sa capacité à gérer des problèmes d'optimisation non linéaires.
2. **Méthode des différences finies limitées (DFP) :** La méthode DFP est une autre méthode de Quasi-Newton qui approxime la matrice hessienne en utilisant une approche différente de celle de BFGS. Elle se base sur la mise à jour itérative de la matrice hessienne inverse en utilisant les informations des itérations précédentes. La méthode DFP est également efficace et largement utilisée pour résoudre des problèmes d'optimisation non linéaires.

Avantages des méthodes de Quasi-Newton :

- Elles sont efficaces pour résoudre des problèmes d'optimisation non linéaires.
- Elles évitent le calcul coûteux de la matrice hessienne.
- Elles sont adaptées aux problèmes à grande dimension.
- Elles peuvent être utilisées dans des situations où le calcul du gradient est le seul calcul possible.

Inconvénients des méthodes de Quasi-Newton :

- Elles ne garantissent pas toujours la convergence globale vers un minimum, mais elles peuvent converger vers un minimum local.
- La performance dépend de la qualité de l'estimation initiale de la matrice hessienne inverse.
- L'implémentation correcte de ces méthodes peut être complexe.

Autres variantes de la méthode Newton Raphson

- **Newton-Raphson modifié:** Approxime le Hessien en le remplaçant par une matrice diagonale, ce qui simplifie le calcul sans perdre trop en précision.
- **Newton incomplet:** Approxime le Hessien en ne calculant que certaines de ses dérivées partielles, réduisant le coût de calcul.
- **Newton à mémoire limitée:** Stocke seulement les dernières valeurs du Hessien calculées, évitant de recalculer sa matrice pleine à chaque itération.
- **Région de confiance:** Fait varier la taille du pas de descente en fonction de la précision du modèle local. Peut mieux gérer les non-linéarités.
- **Newton adaptatif:** Ajuste dynamiquement le degré d'approximation du Hessien en fonction des résultats, alternant entre versions complète/incomplète.

- **SQP (Sequential Quadratic Programming):** Résout une suite de sous-problèmes de programmation quadratique, améliorant la prise en compte des contraintes.
- **Interpolation multivariée:** Interpole le Hessien dans des régions voisines pour une meilleure estimation lorsque peu de données sont disponibles.