Cours IFT6266,
Sélection, comparaison et évaluation de modèles

Sélection de modèle *vs* évaluation d'un modèle

- **sélection**: étant donné un ensemble (parfois fini, parfois infini dénombrable, parfois non-dénombrable) de "modèles" (i.e. un algorithme d'apprentissage spécifique), et un ensemble de données, on voudrait choisir le modèle qui a le plus de chances de bien généraliser (selon une certaine fonction de perte).
- **évaluation**: étant donné un modèle, on voudrait prédire sa performance (au moins moyenne) future.
- Il est clair que si on sait comment bien faire l'évaluation ça nous donne un moyen de faire la sélection. Cependant, certains algorithmes sont plus précis pour évaluer la généralisation mais pas les meilleurs pour la sélection, et vice-versa, car pour la sélection ce n'est pas la valeur qui compte mais le minimum de cette valeur sur un ensemble de modèles.
- Il faut aussi comprendre que l'évaluation nous donne seulement une estimation et qu'il y a une incertitude autour de cette estimation.
- Notons aussi qu'un algo de sélection est par définition un (méta-) algorithme d'apprentissage.
- A cause du biais de sélection (l'erreur du modèle sélectionné, estimée sur les données ayant servi à faire la sélection, est un estimateur optimiste de l'erreur de généralisation) on veut utiliser des données indépendentes pour la sélection et l'évaluation (e.g. séparation des données en ensembles d'apprentissage, de validation (pour la sélection), et de test (pour l'évaluation).
- On utilise parfois le mot modèle pour parler d'une famille d'algorithmes d'apprentissage paramétrée par un ou plusieurs hyper-paramètres de capacité. On utilise aussi parfois le mot modèle pour parler d'un algorithme d'apprentissage spécifique (fonction d'un ensemble d'apprentissage à une fonction). Et on utilise aussi parfois le mot modèle pour parler du résultat de l'apprentissage (la fonction apprise). Attention.
- On peut utiliser l'erreur d'apprentissage pour comparer deux modèles qui ont à peu près la même richesse (plus formellement, capacité ou VC-dimension), et c'est une meilleure utilisation des données. Donc en général la sélection de modèle sert soit à choisir entre différents niveaux de capacité, ou bien à choisir entre des modèles dont la capacité est incomparables car ils sont très différents (e.g., réseau de neurones et arbre de décision). En principe, on pourrait toujours comparer sur l'échantillon d'apprentissage deux modèles très différents en choisissant leur capacités respectives de manière à égaliser l'erreur d'apprentissage, sauf que ca ne correspond pas toujours à la capacité optimale. La méthode que je privilégie est donc de choisir (par sélection de modèle) la capacité optimale pour chaque modèle, puis (toujours sur le même ensemble de validation) choisir entre les deux types de modèles.

Estimateurs de l'optimisme, indépendents des données

- L'optimisme de l'erreur d'apprentissage est la différence entre l'erreur de généralisation (en espérance sur les données de test, mais parfois aussi sur les données d'apprentissage) et l'erreur d'apprentissage (celle du modèle choisi).
- Plusieurs méthodes ont été proposées pour l'estimer, sans utiliser les données elles-même, mais en utilisant de l'information sur la richesse du modèle et le nombre de données n.
  - C_p de Mallow **pour la régression linéaire:**

    $$C_p = MSE + 2\frac{d}{n}\hat{\sigma}_\epsilon^2$$

    où MSE est l'erreur quadratique moyenne d'apprentissage, d est le nombre de paramètres et $\hat{\sigma}_\epsilon^2$ est un estimateur de la variance du bruit (obtenu avec un modèle de faible biais). Cet estimateur est limité à la régression purement linéaire, valide **asymptotiquement** ($n \to \infty$) et si il n'y a **pas de régularisation.**. DONC PAS TRÈS UTILE EN PRATIQUE.
  - Le AIC (*Akaike Information Criterion*) est une généralisation du C_p au cas où le critère d'apprentissage est la log-vraisemblance (encore sans régularisation):

    $$AIC = 2E[-\log P_{\hat{\theta}}(Y)] = 2(E[NLL] + \frac{d}{n})$$

    où $P_{\hat{\theta}}$ est le modèle entraîné aux maximum de vraisemblance, et NLL est la "negative log-likelihood" moyenne d'apprentissage, i.e. $\frac{1}{n}\sum_i -\log P_{\hat{\theta}}(y_i)$. On peut généraliser facilement au cas de probabilités conditionnelles. C'est encore un estimateur asymptotique (mais dans ce cas l'overfitting n'est pas vraiment un problème...) donc qui a tendance à sous-estimer l'erreur de généralisation. ASSEZ LIMITÉ AUSSI AUX MODÈLES LINÉAIRES.
  - On peut généraliser le AIC (donc le C_p) au cas où on a de la régularisation, avec le **nombre effectif de paramètres** (voir le livre).
  - Le BIC (*Bayesian Information Criterion* ou critère de Schwartz est:

    $$BIC = 2(E[NLL] + d\frac{\log n}{2n})$$

    Voir la discussion sur l'inférence Bayesienne au prochain cours. Ce critère est basé sur une approximation de $P(modele|donnees)$. ASSEZ LIMITÉ AUSSI AUX MODÈLES LINÉAIRES.
  - Le critère MDL (*Minimum Description Length*) est basée sur la théorie de l'information mais est en fait très proche du principe de la régularisation. On voit le modèle comme un moyen de "compresser" (résumer) les données. Le principe est de choisir le modèle qui donne la meilleure compression (en tenant compte de l'erreur, c'est à dire de ce qui n'est pas expliqué par le modèle). On prend donc le modèle tel que la "longueur" du modèle plus

la "longueur" des erreurs est minimisée. La "longueur" se mesure en bits. On note que pour compresser optimalement une observation x d'une variable aléatoire X de loi $P(X)$ requière $-\log_2 P(X = x)$ bits. On peut généraliser à des v.a. continues en prenant une précision finie. Ça donne le critère

$$longueur = -\log P(\mathbf{y}|\mathbf{x}, \theta, M) - \log P(\theta|M)$$

où les données d'apprentissage sont $(\mathbf{x}, \mathbf{y})$, θ est le paramètres appris, et M est le modèle. On fait donc en fait de la régularisation avec pénalité $-\log P(\theta|M)$. C'est également une approximation de $P(M|\mathbf{x}, \mathbf{y})$ qu'on utilise avec les approches Bayesiennes.

– VC-dimension: on peut utiliser certaines inégalités de Vapnik basées sur une notion mathématique de la richesse d'une classe de fonction (la VC-dimension). Par exemple pour la classification on peut obtenir la borne probabiliste (avec prob. $\geq 1 - \eta$):

$$Err.gen. \leq Err.appr. + \frac{\epsilon}{2}(1 + \sqrt{1 + 4Err.appr./\epsilon})$$

où

$$\epsilon = a_1(h(1 + \log(a_2 n/h)) - \log(\eta/4))/n$$

avec h la VC-dimension et a_1, a_2 des constantes (prises arbitrairement à 1 dans les expériences de Cherkassky et Mullier 1998). Pour la régression:

$$Err.gen. \leq \frac{Err.appr.}{1 - c\sqrt{\epsilon}}$$

où c est une autre constante (encore prise à 1 par Cherkassky).
CRITÈRE GÉNÉRALEMENT TROP CONSERVATEUR.

Validation croisée

Le problème avec toutes les méthodes ci-haut c'est qu'elles donnent toujours un estimateur biaisé (et parfois fortement) de l'erreur de généralisation. De plus certaines d'entre elles sont fortement liées à un type d'algorithme particulier. **En utilisant les données elles-mêmes on peut obtenir un estimateur (quasiment) non-biaisé, au prix de calculs supplémentaires.**
Il existe plusieurs variantes du principe de validation croisée, qui permet d'estimer l'erreur de généralisation d'un algorithme d'apprentissage pour un certain ensemble de données. Dans tous les cas, on fait la moyenne de plusieurs erreurs de test, mesurées sur des exemples qui n'ont pas été utilisés pour l'apprentissage. La validation croisée peut par exemple être utilisée pour essayer de choisir la capacité "optimale", pour choisir un algorithme d'apprentissage plutôt qu'un autre, ou pour comparer statistiquement deux algorithems.

- *train/test*: une certaine fraction des données est utilisée pour entrainer et le reste pour tester. C'est la méthode la plus simple et la plus rapide, mais elle est peu

précise si on a peu de données (et c'est là que le problème de la précision de l'estimation de l'erreur par validation croisée est plus criant). Cependant les TESTS statistiques pour comparaison entre algorithmes sont très simples dans ce cas.

- *K-fold XV*: on répète la procédure *train/test* pour K différents ensembles de test disjoints dont l'union est l'ensemble original de donnée. L'erreur estimée est la moyenne de toutes les erreurs de tests sur toutes les partitions.

- On peut généraliser la *K-fold XV* ainsi: on crée K partitions aléatoires (en utilisant des permutations aléatoires des données) train/test; pour chacune on entraine sur la partie 'train' et on mesure la généralisation sur la partie 'test'; on fait la moyenne des performances sur tous les tests.

- *validation séquentielle*: les méthodes ci-haut ne s'appliquent pas si les données sont séquentielles (et donc probablement pas i.i.d.). La validation séquentielle procède avec plusieurs phases de *train/test* à chaque fois en entrainant avec les données jusqu'au temps t et en testant avec les données de $t+1$ à $t+s$, et en recommançant avec l'entraînement jusqu'à $t+s$ et test de $t+s+1$ à $t+2s$, etc... L'erreur estimée est encore la moyenne de toutes les erreurs de tests.

Notez que la méthode *K-fold XV* et la méthode de *validation séquentielle* ont une complexité de calcul qui introduit un facteur proportionnel au nombre d'exemples (ou au nombre de partitions K) dans le temps de calcul total...

Bootstrap

- Une statistique est une mesure quelquonque obtenue à partir d'un ensemble de données, comme par exemple (simple) sa moyenne, ou bien, (compliqué) un estimé de l'erreur de généralisation obtenue par validation croisée.

- Une manière générale d'obtenir un estimé des variations qu'une statistique peut subir à cause de notre ensemble d'apprentissage particulier est la méthode du Bootstrap. L'idée de base de la méthode du Bootstrap est la suivante: on va supposer vraie une hypothèse sur nos données (par exemple, elles sont i.i.d.), et on va s'en servir pour générer des ensembles de données alternatifs mais qui sont tout aussi plausibles que l'ensemble original (selon notre hypothèse).

- Avec le bootstrap non-paramétrique, on va supposer que les données sont i.i.d., et on va tirer un nouvel ensemble d'exemples de la même taille l que l'original en faisant l tirages avec remplacement à partir des exemples originaux. Notez que cela est équivalent à substituer la véritable distribution des données par la distribution empirique correspondant à notre ensemble de données.

- On va tirer B tels ensembles de données sur chacun desquels on va mesurer notre statistique (e.g., l'erreur de généralisation estimée par validation croisée), et on va regarder par exemple l'écart-type de ces valeurs (à travers les B différentes

expériences) pour mesurer la précision de la valeur de notre statistique.

COMPARAISONS D'ALGORITHMES: estimation de l'incertitude autour de l'estimateur de généralisation

Dans le cas le plus simple train/test à la base de la plupart des méthodes d'évaluation de la performance de généralisation, le processus dans son ensemble est le suivant. À partir de la distribution P (généralement inconnue), un ensemble d'apprentissage D_1 est tiré. Étant donné D_1, on obtient une fonction f avec un algorithme d'apprentissage. On tire un ensemble de test D_2 de P, et on mesure l'erreur de f sur les exemples de D_2 (en faisant la moyenne des erreurs sur ces exemples). On peut donc considérer deux sources de variation qui font que notre estimé de l'erreur de généralisation est "bruité":

1. le choix particulier des exemples d'apprentissage D_1 (tirés d'une distribution de probabilité inconnue P),

2. le choix particulier des exemples de test D_2.

Dans le cas où on ne s'intéresse qu'à évaluer la qualité de la fonction obtenue après apprentissage, on peut heureusement ignorer la première source de variabilité. Malheureusement, on est souvent intéressé à évaluer la qualité de l'algorithme d'apprentissage (et donc les deux sources sont importantes) plutôt que la qualité de la fonction obtenue après apprentissage.
En réalité on a plutôt le processus suivant:

1. On tire D_1 de P.

2. On estime l'erreur de généralisation des l'algorithmes A_i en utilisant D_1. Soit $\hat{e}(A_i, D_1)$ notre estimateur.

3. On utilise A_i et D_1 pour choisir f_i.

4. On applique f_i sur de nouveaux exemples D_2 (la vraie généralisation!) tirés de P.

Si on est un **usager** de l'algorithme d'apprentissage (celui qui va utiliser ce f_i particulier pour son application), on considère D_1 fixe (donné) et on s'intéresse seulement à la performance en généralisation de f_i. Par contre, si on est un chercheur et qu'on veuille recommander l'utilisation de A_j versus A_i pour des données du genre de celles utilisées, on ne veut pas se limiter à l'incertitude due aux données de test, on veut considérer tout D_1 comme aléatoire.
Pour COMPARER entre plusieurs f_i on voudrait savoir si la différence observée entre deux algorithmes $(\hat{e}(A_i, D_1) - \hat{e}(A_j, D_1))$ est "significative", dans le sens qu'elle indique réellement que f_j fonctionnera mieux que f_i, par opposition à seulement refléter des variations qui pourraient être dues au "bruit", c'est à dire à la taille finie de l'échantillon. Pour cela **il nous faut un estimateur de l'incertitude autour de** $\hat{e}(A_i, D_1) - \hat{e}(A_j, D_1)$. Cela nous permettra de décréter que l'on est sur qu'il vaut mieux utiliser f_j que f_i, et de quantifier notre confiance dans cet énoncé. Comme $\hat{e}(A_i, D_1)$

est généralement une moyenne, on peut raisonnablement approximer sa distribution par une normale (ou une Student). Dans les deux cas la quantité la plus importante pour caractériser l'incertitude est $\sigma^2 = Var[\hat{e}(A_i, D_1) - \hat{e}(A_j, D_1)]$. On ne peut connaître exactement cette variance, mais on peut l'estimer. On pourra ensuite l'utiliser dans le contexte d'un test d'hypothèse, avec l'hypothèse nulle $E[\hat{e}(A_i, D_1) - \hat{e}(A_j, D_1)] = 0$. Sous l'hypothèse de normalité de $\hat{e}(A_i, D_1) - \hat{e}(A_j, D_1)$, on obtiendra la p-valeur correspondant au test classique, par $P(Z > z)$ (test d'un seul côté) ou $P(|Z| > |z|)$ (test des deux côtés), où Z est une variable aléatoire normale et $z = (\hat{e}(A_i, D_1) - \hat{e}(A_j, D_1))/\hat{\sigma}_{ij}$ et $\hat{\sigma}_{ij}^2$ est notre estimateur de $Var[\hat{e}(A_i, D_1) - \hat{e}(A_j, D_1)]$.
Dans le cas où on utilise la validation croisée ou la validation séquentielle les erreurs de tests correspondant à différents blocs sont corrélées entre elles et il n'est pas possible d'obtenir un estimateur non-biaisée de $Var[\hat{e}(A_i, D_1) - \hat{e}(A_j, D_1)]$ (voir l'article par Bengio et Grandvalet, "No Unbiased Estimator of the Variance of K-Fold Cross-Validation", *Journal of Machine Learning Research, volume 5, pages 1089-1105*, 2004, disponible sur `http://www.iro.umontreal.ca/ lisa/pointeurs/grandvalet04a.pdf`).

Le biais peut être pessimiste (on surestime σ^2 et on décrète qu'on ne peut départager A_i et A_j alors qu'on aurait pu, ce qui serait acceptable) ou optimiste (on sous-estime σ^2 et on décrète que A_i est meilleur que A_j alors que c'est faux, ce qui est beaucoup moins acceptable).

Méthodes d'estimation de la variance

- Variance due aux exemples de test dans une expérience train/test.
  Soit un ensemble i.i.d. d'erreurs e_i (i de 1 a l), par exemple dans le cas de la régression $e_i = (y_i - f(x_i))^2$ hors-échantillon. Considérons la moyenne des erreurs

$$\bar{e} = \frac{1}{l} \sum_i e_i.$$

  et l'écart-type de l'échantillon

$$s = \sqrt{\frac{1}{l-1} \sum_i (e_i - \bar{e})^2}.$$

  On sait que $E[\bar{e}] = E[e]$, i.e., $\bar{e}$ est un estimé non-biaisé de la véritable erreur de généralisation espérée $E[e]$. Par la loi des grands nombres, on sait aussi que asymptotiquement ($l \to \infty$),

$$Var[\bar{e}] \to \frac{Var[e]}{l}$$

  ce qui nous donne immédiatement un "écart-type" $\hat{\sigma} = \frac{s}{\sqrt{l}}$ autour de $\bar{e}$.
  NOTE: si e_i est binaire (0 ou 1), une meilleure estimation de l'écart-type est donnée en considérant non pas un modèle Gaussien mais un modèle binomial, ce qui donne

$$\hat{\sigma} = \sqrt{s^2/l} = \sqrt{\bar{e}(1-\bar{e})/l}.$$

De plus, par le théorème centrale limite, $\bar{e}$ converge vers une Normale asymptotiquement, donc on peut faire des test d'hypothèses (par exemple pour vérifier si on peut rejeter l'hypothèse nulle $E[e] = \theta$: il n'y a qu'à voir la masse $P(\bar{e} <= \textit{observation de } \bar{e})$ (selon une distribution normale centrée à θ et d'écart-type $\hat{\sigma}$) en deça de ce seuil pour obtenir la p-valeur de cette hypothèse nulle. On peut raffiner un peu en observant qu'on a en fait affaire à une distribution t de Student (qui est une meilleure approximation que la Normale pour l petit) avec $l-1$ degrés de libertés.

- Dans le cas de la validation croisée l'estimateur ci-haut est biaisé, mais il est néanmoins souvent utilisé dans la littérature d'apprentissage machine. On considère l'ensemble de toutes les erreurs de tests (ou plutôt les différences des erreurs entre les deux algorithmes) pour tous les blocs (toutes les partitions) comme si elles provenaient d'une seule expérience d'apprentissage. On obtient un estimateur biaisé mais qui est simple à à calculer.

- Un autre estimateur, généralement un peu plus fiable (plus souvent pessimiste) est celui obtenu en comparant les moyennes d'erreurs dans chaque bloc d'une validation croisée. On obtient donc la variance échantillonale des moyennes. Soit μ_k la moyenne des différences d'erreur entre nos deux algorithmes sur le k-ieme bloc de test de la validation croisée. Alors on peut estimer σ^2 par $1/(K-1)\sum_k(\mu_{k=1}^K - (1/K)\sum_{i=1}^K \mu_i)^2$ (voir Nadeau et Bengio 1999, NIPS-12, `www.iro.umontreal.ca/lisa/pointeurs/var-err.ps` ou `www.iro.umontreal.ca/lisa/pointeurs/nadeau_MLJ1597.pdf` pour plus de détails et une justification).

- Un bon estimateur, garanti d'être légèrement pessimiste (donc jamais optimiste), est basé sur une idée très simple: si on divise notre échantillon en deux parties D' et D'' aléatoirement, les estimateurs $\hat{e}(A_j, D')$ et $\hat{e}(A_j, D'')$ sont des variables aléatoires indépendantes. L'estimateur qui résulte de ce principe est plus coûteux en calculs (environ un facteur 10 par rapport aux techniques ci-haut qui ne font pas plus de calculs au delà de la validation croisée). L'algorithme est le suivant:

  1. pour $j = 1$ à J (e.g. $J = 5$ ou $J = 10$)
  2. On permute aléatoirement les données dans D_1, que l'on partitionne en D' et D'' de tailles approximativement égales.
  3. On applique la validation croisée séparement sur chaque partie, en calculant les différences de performance entre nos deux algorithmes A_i et A_j: $\mu'_j = \hat{e}(A_i, D') - \hat{e}(A_j, D')$ et $\mu''_j = \hat{e}(A_i, D'') - \hat{e}(A_j, D'')$.
  4. L'estimateur de σ^2 est $(1/(2J))\sum_{j=1}^J(\mu'_j - \mu''_j)^2$.

  Empiriquement cet estimateur apparait donner une plus faible erreur quadratique (comme estimateur de σ^2) que tous ceux ci-haut et que le bootstrap ci-bas. Il plus coûteux en calculs que ceux ci-haut mais comparable ou moins cher que le Bootstrap.

- La méthode du Bootstrap (voir plus haut le principe) permet de simuler tout le processus (quelle que soit notre estimateur de l'erreur de généralisation) et donc d'obtenir en principe un estimé de σ^2 qui tient compte des deux sources de variation. Cela revient à substituer un estimé $\hat{P}$ de P dans le processus ci-haut, par exemple $\hat{P}$ est la distribution empirique i.i.d. (bootstrap non-paramétrique), ou bien $\hat{P}$ est un modèle probabiliste des données (bootstrap paramétrique). Cependant cette méthode a tendance a sous-estimer la véritable variance (donc à donner des tests plus "libéraux" que "conservateurs") car elle est basée sur l'unique échantillon original D_1 (qui n'est pas nécessairement representatif de toute la variabilité présente dans P), et ne tient pas compte des recoupements qui existent entre les différents ensembles d'apprentissage ou entre les différents ensembles de test. De plus cette méthode est assez coûteuse en temps de calcul (on voit généralement B=100 tirages dans les études).

- Dans le cas de la validation séquentielle (à un ou deux niveaux), on peut faire une hypothèse supplémentaire et obtenir un autre type d'estimateur. On peut exploiter l'ordre dans la séquence des erreurs hors-échantillon et supposer que $Cov(\hat{e}_t, \hat{e}_{t-\tau})$ tend vers 0 quand τ augmente, avec $\hat{e}_t$ l'erreur ou la différence d'erreur qui nous intéresse, pour l'exemple de test au temps t. On peut aussi supposer la stationarité des covariances, i.e. $Cov(\hat{e}_t, \hat{e}_{t-\tau}) = \gamma_\tau$, qui ne dépend pas de t mais seulement de l'intervalle τ. La variance totale de la somme des $\hat{e}_t$ est la somme des covariances. Nos deux hypothèses nous permettent d'approximer les $T(T+1)/2$ termes de covariances par seulement M termes correspondants à γ_0, γ_1, ..., γ_{M-1}. On suppose $\gamma_\tau = 0$ pour $\tau \geq M$ (par la première hypothèse). Le choix de M proportionnel à $\sqrt{T}$ est motivé par des considérations asymptotiques (de convergence de l'estimateur). Chaque γ_τ peut être estimé par un estimateur classique de la covariance:

$$\hat{\gamma}_\tau = \frac{1}{T-\tau-1} \sum_{t=\tau+1}^{T} (\hat{e}_t - \bar{e})(\hat{e}_{t-\tau} - \bar{e})$$

où $\bar{e} = (1/T) \sum_{t=1}^{T} \hat{e}_t$ la moyenne empirique. Donc notre estimateur de σ^2 est

$$\hat{\sigma}^2 = \frac{1}{T^2} \sum_{t=1}^{T} \sum_{s=1}^{T} \hat{\gamma}_{|t-s|}$$

en oubliant pas que $\hat{\gamma}_\tau = 0$ pour $\tau \geq M$.