ST790 — Fall 2022

Imprecise-Probabilistic Foundations of Statistics

Ryan Martin
North Carolina State University
`www4.stat.ncsu.edu/~rmartin`

Week 12b

This lecture

- Recap some of general ML details
- More classification (with imprecise probability)
- In particular:
  - Denoeux's *evidential neural network classifier*
  - conformal prediction and IMs
- .....

Quick recap of ML

- Ingredients:
  - data, e.g., features X_i and labels Y_i
  - class $\mathcal{F}$ of functions, hopefully $y \approx f^\star(x)$ for some $f^\star \in \mathcal{F}$
  - loss function, ℓ_f, to rate quality of f
- Note the absence of a statistical model...
- Training step boils down to "estimating" f via empirical risk minimization,[1] i.e., $\hat{f}_n = \arg\min_{f \in \mathcal{F}} n^{-1} \sum_{i=1}^n \ell_f(X_i, Y_i)$
- e.g., in classification,

$$\hat{f}_n(x) = \arg\max_{y \in \mathbb{Y}} \underbrace{\widehat{\mathrm{P}}(Y = y \mid X = x)}_{\text{estimated predictive prob}}$$

- Use the trained $\hat{f}_n$ to predict/classify new examples

[1] Stochastic gradient descent is commonly used

Quick recap, cont.

- Huge $\mathcal{F}$ and fancy algorithms/technology won't eliminate uncertainty, so UQ will always be relevant
- Two dominant statistical schools of thought?
  - *frequentist*
    - $\rightarrow$ estimation is relatively easy
    - $\rightarrow$ UQ isn't at all automatic
    - $\rightarrow$ if it can be done, then likely inefficient ("model agnostic")
  - *Bayesian*
    - $\rightarrow$ difficult to do (if one's being "honest")
    - $\rightarrow$ UQ is an immediate by-product
    - $\rightarrow$ meaningfulness of UQ wrt a single posterior dist?
- Imprecise-prob methods are a promising middle-ground...?

Classification

- **Last time:** *naive credal classifier*
- Extension/imprecise version of naive Bayes classifier
- Key features:
  - weaker prior assumptions (re: Manski)
  - able to classify examples to multiple labels
  - computationally tractable (thanks to IDM connection)

- **Today:** belief function/Dempster–Shafer approaches
  - *evidential neural net classifier*[2]
  - deep version, based on convolutional neural nets[3]

[2] Denoeux (*IEEE SMC* 2000)
[3] Tong, Xu, and Denoeux (*Neurocomputing* 2021)

Evidential classifier

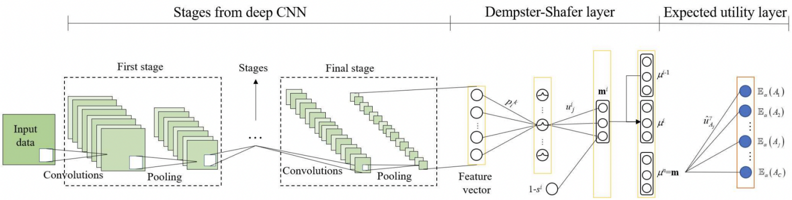
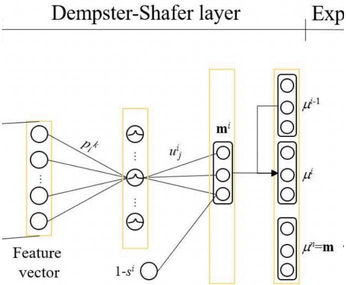


Figure 1: Architecture of an evidential deep-learning classifier.

- Multiple layers/stages:[4]
  - input gets processed through neural nets
  - neural net output gets converted into a mass/belief function
  - "expected utility" calculation for decision-making
- I'll focus exclusively here on the *DS layer*, which itself consists of several steps

[4]Screenshot from Tong, Xu, and Denoeux (2021)

Evidential classifier, cont.



- DS-layer consists of three steps:
    - distance-based support between input and references
    - mass function constructed for each reference
    - reference-specific mass functions combined via Dempster's rule
- Depends on parameters to-be-learned from training set

Details

- Data consists of (X, Y) pairs
  - Y's are labels
  - X's represent images, chunks of text, etc
- Processing: $X \rightsquigarrow Z = Z(X) \in \mathbb{R}^q$
  - "$\rightsquigarrow$" designed to extract important characteristics
  - depends on the form of the input
  - depends on lots of to-be-learned parameters
- For our purposes, it suffices to proceed as if (Y, Z) is the available data, ignoring the processing
- Focus on mapping Z to a belief/mass function for Y

Details, cont.

- Fix a set of *prototypes* $p^1, \ldots, p^R$ in $\mathbb{R}^q$
- Assign weight vectors α^r to each prototype:
  - $\beta_y^r :=$ p^r's degree of membership to class y
  - with constraint $\sum_y \beta_y^r = 1$ for each r
  - these are to-be-learned parameters
- For a generic $z \in \mathbb{R}^q$, calculate the distance to prototypes

$$d^r = d^r(z) = \|z - p^r\|, \quad r = 1, \ldots, R$$

- Factors influencing association between input z and label y
  - distance of z from prototypes
  - prototype membership degree with label y

Details, cont.

- Given z, for each prototype $r = 1, \ldots, R$, define a random set with mass function $m^r(\cdot)$,

$$m^r(\{y\}) = \alpha^r \beta^r_y \exp\{-\gamma^r (d^r)^2\}, \quad y \in \mathbb{Y}$$
$$m^r(\mathbb{Y}) = 1 - \alpha^r \exp\{-\gamma^r (d^r)^2\}$$

- α^r's, β^r's, and γ^r's are to-be-learned parameters
- Easy to check that this is a genuine mass function

$$\sum_y m^r(\{y\}) + m^r(\mathbb{Y}) = 1$$

- Defines a belief/plausibility function on $\mathbb{Y}$
- This gives a prototype-specific quantification of uncertainty about which label y is associated with input z

Details, cont.

- Goal is overall UQ, not a prototype-specific UQ
- Denoeux's idea:
  - since each prototype-specific UQ is a belief function
  - just combine $m^1, \ldots, m^R$ via Dempster's rule
- In symbols, $m = \bigoplus_{r=1}^R m^r$, Shafer's orthogonal sum
- Detailed formulas are messy[5] and, hence, omitted
- Given this (z-dependent) mass function m, there are some options for carrying out classification:
  - naive strategy, $\arg\max_y m(\{y\})$
  - belief function yields a Choquet integral, so we can classify based on optimizing lower/upper expected utility[6]

[5] Denoeux uses some recursive relations...

[6] I'll cover general decision-theory details later

Evidential classifier: summary

- Process raw X through, say, a convolutional neural net
- Output Z and labels Y go into the DS-layer
- Returns a belief function on $\mathbb{Y}$ for classification
- Parameters to be tuned in both the initial processing and the DS-layer, can be handled simultaneously via SGD
- For a new example, the feature X_{n+1} gets mapped to Z_{n+1} and then to a belief function on $\mathbb{Y}$
- Classification rule can be tailored so that set-valued classifications are made, more conservative, less error-prone

Conformal prediction (again)

- Roughly, Denoeux takes some existing machinery and uses the output to construct a belief function for UQ
- There are other ways to implement such a strategy
- *Conformal prediction*[7] is a powerful method to leverage
- Recall:
  - set $Z_i = (X_i, Y_i)$ for $i = 1, \ldots, n$
  - set $Z_{n+1} = (x, y)$ for generic (x, y)
  - define a non-conformity score $M(B, z)$
  - compute $\mu_i = M(\{Z_1, \ldots, Z_{n+1}\} \setminus \{Z_i\}, Z_i)$, $i = 1, \ldots, n+1$
  - return $\pi_n(y \mid x) = (n+1)^{-1} \sum_{i=1}^{n+1} 1\{\mu_i \geq \mu_{n+1}\}$
  - prediction region: $C_\alpha(Z^n; x) = \{y : \pi_n(y \mid x) > \alpha\}$

[7] Vovk et al's *Algorithmic Learning in a Random World*

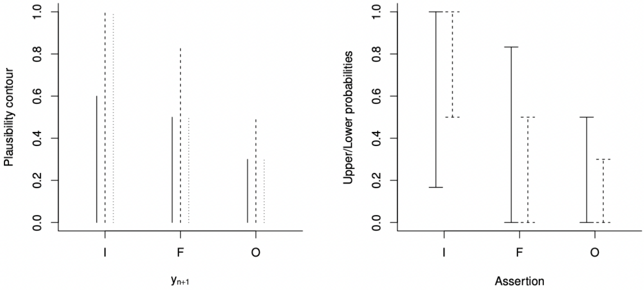
Conformal prediction, cont.

- It turns out that conformal prediction can be related to (nested) random sets and belief functions[8]
- Conformal prediction's coverage reliability aligns with IM validity, so it's a special kind of belief function
- With finite $\mathbb{Y}$, the random set can be empty with non-zero probability; implies $\pi_n(y \mid x) < 1$ for all y
- This is bad — coherence & validity fail
- Two remedies:
  - condition on random set $\neq \varnothing$ (Dempster-style)
  - appropriately "stretch" random set[9]
- Both preserve validity, but latter is more efficient!

[8] Cella & M. (*IJAR* 2022), `arXiv:2112.10234`

[9] M. and Liu, *Inferential Models*, Ch. 5

Conformal prediction, cont.



(a) (b)

Figure 3: Panel (a): Plausibility contours in Equation (23), derived from an IM construction with no adjustment (solid lines), conditioning adjustment (dashed lines) and stretching adjustment (dotted lines). Panel (b): Upper and lower probabilities for the singleton assertions $\{I\}$, $\{F\}$ and $\{O\}$ derived from an IM construction with the conditioning adjustment (solid lines) and the stretching adjustment (dashed lines). These predictions are based on a new alligator of length $x_{n+1} = 2$ meters.

Conclusion

- I gave a high-level explanation of two imprecise-probability-based classification methods
  - evidential classifier: neural nets & Dempster–Shafer
  - IM classifier: conformal prediction & nested random sets
- Comparison:
  - conformal prediction can be used in conjunction with deep learning, but it's likely expensive[10]
  - evidential classifier (probably) doesn't have error rate controls
- Other methods...?

[10] "Split" conformal prediction is faster, but validity is only approximate

Next lecture

- Prediction in regression
- i.e., supervised learning with continuous Y
- More IMs and conformal prediction
- Brand new stuff on *random fuzzy numbers*