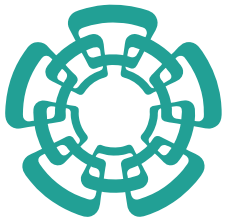

LTI Cinvestav Tamaulipas

Técnicas de validación y remuestreo[1.e.ii-v]

Rafael Pérez Torres

Tópicos Selectos en Reconocimiento de Patrones

Profesor Dr. Wilfrido Gómez Flores

Resumen

El presente documento presenta una descripción de distintas familias de técnicas de validación cruzada y remuestreo, en particular bootstrap. Gracias a este tipo de mecanismos, es posible realizar una mejor definición del modelo a utilizar en la clasificación, intentado obtener los mejores resultados y asegurar la independencia de los mismos respecto a los conjuntos de entrenamiento y clasificación. Asimismo, se presenta una breve metodología para realizar el diseño de una estrategia de clasificación utilizando tres subconjuntos: entrenamiento, validación y clasificación.

1. Introducción

Los métodos de remuestreo y validación cruzada resultan útiles para dos tareas en particular, la selección del modelo y la estimación del performance de

los clasificadores.

En el caso de la selección del modelo, es importante destacar que todas las técnicas de clasificación cuentan con uno o más parámetros que deben ser adaptados. Por ejemplo, el número de vecinos en un clasificador k-nn o el tamaño de la red, parámetros de aprendizaje y pesos en una red neuronal. ¿Cómo poder adaptar dichos parámetros? En el caso de la estimación del performance, es importante destacar que dicha medida debería ser realizada cuando se aplica el clasificador a la población total. ¿Cómo estimar dicho performance?

Si se tuviera acceso a un número ilimitado de muestras, ambas preguntas tendrían una respuesta inmediata: bastaría elegir el modelo que provea la tasa de error más baja en la población completa, y que dicha tasa de error sea igual al valor verdadero. Sin embargo, en el mundo real solamente se tiene acceso a un conjunto finito de muestras que es usualmente menor a la cantidad que se desearñia. Un enfoque aplicable sería utilizar el conjunto de datos completo para seleccionar al clasificador y estimar el error. Sin embargo, este enfoque tiene dos problemas fundamentales:

- El modelo final estaría sobreentrenado, lo cual se hace evidente en modelos con un número grande de parámetros. En la práctica el clasificador fallaría al asignar las etiquetas a las muestras que *no vio* durante la etapa de entrenamiento.
- El error estimado sería marcadamente optimista (menor que el error real).

En general, las técnicas de remuestreo y validación cruzada intentar dar respuesta a este problema a través de mecanismos para realizar la división del conjunto de datos. Por un lado, las técnicas de validación cruzada dividen el conjunto de datos en partes que son independientes una de las otras; en otras palabras, no existen elementos repetidos en cada una de dichas partes. Por otro lado, las técnicas de remuestreo permiten crear particiones en las que los elementos son seleccionados *con reemplazo*, pudiendo aparecer más de una ocasión tanto en el conjunto de entrenamiento como en el de prueba.

El sobreentrenamiento o sobre-ajuste ocurre cuando el tamaño de los datos de entrenamiento es sumamente pequeño o cuando el número de parámetros del modelo es grande, y en general se refiere al hecho de que el clasificador es incapaz de reconocer correctamente muestras que *no haya visto* durante la etapa de entrenamiento.

En base a las particularidades de la división del conjunto de datos, es posible estimar la magnitud del error a obtener por el clasificador. Habitualmente, la selección de los conjuntos y el entrenamiento y clasificación asociados, se realiza en varias intervenciones, obteniendo la media de los resultados como el valor final del error.

2. Técnicas de validación cruzada

Las técnicas de validación cruzada básicamente dividen al conjunto de datos, de tal manera que cada una de las particiones contienen elementos únicos y

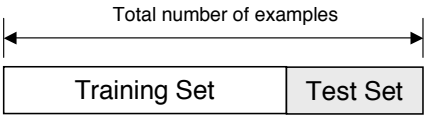

Figura 1: Holdout validation

cada elemento puede aparecer únicamente en una sola partición. La división es aleatoria obteniendo dos conjuntos, uno que se utiliza como el clásico conjunto de entrenamiento, y el otro – de validación – es utilizado para estimar el error de generalización del conjunto de validación. Dado que el objetivo final en el diseño de un clasificador es alcanzar un error bajo de generalización, se entrena al clasificador hasta que se obtiene un valor mínimo de este error. La validación cruzada puede ser aplicada a virtualmente cada método de clasificación, con la salvedad de adaptar el método de validación a las particularidades de cada técnica.

En general, la validación cruzada sólo produce resultados significativos si el conjunto de validación y prueba de conjunto se han extraído de la misma población. Las siguientes son las técnicas de validación cruzada más utilizadas.

2.1. Hold out validation

El método holdout, algunas veces llamado estimación de muestra de prueba, divide a los datos en dos subconjuntos mutuamente exclusivos, denominados conjunto de entrenamiento y conjunto de prueba (o conjunto holdout). Comúnmente, se designa 2/3 del conjunto de datos para el entrenamiento y el 1/3 restante para estimar el error del clasificador entrenado, como se muestre en la Figura 1.

Las principales desventajas del método holdout son referentes a que no en todos los problemas se cuenta con una cantidad suficiente de instancias para poder aislar un subconjunto especial para la prueba. Además, dado que su naturaleza es de un sólo experimento de entrenamiento y prueba, la estimación del error tiende a ser no representativa. En última instancia, la técnica holdout hace un uso ineficiente de los datos, ya que casi un tercio del dataset no es utilizado para entrenar el clasificador. La mayor cantidad de instancias destinadas al conjunto de prueba incrementan el bias de la estimación.

2.2. K-fold cross validation

En la validación cruzada K-fold, también llamada estimación de rotación, el conjunto de datos se divide aleatoriamente en k particiones mutuamente excluyentes. La ejecución del clasificador es realizada k veces en los tiempos $t \in \{1, 2, \ldots, k\}$. Por cada experimento realizado en el tiempo t, se utilizan $k-1$ particiones para el entrenamiento y la restante se destina para la prueba, como se muestra en la Figura 2

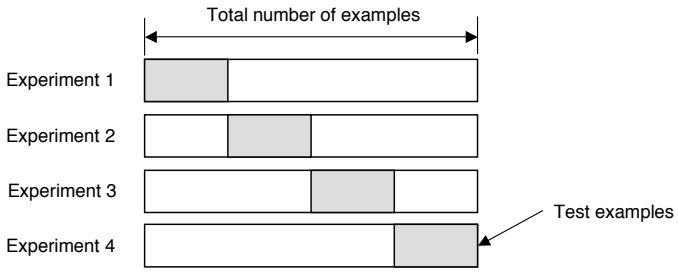

Figura 2: K-fold cross validation

La principal ventaja de la validación cruzada k-fold es que todos los ejemplos del dataset son eventuelmente utilizados para entrenamieto y prueba. El error real puede ser estimado como el error promedio:

$$E = \frac{1}{k} \sum_{i=1}^{k} E_i$$

Cuántas particiones se necesitan?

Un número grande de particiones arrojaría los siguientes resultados:

- ✓El bias del estimador del error real sería pequeño.
- ×La varianza del estimador del error real sería grande.
- ×El tiempo de cómputo sería muy grande (muchos experimentos).

Un número pequeño de particiones obtendría:

- ✓El número de experimentos y tiempo computacional sería reducido.
- ✓La varianza del estimador del error real sería pequeña.
- ×El bias del estimador sería grande

En la práctica, la elección del número de particiones depende del tamaño del dataset. Para datasets grandes, incluso 3-fold cross validation sería preciso. Para datasets dispersos, es aconsejable utilizar leave-one-out para entrenar con la mayor cantidad de ejemplos posibles. Una elección comúm para k-fold cross validation es $k = 10$.

2.3. Leave-one-out cross validation (LOOCV)

Es un caso especial de k-fold cross validation, cuando se indica que K sea el número total de ejemplos. Para un dataset de n muestras, se desarrollan n experimentos. En cada experimento, se utilizan entonces $n - 1$ instancias para

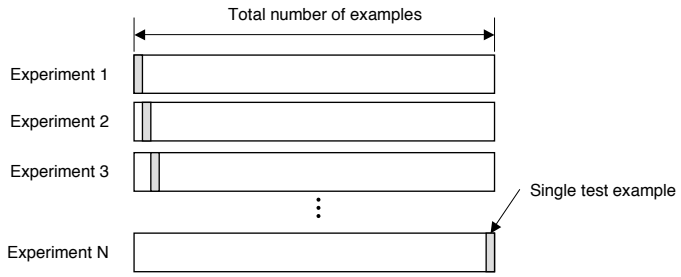

Figura 3: Leave-one-out cross validation

entrenamiento y la restante para la prueba, tal como se muestra en la Figura Esta validación cruzada indica separar los datos de tal forma que se tenga una sola muestra para el subconjunto de de prueba y el resto de datos para el subconjunto de entrenamiento, tal como se muestra en la Figura 3.

El error real puede ser estimado como el error promedio obtenido en los ejemplos de prueba:

$$E = \frac{1}{n}\sum_{i=1}^{n} E_i$$

3. Bootstrap, bootstrap .632 y bootstrap .632+

De forma general, la familia de métodos *bootstrap* son técnicas de remuestreo en las que de un dataset conformado por n muestras se eligen n, con la característica distintiva de que la selección es con reemplazo, permitiendo que cada instancia aparezca múltiples veces en el conjunto originado. Esta familia de técnicas es sumamente útil cuando el conjunto de datos es pequeño, permitiendo generar conjuntos sintéticos a partir del dataset original para las tareas de entrenamiento y clasificación.

Nuevamente, de forma general, para realizar la estimación del error se procede a realizar la selección de N subconjuntos bootstrap y calcular la media de los resultados de la clasificación. En la práctica, la repetición de estas operaciones funciona dado que, si las muestras son elegidas de forma aleatoria, mantendrán las mismas características de la población en la que fueron originadas.

3.1. Cálculo de la estimación del error en la familia de bootstrap

Si el dataset de entrenamiento es $\mathbf{X} = (x_1, \ldots, x_N)$ entonces es posible tomas $\mathcal{B}$ muestras bootstrap con reemplazo denominadas $\mathbf{Z_1}, \ldots, \mathbf{Z_B}$ donde cada $\mathbf{Z_i}$ contiene n instancias. Entonces es posible utilizar al conjunto de muestras $\mathbf{Z}$

para estimar el error real en la predicción del clasificador como:

$$\mathrm{Err}_{\mathrm{boot}} = \frac{1}{B}\sum_{b=1}^{B}\frac{1}{N}\sum_{i=1}^{N}\mathcal{L}(y_i, f^b(x_i))$$

donde $f^b(x_i)$ es el valor predecido para x_i según el modelo generado para el i-ésimo dataset bootstrap y $\mathcal{L}$ es una función para medir el error del clasificador.

Sin embargo, este no es un buen estimador del error porque las muestras utilizadas para el entrenamiento del clasificador pudieron haber contenido a x_i. El estimador leave-one-out bootstrap ofrece una mejora que intenta imitar al enfoque the cross-validation y se define como:

$$\mathrm{Err}_{\mathrm{boot}(1)} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|C^{-1}|}\sum_{b\in C^{-i}}\mathcal{L}(y_i, f^b(x_i))$$

donde C^{-i} es el conjunto de índices para los que las conjuntos bootstrap no incluyen la observación i y $|C^{-i}|$ es la cantidad de dichos conjuntos.

$\mathrm{Err}_{\mathrm{boot}(1)}$ soluciona el problema de sobreajuste, pero continua manteniendo un bias originado por las observaciones no distintas originadas por el muestreo con reemplazamiento. Se ha calculado, que el promedio de instancias distintas en cada conjunto bootstrap es aproximadamente de $0{,}632N$. Efron y Tibshirani propusieron el estimador 0,632 definido como:

$$\mathrm{Err}_{,632} = 0{,}368\overline{\mathrm{err}} + 0{,}632\mathrm{Err}_{\mathrm{boot}(1)} \tag{1}$$

donde

$$\overline{\mathrm{err}} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}(y_i, f^b(x_i))$$

La formulación de la ecuación 1 puede ser generalizada como:

$$\mathrm{Err}_{,632} = w\overline{\mathrm{err}} + (1-w)\mathrm{Err}_{\mathrm{boot}(1)} \tag{2}$$

donde $w = 0{,}632$.

Bajo algunos escenarios, el método bootstrap .632 puede seguir describiendo un sobreajuste grande, que puede ser medido por $\mathrm{Err}_{\mathrm{boot}(1)} - \overline{\mathrm{err}}$. Típicamente, esto sucede cuando el error de resustitución ($\mathrm{Err}_{\mathrm{boot}(1)}$) es 0. Para solucionar esto, el estimador *bootstrap .632+* pone un peso w mayor en $\mathrm{Err}_{\mathrm{boot}(1)}$, calculado a partir de *la tasa de traslape relativo* $\hat{R}$ mediante

$$\hat{R} = \frac{\overline{\mathrm{err}} - \mathrm{Err}_{\mathrm{boot}(1)}}{\hat{\gamma} - \mathrm{Err}_{\mathrm{boot}(1)}}$$

donde $\hat{\gamma}$ es la tasa de error de no-información, estimada mediante la permutación de las respuestas y_i y los predictores (patrones) t_j. Para un problema de clasificación dicotómico dicha tasa puede ser calculada de la siguiente manera.

Sea $\hat{p}_1$ la proporción de respuestas $y_i = 1$ y $\hat{q}_1$ la proporción de predicciones observadas $r_x(t_j) = 1$, entonces:

$$\hat{\gamma} = \hat{p}_1(1 - \hat{q}_1) + (1 - \hat{p}_1)\hat{q}_1 \tag{3}$$

Finalmente, la estimación del error bootstrap .632+ es definida como:

$$\text{Err}_{,632+} = w\overline{\text{err}} + (1 - w)\text{Err}_{\text{boot}(1)},\ w = \frac{,632}{1 - ,368\hat{R}} \tag{4}$$

El estimador bootstrap .632+ es sumamente confiable cuando $n > p$ (la cantidad de instancias es mayor a la cantidad de atributos o dimensiones).

4. División de datos en tres conjuntos

Normalmente, si la selección del modelo y las estimaciones del error real van a ser calculadas simultáneamente, los datos necesitan ser divididos en tres conjuntos disjuntos.

- **Entrenamiento**: Un conjunto de ejemplos utilizados para aprendizaje, en específico para ajustar los parámetros del clasificador.
- **Validación**: Un conjunto de instancias utilizadas para adaptar los parámetros del clasificador.
- **Prueba**: Un conjunto de ejemplos utilizados únicamente para evaluar el desempeño de un clasificador compltamente entrenado. Después de esta evaluación, el modelo ya no debe ser modificado.

El procedimiento para su ejecución es mostrado en el Algoritmo 1. Dicho algoritmo asume un métdo holdout, en caso de utilizar validación cruzada o bootstrap, los pasos 3 y 4 deben ser repetidos por cada una de las k particiones.

Algoritmo 1 Algoritmo para realizar entrenamiento, validación y prueba de un clasificador

1: Dividir el conjunto de datos disponible en subconjuntos de entrenamiento, validación y prueba.
2: Seleccionar una arquitectura de clasificador y parámetros de entrenamiento.
3: Entrenar el modelo a través del conjunto de entrenamiento.
4: Evaluar el modelo utilizando el conjunto de validación.
5: Repetir los pasos 2 al 4 utilizando distintas arquitecturas y parámetros de entrenamiento.
6: Seleccionar el mejor modelo y entrenarlo utilizando los datos de los conjuntos de entrenamiento y validación.
7: Evaluar el desempeño de este modelo final utilizando el conjunto de prueba.

Referencias

[1] Bradley Efron and Robert Tibshirani. Improvements on cross-validation: the 632+ bootstrap method. *Journal of the American Statistical Association*, 92(438):548–560, 1997.

[2] Wilfrido Gómez. El método bootstrap .632+. Technical report, Cinvestav Tamaulipas, 2015.

[3] Ricardo Gutierrez-Osuna. Validation. Technical report, Wright State University, 2002.

[4] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'95, pages 1137–1143, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[5] StackExchange. What is the .632+ rule in bootstrapping?, 2014.