Lecture 4: So How Can We Design Explainable Fuzzy AI: Ideas

1 Machine Learning Revisited

Formulation of the problem. We want to find the dependence $y = f(x)$ of a quantity y on quantities $x = (x_1, \ldots, x_n)$. In each of several situations $k = 1, \ldots, K$, we know both:

- the value $x^{(k)}$ and
- the value $y^{(k)}$.

Based on this information, we want to find a function $f(x)$ for which, for all k, we have the value $f\left(x^{(k)}\right)$ is close to $y^{(k)}$: $f\left(x^{(k)}\right) \approx y^{(k)}$.

Details. Usually, we select a family of functions $f(x, c)$ characterized by some parameters $c = (c_1, \ldots, c_m)$. We want to find the values c_i of these parameters for which $f\left(x^{(k)}, c\right) \approx y^{(k)}$ for all k.

Least squares idea. We want to make sure that:

- the value $f\left(x^{(1)}, c\right)$ is close to $y^{(1)}$;
- the value $f\left(x^{(2)}, c\right)$ is close to $y^{(2)}$;
- $\ldots$, and
- the value $f\left(x^{(K)}, c\right)$ is close to $y^{(K)}$.

In other words, we want to make sure that the tuple

$$\ell = \left(f\left(x^{(1)}, c\right), f\left(x^{(2)}, c\right), \ldots, f\left(x^{(K)}, c\right)\right)$$

formed by the left-hand sides is close to the tuple

$$r = \left(y^{(1)}, y^{(2)}, \ldots, y^{(K)}\right)$$

formed by the right-hand sides.

It is reasonable to select the values of the parameters c for which the distance $D(\ell, r)$ between these two tuples is the smallest, i.e., equivalently, for which the square of this distance is the smallest:

$$D^2(\ell, r) =$$

$$\left(f\left(x^{(1)}, c\right) - y^{(1)}\right)^2 + \left(f\left(x^{(2)}, c\right) - y^{(2)}\right)^2 + \ldots + \left(f\left(x^{(K)}, c\right) - y^{(K)}\right)^2.$$

This formulation is known as the *Least Squares approach.*

Comment. As mentioned in the previous lecture, the least squares approach guarantee smallness on average. If we want all individual differences to be small, we need to use other data processing techniques.

How do we find the minimum: case of linear dependence. When the dependence on the coefficients c_i is linear, we can do what we did earlier:

- differentiate with respect to each c_i, and
- equate the resulting derivatives to 0.

Linear dependence: case of $n = 1$. Suppose that we are looking for a linear dependence $y = c_1 \cdot x + c_2$. In this case, the expression that we want to minimize takes the form

$$\left(c_1 \cdot x^{(1)} + c_2 - y^{(1)}\right)^2 + \ldots + \left(c_1 \cdot x^{(K)} + c_2 - y^{(K)}\right)^2.$$

Differentiating this expression with respect to c_2 and equating the derivative to 0, we get

$$2 \cdot \left(c_1 \cdot x^{(1)} + c_2 - y^{(1)}\right) + \ldots + 2 \cdot \left(c_1 \cdot x^{(K)} + c_2 - y^{(K)}\right) = 0.$$

Dividing both sides of this equality by 2 and opening the parentheses, we conclude that

$$c_1 \cdot x^{(1)} + c_2 - y^{(1)} + \ldots + c_1 \cdot x^{(K)} + c_2 - y^{(K)} = 0.$$

Bringing terms proportional to the unknowns c_1 and c_2 together and moving all other terms to the right-hand side, we get

$$c_1 \cdot \overline{x} + c_2 \cdot K = \overline{y}, \tag{1}$$

where we denoted

$$\overline{x} = \sum_{k=1}^{K} x^{(k)} \text{ and } \overline{y} = \sum_{k=1}^{K} y^{(k)}.$$

Differentiating this expression with respect to c_1 and equating the derivative to 0, we get

$$2 \cdot x^{(1)} \cdot \left(c_1 \cdot x^{(1)} + c_2 - y^{(1)}\right) + \ldots + 2 \cdot x^{(K)} \cdot \left(c_1 \cdot x^{(K)} + c_2 - y^{(K)}\right) = 0.$$

Dividing both sides of this equality by 2 and opening the parentheses, we conclude that

$$c_1 \cdot \left(x^{(1)}\right)^2 + c_2 \cdot x^{(1)} - x^{(1)} \cdot y^{(1)} + \ldots + c_1 \cdot \left(x^{(K)}\right)^2 + c_2 \cdot x^{(K)} - x^{(K)} \cdot y^{(K)} = 0.$$

Bringing terms proportional to the unknowns c_1 and c_2 together and moving all other terms to the right-hand side, we get

$$c_1 \cdot \overline{x^2} + c_2 \cdot \overline{x} = \overline{x \cdot y}, \tag{2}$$

where we denoted

$$\overline{x^2} = \sum_{k=1}^{K} \left(x^{(k)}\right)^2 \text{ and } \overline{x \cdot y} = \sum_{k=1}^{K} x^{(k)} \cdot y^{(k)}.$$

Now, we get a system of two linear equations (1) and (2) to find the two unknowns c_1 and c_2

To find the value c_1, we can do the following:

- first, we multiply both sides of the equation (1) by the coefficient $\overline{x}$ at c_2 in the equation (2), getting

$$c_1 \cdot (\overline{x})^2 + c_2 \cdot K \cdot \overline{x} = \overline{y} \cdot \overline{x};$$

- then, we multiply both sides of the equation (2) by the coefficient K at c_2 in the equation (1), getting

$$c_1 \cdot K \cdot \overline{x^2} + c_2 \cdot K \cdot \overline{x} = K \cdot \overline{x \cdot y};$$

- after that, we subtract the resulting equalities; in the resulting difference, the coefficient at c_2 is 0, so we get

$$c_1 \cdot \left(K \cdot \overline{x^2} - (\overline{x})^2\right) = K \cdot \overline{x \cdot y} - \overline{x} \cdot \overline{y};$$

- we then divide both sides by the coefficient at c_1, resulting in

$$c_1 = \frac{K \cdot \overline{x \cdot y} - \overline{x} \cdot \overline{y}}{K \cdot \overline{x^2} - (\overline{x})^2}.$$

To find the value c_2, we can do the following:

- first, we multiply both sides of the equation (1) by the coefficient $\overline{x^2}$ at c_1 in the equation (2), getting

$$c_1 \cdot \overline{x} \cdot \overline{x^2} + c_2 \cdot K \cdot \overline{x^2} = \overline{y} \cdot \overline{x^2};$$

- then, we multiply both sides of the equation (2) by the coefficient $\overline{x}$ at c_1 in the equation (1), getting

$$c_1 \cdot \overline{x^2} \cdot \overline{x} + c_2 \cdot (\overline{x})^2 = \overline{x \cdot y} \cdot \overline{x};$$

- after that, we subtract the resulting equalities; in the resulting difference, the coefficient at c_1 is 0, so we get

$$c_2 \cdot \left(K \cdot \overline{x^2} - (\overline{x})^2\right) = \overline{x^2} \cdot \overline{y} - \overline{x} \cdot \overline{x \cdot y};$$

- we then divide both sides by the coefficient at c_2, resulting in

$$c_2 = \frac{\overline{x^2} \cdot \overline{y} - \overline{x} \cdot \overline{x \cdot y}}{K \cdot \overline{x^2} - (\overline{x})^2}.$$

Numerical example. Suppose that we have the following $K = 3$ measurement results:

- $x^{(1)} = -1$, $y^{(1)} = -2$;
- $x^{(2)} = 0$, $y^{(2)} = 1$;
- $x^{(3)} = 1$, $y^{(3)} = 2$.

In this case:

$$\overline{x} = (-1) + 0 + 1 = 0; \quad \overline{y} = (-2) + 1 + 2 = 1;$$

$$\overline{x^2} = (-1)^2 + 0^2 + 1^2 = 2; \quad \overline{x \cdot y} = (-1) \cdot (-2) + 0 \cdot 1 + 1 \cdot 2 = 4.$$

Thus,

$$c_1 = \frac{3 \cdot 4 - 0 \cdot 1}{3 \cdot 2 - 0^2} = \frac{12}{6} = 2;$$

$$c_2 = \frac{2 \cdot 1 - 0 \cdot 4}{3 \cdot 2 - 0^2} = \frac{2}{6} = 0.333\ldots$$

So, the desired dependence is $y = 2x + 0.333\ldots$

General case. For linear dependence on c_i, we get a system of linear equations, for solving which there are efficient algorithms, starting with Gaussian elimination. In the general case of non-linear dependence, if we simply differentiate and equate derivatives to 0, we get a system of non-linear equations. In this case, we need to use more complex techniques.

Alternatively, we can use more sophisticated optimization techniques, such as gradient descent, genetic algorithms, etc. There are many optimization algorithms – their description can fill yet another course – and many packages implementing these algorithms. Usually, people use some of the known optimization algorithms, in most cases, this is more efficient than coming up with your own algorithm.

So how can we design an explainable fuzzy system. We start with expert rules – this what makes this approach explainable. We then use general fuzzy methodology – explained in the previous lectures – to find the first-approximation dependence $y = f(x_1, \ldots, x_n)$.

When applying the fuzzy methodology, we used some parameters – e.g., for negligible, we selected 5 as the borderline value starting with which the difference is absolutely not negligible. The choice of these parameters is rather arbitrary. For example, to describe what is negligible, we could use 4 or 6 instead of 5.

So, instead of picking a single such value:

- we make this value a parameter, and then
- we find the values of all these parameters for which, for each k, the predictions of the resulting fuzzy system are the closest to the desired value $y^{(k)}$.

In fuzzy techniques, this process is known as *tuning*.