Departamento de Computación, FCEyN, UBA

Procesamiento del Habla

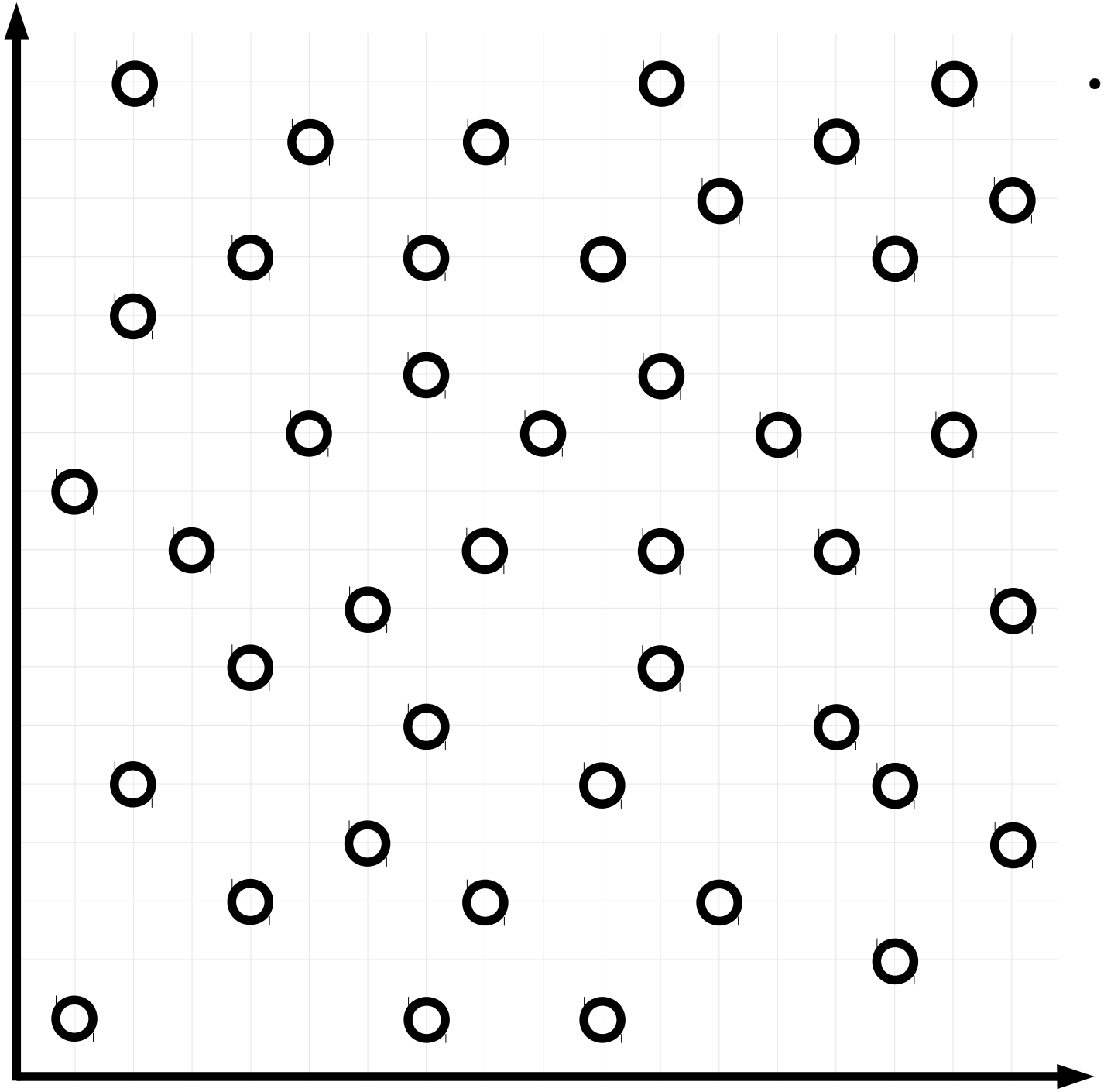
Agustín Gravano

1er Cuatrimestre 2017

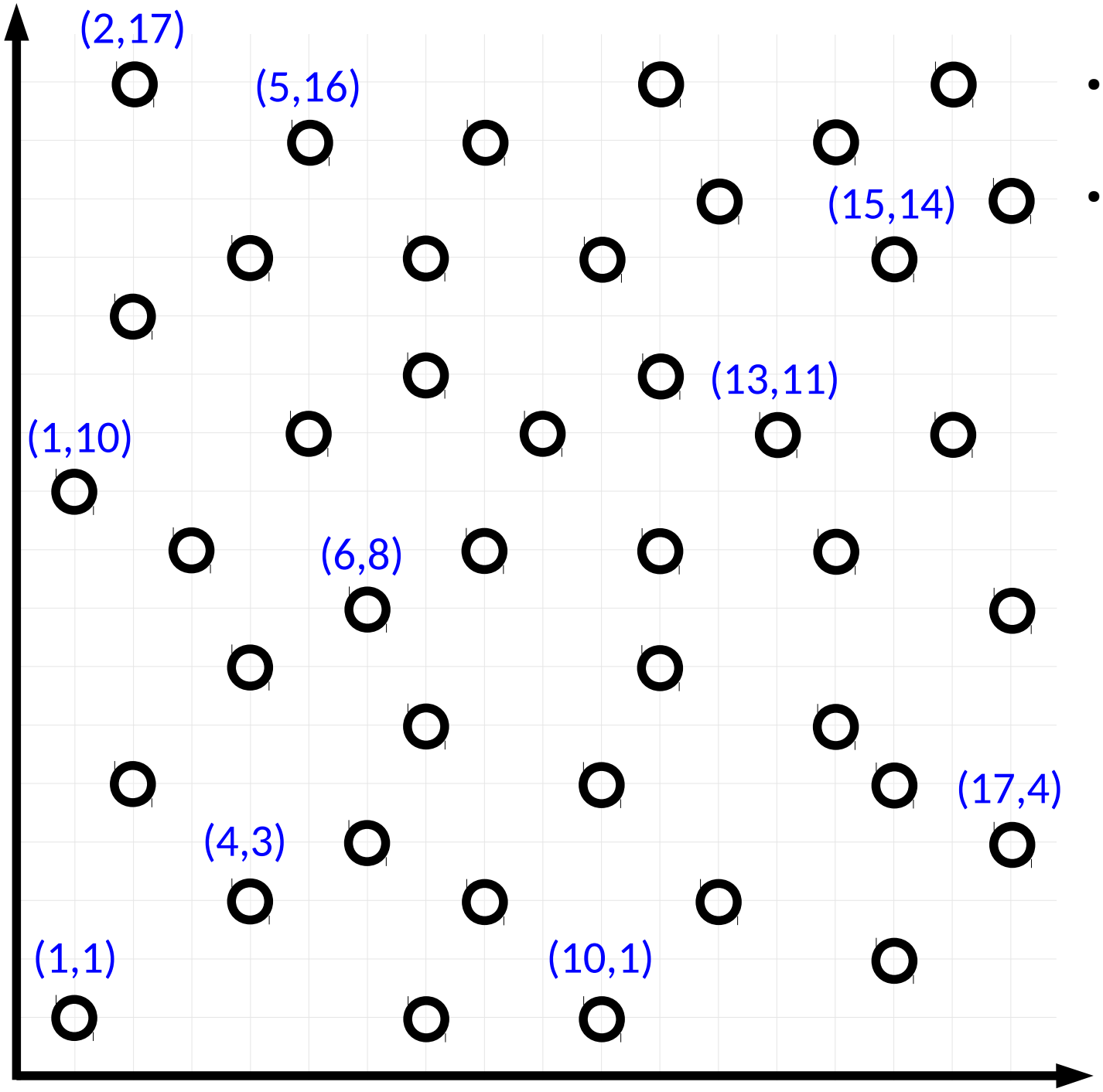
Procesamiento del Habla

- Acústica, fonética y fonología
- Síntesis del habla - TTS
- Sistemas de diálogo hablado
- Aprendizaje automático en habla
- Reconocimiento del habla – ASR
- Temas avanzados

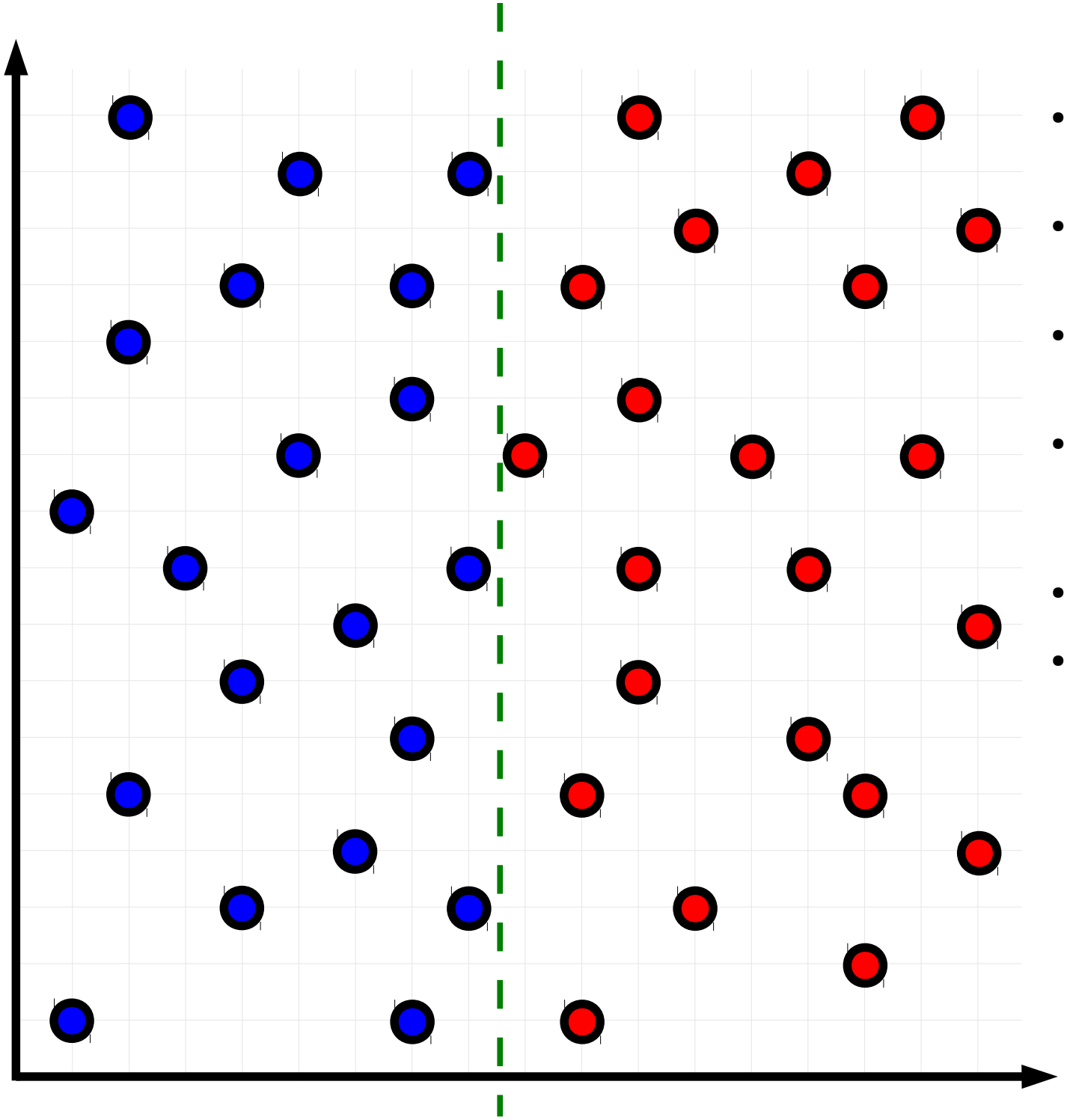
Breve introducción al Aprendizaje Automático (Machine Learning)



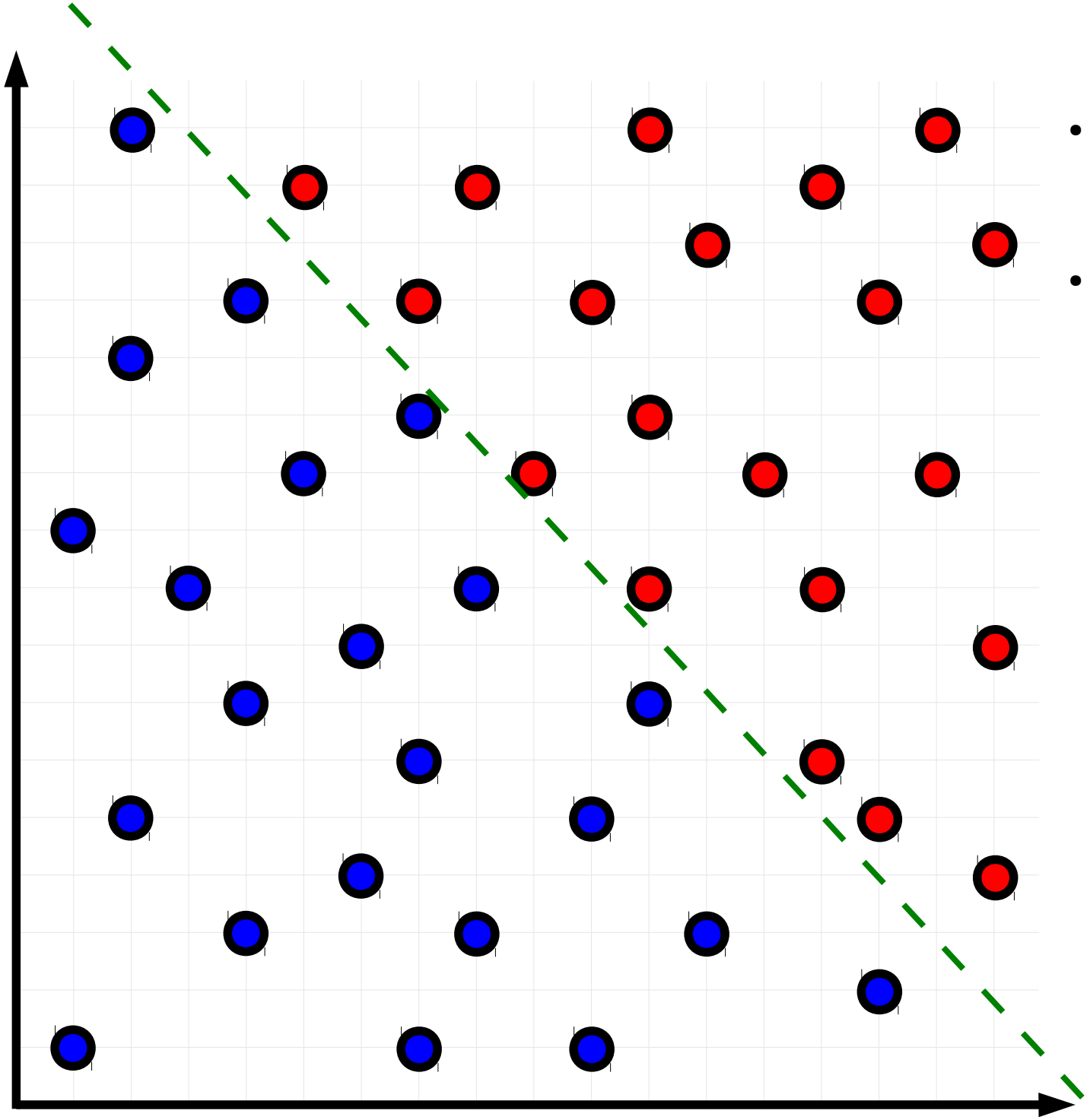
- Tenemos *N* puntos en el plano.



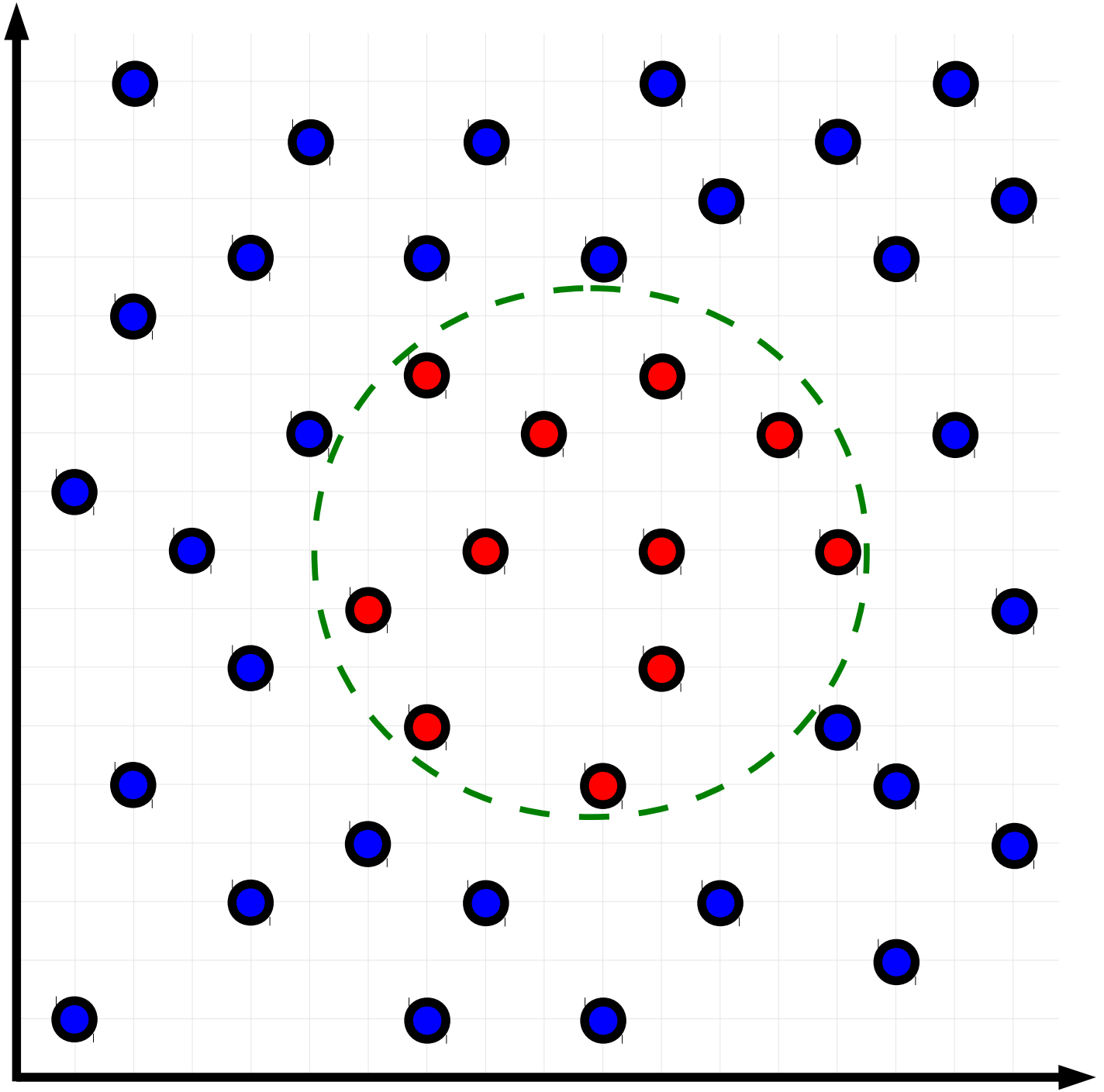
- Tenemos *N* puntos en el plano.
- Cada punto tiene dos coordenadas: (*x*, *y*).

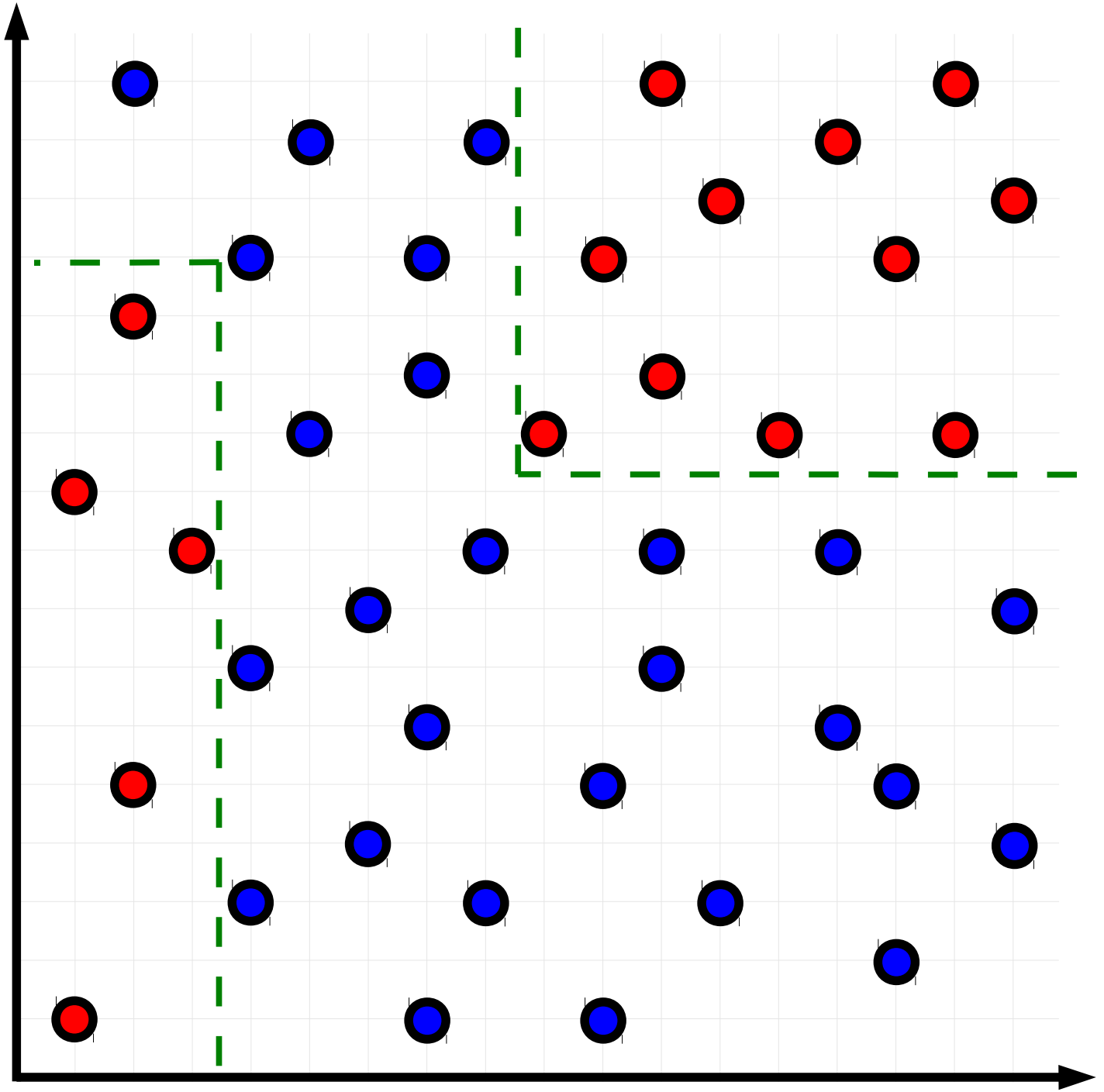


- Tenemos *N* puntos en el plano.
- Cada punto tiene dos coordenadas: (*x*, *y*).
- Cada punto tiene un color: **azul** o **rojo**.
- Queremos hallar una forma de **predecir** el color de puntos **nuevos**.
- Función $f(x, y) \rightarrow$ color
- $f(x, y)$ =
  rojo si $x > 8$
  azul en caso contrario



- $f(x, y)$ =
  rojo si $y > m\,x + b$
  azul en caso contrario
- ***m*** y ***b*** son constantes del modelo que deben ajustarse a los datos.





Humanos vs. Máquinas

- Los humanos somos buenos encontrando (y programando) estas reglas en 2D.
- Pero, ¿qué pasa si los puntos tienen miles de coordenadas?
- Ejemplo: Detección de caras.

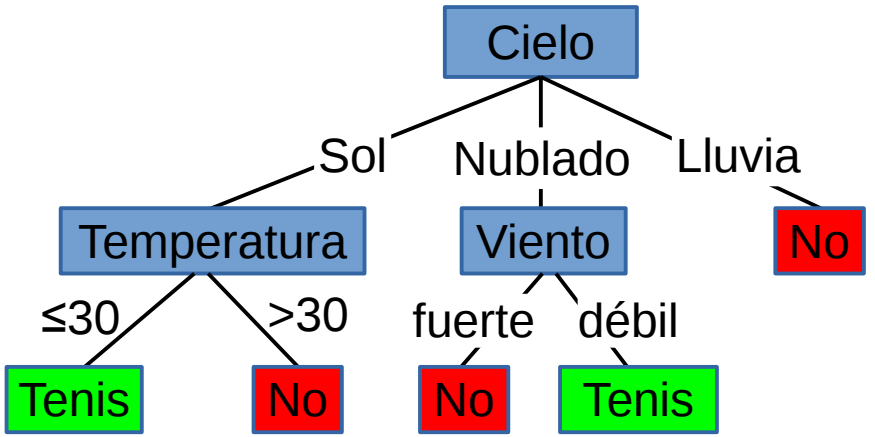
  - Los humanos somos muy buenos detectando (y reconociendo) caras.
  - Pero ¿podemos **programar estas funciones**?

Terminología de ML

- Puntos → Instancias.
- *x*, *y* → Atributos de las instancias.
- Color → Clase de las instancias.
- Encontrar una función → Entrenar un modelo.

Algoritmos de ML

- Árboles de decisión (C4.5)



- Reglas (Ripper)

  IF (Cielo=Sol ∧ Temperatura>30) THEN Tenis=No

  IF (Cielo=Nublado ∧ Viento=Débil) THEN Tenis=Sí

  IF (Cielo=Lluvia) THEN Tenis=No

- Naive Bayes
- Support Vector Machines
- …

Herramientas de ML

- Python scikit learn

  http://scikit-learn.org

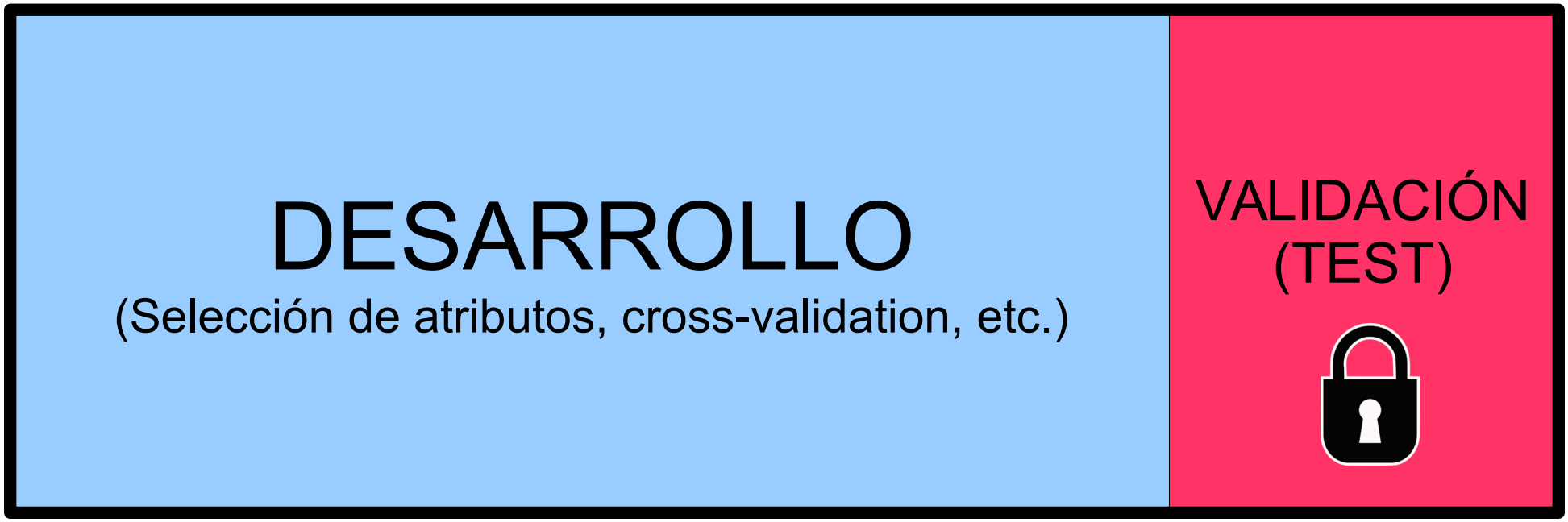
- Weka (algo vieja ya)

  http://www.cs.waikato.ac.nz/ml/weka/

Esquema General de ML

- Datos:
  - Separar datos de desarrollo y validación.
  - Definir instancias, clases y atributos.
- Experimentación:
  - Selección de atributos.
  - Medidas de performance.
  - Validación cruzada de *k* iteraciones.
- Validación de los modelos.

Datos de Desarrollo y Validación



- Lo antes posible, hay que separar un conjunto de datos de validación.
- Todas las pruebas y ajustes se hacen sobre el conjunto de desarrollo.
- Cuando termina el desarrollo, se evalúa sobre los datos separados.

Instancias, clases y atributos

- Tenemos que definir:
  - cuáles son las instancias de nuestra tarea;
  - cuáles son sus clases y sus atributos.
- En nuestro ejemplo:
  - Instancias: Strings reconocidos como siglas en una etapa anterior del front-end.
  - Clases: Deletrear vs. Leer como un acrónimo.
  - Atributos: longitud de la sigla; #consonantes; #vocales; #consonantes consecutivas; ...

Selección de Atributos

- ¿Demasiados atributos?
  - Aprendizaje muy lento (ej, para SVM).
  - Riesgo de sobreajuste (*overfitting*).
- Selección de atributos: usar sólo un subconjunto útil.
- Ejemplos:
  - Búsqueda exhaustiva (normalmente impracticable).
  - Ranking según su ganancia de información.
  - Greedy forward selection
    - $S \leftarrow \emptyset$ (S = conjunto de atributos)
    - Para cada atributo a_i no usado, evaluar $S \cup \{a_i\}$.
    - Si ningún $S \cup \{a_i\}$ produce una mejora, devolver S.
    - En caso contrario, $S \leftarrow S \cup \{a_i\}$ y volver a 2).
  - Greedy backward elimination

Experimentación con Clasificadores

- Ya elegimos los mejores atributos.
- Siguiente: elegir un clasificador para entrenar un modelo.
- Para ello, experimentamos con diferentes clasificadores y configuraciones, siempre sobre los datos de desarrollo:
  - Árboles de decisión
  - Reglas
  - Naive Bayes
  - Support Vector Machines
  - …
- Una vez elegido el mejor clasificador, entrenamos el modelo usando todos los datos de desarrollo.

Medidas de Performance

- Ejemplo: Detección de spam. Clases: {*mail*, *spam*}
- Matriz de confusión
  - 100 datos de test: 50 siglas y 50 acrónimos.

		Valor real: *mail*	Valor real: *spam*
Valor predicho	*mail*	40 (tn)	5 (fn)
	spam	10 (fp)	45 (tp)

tp = *true positive*
fp = *false positive*
tn = *true negative*
fn = *false negative*

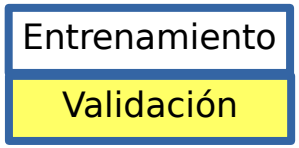
- Accuracy (% de aciertos) = (tp + tn) / total = 0.85
- Precisión = tp / (tp + fp) = 45 / (45 + 10) = 0.82
- Recall = tp / (tp + fn) = 45 / (45 + 5) = 0.90
- *F*-measure = (2 · precision · recall) / (precision + recall) = 0.86

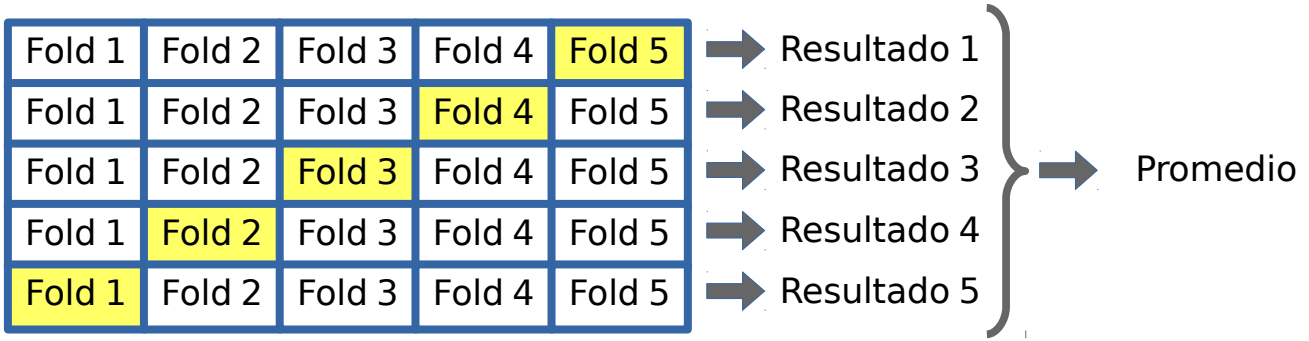
Validación Cruzada de *k* iteraciones
(*k-fold cross validation*)

- ¿Qué puede pasar si tenemos mala suerte al separar los datos para entrenamiento/validación?
- *k*-Fold Cross Validation:
  1) Desordenar los datos.
  2) Separar en *k* folds del mismo tamaño.
  3) Para *i* = 1..*k*:

     Entrenar sobre todos los folds menos el *i*.

     Evaluar sobre el fold *i*.
- Ej. para *k*=5:





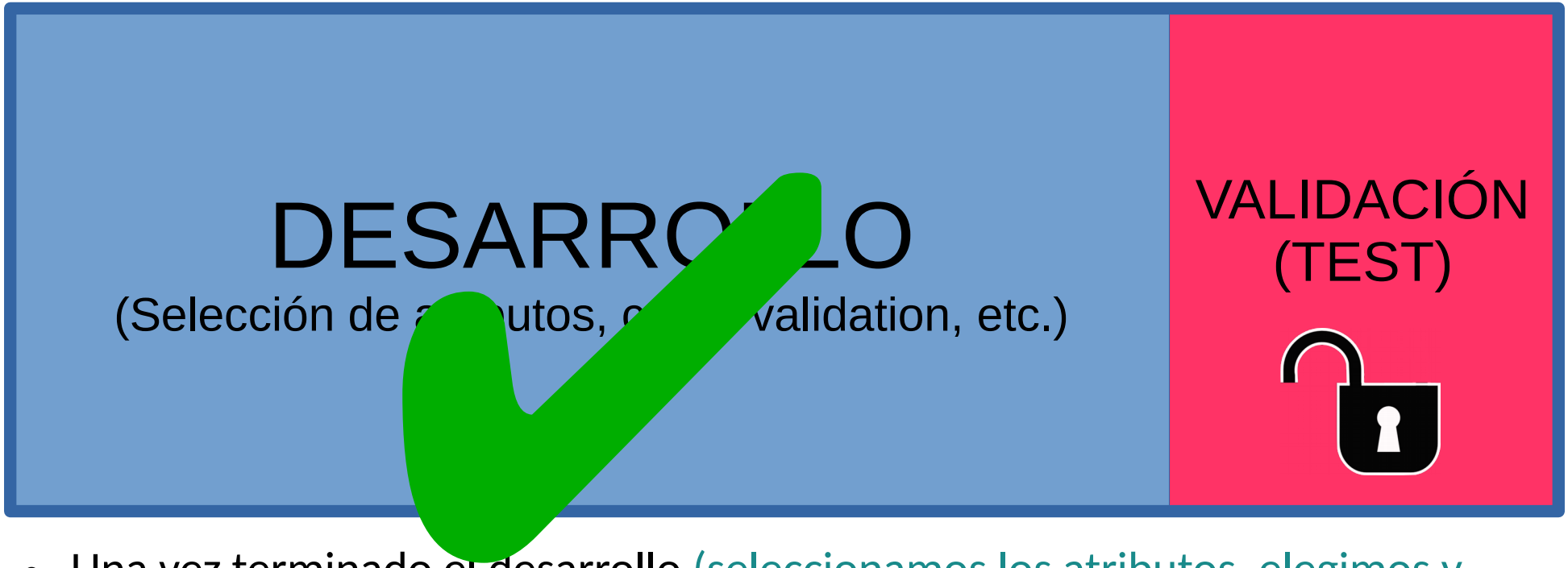
Aplicaciones en PLN y Habla

- Segmentación en oraciones.
- Etiquetado de clases de palabra (*POS tagging*).
- Desambiguación del sentido (ej: abreviaturas, expresiones numéricas).
- Asignación de prosodia (TTS front-end).
- Detección del idioma (p.ej., para saber cómo sintetizar).
- **Pronunciación de siglas.**
- …

Ejemplo: Pronunciación de Siglas

- Tarea: Determinar cómo pronunciar siglas en español.
  - **Input:** sigla.
    - Ejemplos: DGI, IBM, FBI, FMI, ATP, UBA, ALUAR, CONADUH, CONADEP, APTRA, AFA, FIFA.
  - **Output:** decidir si debe deletrearse, o leerse como un acrónimo.
- Este clasificador puede ser útil cuando encontramos una sigla desconocida en el texto a sintetizar.
- Por simplicidad, excluimos siglas con pronunciación especial: MIT (*emaití*), CNEA (*conea*), FCEN (*efecen*).
- siglas.csv, clasificador-siglas.ipynb

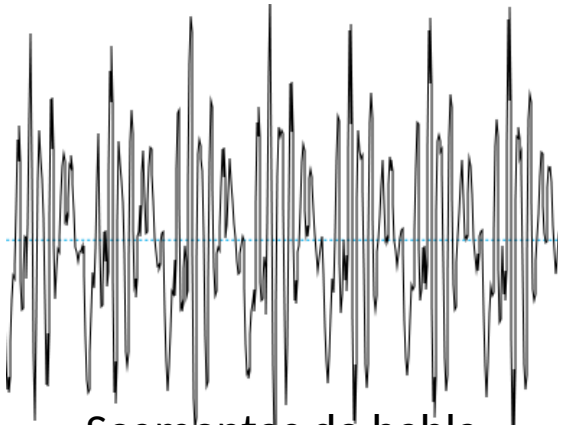
Validación del Modelo Final



- Una vez terminado el desarrollo (seleccionamos los atributos, elegimos y configuramos el clasificador con mejores resultados con validación cruzada sobre los datos de desarrollo, y entrenamos el clasificador usando todos los datos de desarrollo), podemos **evaluar** el modelo final sobre **los datos de validación**.
- Esto nos da una **estimación realista** de la performance del modelo.
- Una vez usados los datos de validación, **no se debe volver atrás**.

Aprendizaje Automático en Habla

Aprendizaje Automático y el Procesamiento del Habla



Segmentos de habla

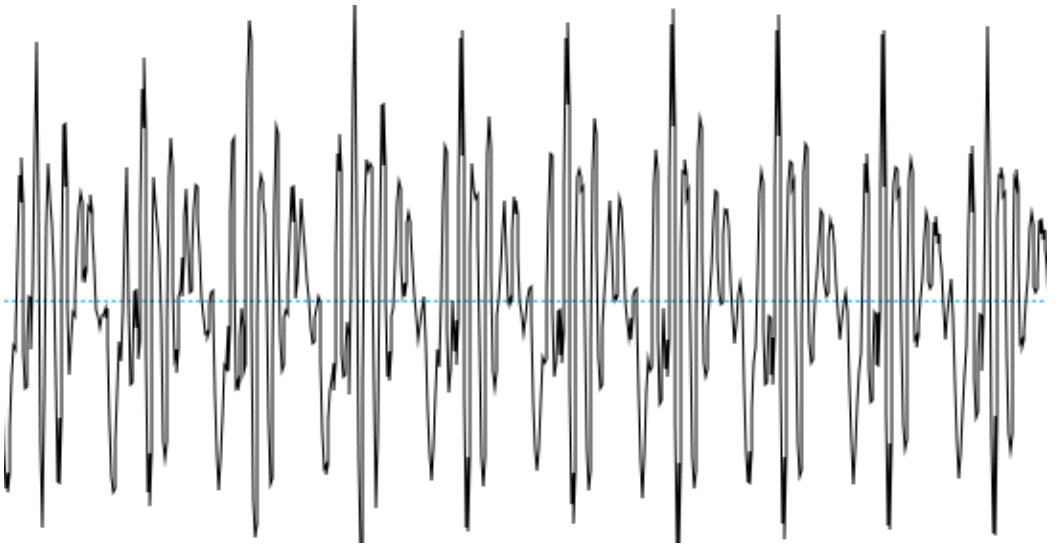
Atributo #1
Atributo #2
Atributo #3
Atributo #4
Atributo #5
Atributo #6
...

Modelo de Aprendizaje Automático

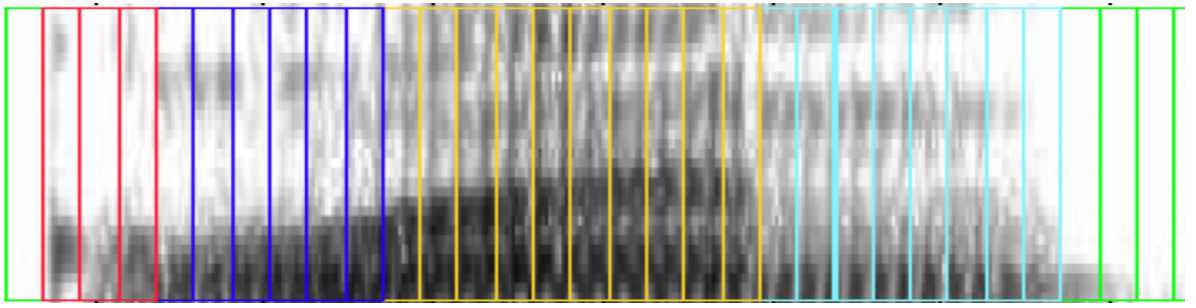
Info

- Palabras (reconocimiento del habla).
- Identidad, género, edad del hablante.
- Lenguaje, dialecto, variantes regionales.
- Emociones, intoxicación, cansancio, atención.
- Estructura del discurso, niveles de prominencia.

¿De qué segmentos de habla extraemos atributos?

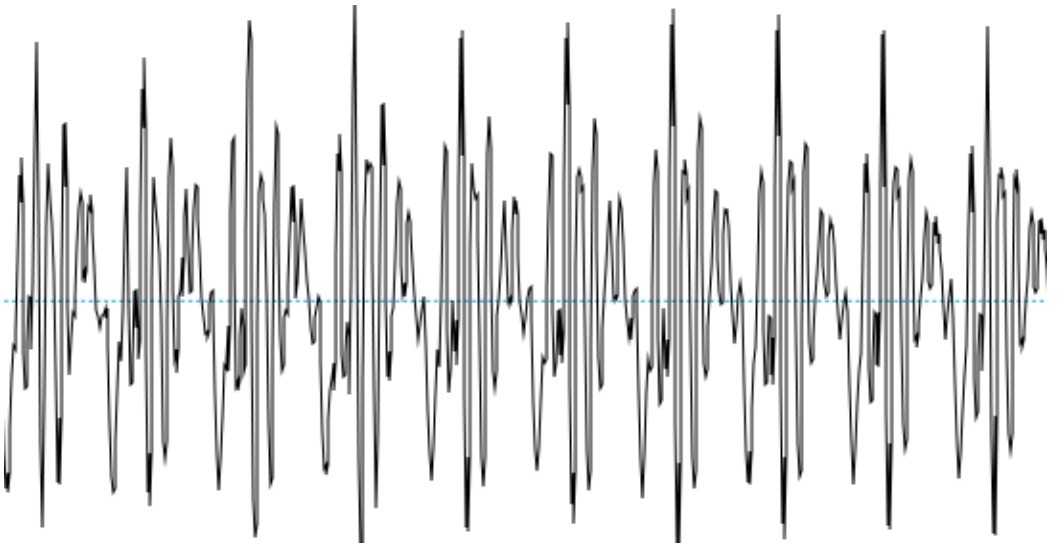


- La longitud del segmento depende de la tarea.
  - Desde pocos milisegundos (ej: ASR ~20ms).



  - Hasta múltiples frases (ej: detección de emociones, identificación del hablante).

¿Qué atributos podemos extraer de un segmento de habla?



- Nivel tonal (pitch)
- Intensidad (RMS)
- Jitter
- Shimmer
- Relación ruido-armónico (NHR)

×

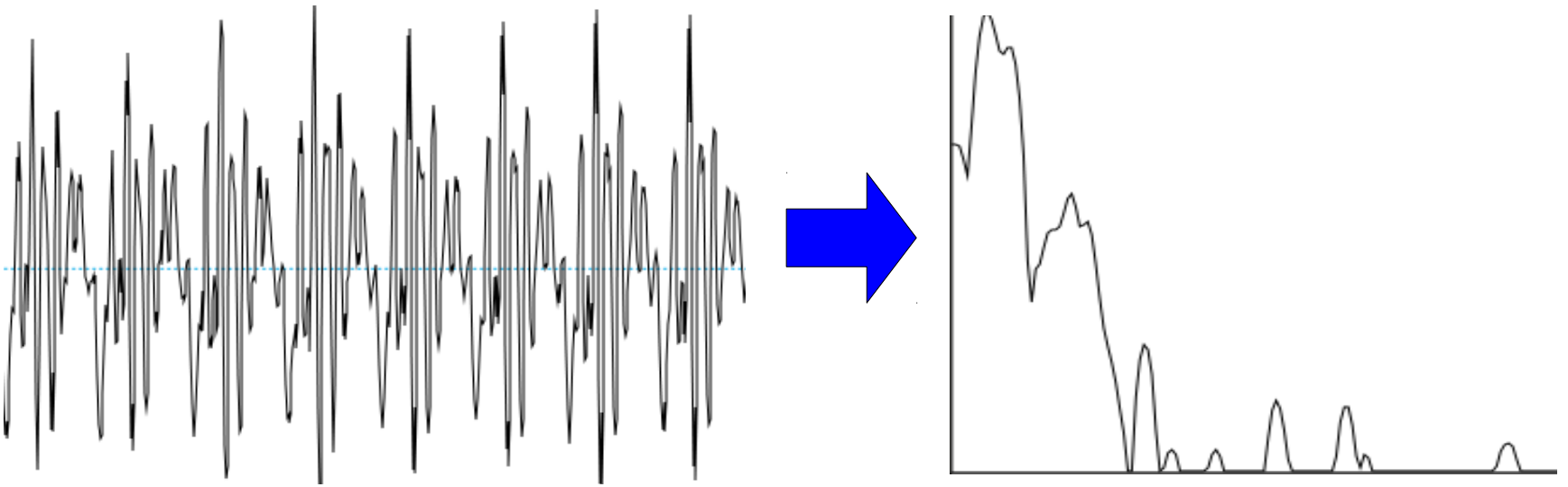
- Máximo, mínimo
- Media, mediana
- Percentiles
- Desviación estándar

×

- Suavizado
- Derivadas

MFCC

- Atributos espectrales muy usados en muchas tareas.
- MFCC (*Mel frequency cepstral coefficient*):
  1) Aplicar FFT a la señal. → Espectro

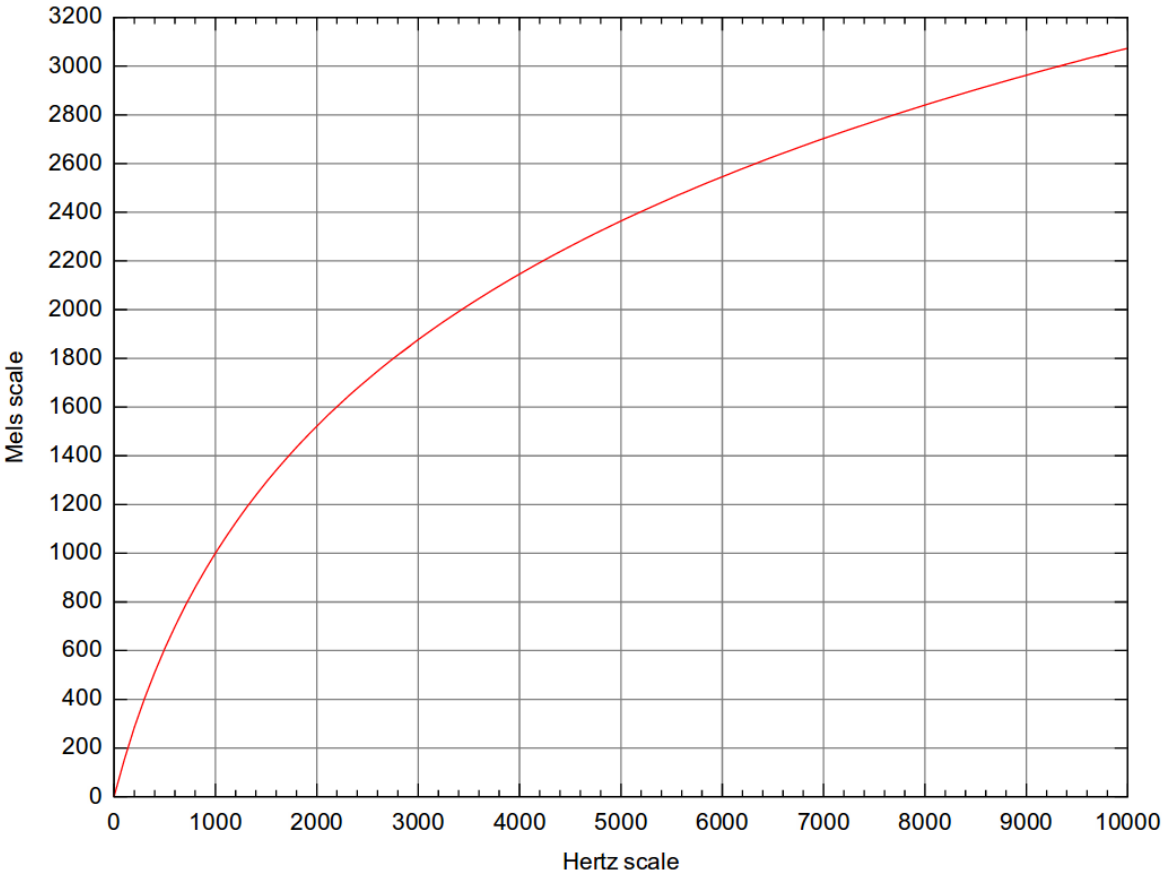


MFCC

- Atributos espectrales muy usados en muchas tareas.
- MFCC (*Mel frequency cepstral coefficient*):
  1) Aplicar FFT a la señal. → Espectro
  2) Mapear las amplitudes del espectro a la escala mel.

Escala Mel

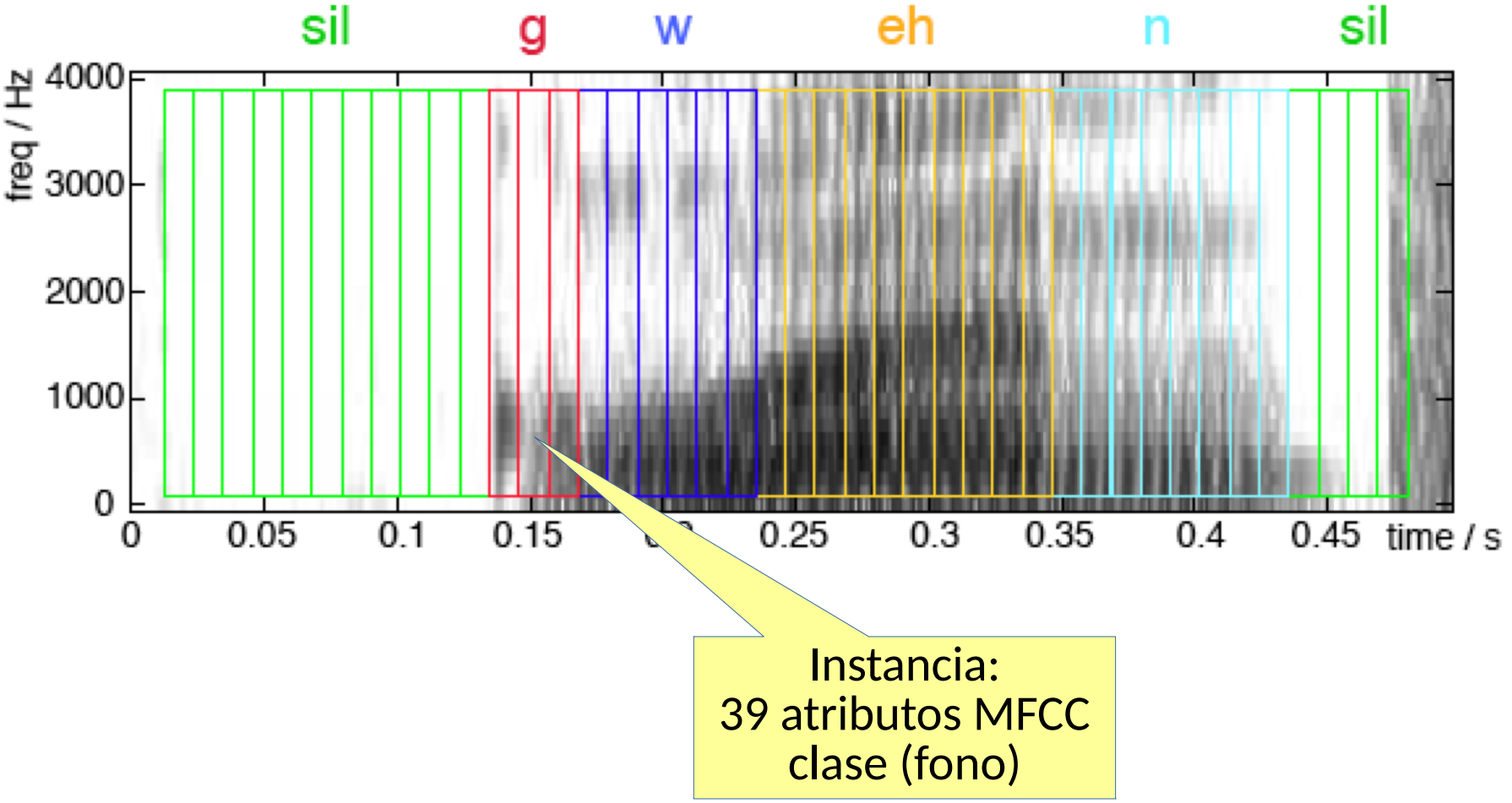
- Escala tonal basada en la percepción humana.
- $mel = 1127{,}01048 \ln(1 + f / 700)$



MFCC

- Atributos espectrales muy usados en muchas tareas.
- MFCC (*Mel frequency cepstral coefficient*):
  1) Aplicar FFT a la señal. → Espectro
  2) Mapear las amplitudes del espectro a la escala mel.
  3) Calcular el logaritmo de las amplitudes del espectro.
  4) Aplicar la transformada discreta del coseno (DCT) de la lista de valores resultantes (como si fuera una señal).
  5) Los MFCC son las amplitudes del espectro resultante.

Reconocimiento del Habla



Con los MFCC vamos a poder clasificar frames (~20ms) de la señal en fonos, según sus características espectrales, usando GMM o DNN. (Clase que viene)

Herramientas
(todas código abierto, GNU-GPL, multi-plataforma)

- Praat (praat.org)
- Opensmile (audeering.com/technology/opensmile/)
  - En los labos, instalado en /home/ph-30/opensmile, se puede usar con:

    $DIR/SMILExtract -C $DIR/config/IS10_paraling.conf -I in.wav -O out.arff

    donde $DIR=/home/ph-30/opensmile

    IS10_paraling.conf es una configuración con 1582 atributos usada en el *INTERSPEECH 2010 Paralinguistic Challenge* (+info: *openSMILE book*, p32).
  - Biblioteca en Python para parsear ARFF: liac-arff (pip install --user liac-arff)
- Aubio (aubio.org)
  - Biblioteca para C, Python. Versión actual: 0.4.5

Ejercicio: Detección del género del hablante

0) **Datos:** /home/ph-30/clase-05/datos/NNNgMMh.{wav,ipu}
   NNN=id hablante (000-092) g=f/m MM=edad h=r/s (habla leída o espontánea)
   wav=audio ipu=transcripción de unidades sin pausas
   184 audios, grabados por 92 personas (46m, 46f; edad: 20-74).
   (Descripción más detallada en la clase de Estadística.)

1) **Extraer atributos** de los audios y/o transcripciones.
   Para el audio, usar Praat, openSMILE, Aubio o cualquier otra herramienta.

2) **[opcional] Realizar selección de atributos.** Idealmente, #atributos debe ser un orden de magnitud inferior a #instancias (conservar ~40 atributos).

3) **Entrenar un modelo** de ML: árboles, SVM, Random Forests, etc. Evaluar la performance usando cross-validation. Quedarse con el mejor modelo.

4) **Escribir un programa** que, dado un wav nuevo, extraiga los atributos necesarios, invoque al modelo aprendido para realizar la clasificación, y devuelva “m” o “f”.

Otras tareas: - clasificar en habla leída vs. espontánea;
- predecir la edad del hablante (clasificación o regresión).

Repaso de ML y Habla

- Fundamentos de aprendizaje supervisado.
  - Clasificación, instancias, atributos, clases.
  - Extracción y selección de atributos.
  - Medidas de performance: accuracy, precisión, recall, F.
- Aplicaciones en habla.
  - ASR, id, género, edad, lenguaje, emociones, discurso, etc.
  - Atributos acústicos.
- Próxima clase:
  - Reconocimiento automático del habla.