Estimating Quality of Support Vector Machines Learning Under Probabilistic and Interval Uncertainty: Algorithms and Computational Complexity

Canh Hao Nguyen[1], Tu Bao Ho[1],
and Vladik Kreinovich[2]

[1]School of Knowledge Science
JAIST, Japan

[2]University of Texas at El Paso
El Paso, TX 79968, USA
vladik@utep.edu

1. Outline

- Support Vector Machines (SVM) is one of the most widely used technique in machines leaning.
- It is desirable to learn how well this classification fits the data.
- There exist several measures of fit. item Among them the most widely used is kernel target alignment.
- These measures, however, assume that the data are known exactly.
- In reality, the data points are only known with uncertainty.
- We show how to take this uncertainty into account.

2. Machine learning: reminder

- *Machine learning:*
  - we have several objects from different classes;
  - each object x is described by d parameters $x_1, \ldots, x_d$;
  - thus, we have several points

    $$x^{(i)} = (x_1^{(i)}, \ldots, x_k^{(i)}, \ldots, x_d^{(i)}), \quad 1 \le i \le n$$

    from different classes;
  - we must classify a new point x to one of these classes.

itemLinear classification: find a linear function for which

  - $c_1 \cdot x_1^{(i)} + \ldots + c_d \cdot x_d^{(i)} > c_0$ for all *positive* examples, and
  - $c_1 \cdot x_1^{(i)} + \ldots + c_d \cdot x_d^{(i)} < c_0$ for all *negative* examples.
- *Limitations:* not always possible (e.g., exclusive or).

3. Support Vector Machines

- There exists a continuous function $f(x_1, \ldots, x_d)$ s.t.:
  - $f(x_1^{(i)}, \ldots, x_d^{(i)}) > 0$ for all positive examples and
  - $f(x_1^{(i)}, \ldots, x_d^{(i)}) < 0$ for all negative examples.
- A continuous function $f(x_1, \ldots, x_d)$ can be, with arbitrary accuracy, approximated by a polynomial

$$\widetilde{f}(x_1, \ldots, x_d) = c_0 + c_1 \cdot x_1 + \ldots + c_d \cdot x_d + \sum_{k=1}^{d} \sum_{l=1}^{d} c_{kl} \cdot x_k \cdot x_l + \ldots$$

- *Conclusion:* we linearly separate points

$$(x_1, \ldots, x_n, x_1^2, x_1 \cdot x_2, \ldots).$$

- *Comment:* instead of polynomials, we could use trigonometric polynomials, Gaussians, etc.

4. Support Vector Machines: general description

- We start with points

$$x^{(1)}, \ldots, x^{(n)}$$

in the d-dimensional space.

- We map each point x into a point

$$\phi(x) = (\phi_1(x), \ldots, \phi_p(x), \ldots, \phi_N(x))$$

in a higher-dimensional space (of dimension $N \geq d$).

- We use linear separation to separate the resulting points

$$\phi(x^{(1)}), \ldots, \phi(x^{(n)})$$

in the N-dimensional space.

5. Need to estimate classification quality

- *Main idea:* points close to $x^{(i)}$ should be classified to the same class as $x^{(i)}$.
- *Geometric reformulation:* all the examples are sufficiently far away from the separating surface.
- *Auxiliary notion:* kernel matrix $k_{ij} \stackrel{\text{def}}{=} \langle \phi(x^{(i)}), \phi(x^{(j)}) \rangle$, where $\langle \phi, \phi' \rangle \stackrel{\text{def}}{=} \sum_{p=1}^{N} \phi_p \cdot \phi'_p$.
- *Ideal situation:* separation as sharp as possible:
  - the vectors $\phi(x^{(i)})$ corresponding to the positive examples to be equal to some unit vector e;
  - all the vectors corresponding to the negative examples to be equal to $-e$.
- In this case, the kernel matrix is $y_i \cdot y_j$, where $y_i = 1$ for positive examples and $y_i = -1$ for negative examples.

6. Need to estimate classification quality (cont-d)

- *Reminder:* ideal kernel matrix is $y_i \cdot y_j$,
- The closer k_{ij} to this ideal matrix, the better.
- Matrices are $N \times N$ vectors, so closeness can be measured as the cosine between these vectors:

$$A = \frac{\sum\limits_{i=1}^{n} \sum\limits_{j=1}^{n} k_{ij} \cdot y_i \cdot y_j}{n \cdot \sqrt{\sum\limits_{i=1}^{n} \sum\limits_{j=1}^{n} k_{ij}^2}}.$$

- This cosine is called kernel target alignment (KTA).

7. Class Separability Measure (CSM)

- *Main idea:*
  - data points within each class should be close to each other, while
  - data points from different classes should be far away from each other.
- *Reformulation:* "within-class" scatter s_w should be much smaller than the "between-classes" scatter s_b.
- Each class is naturally characterized by its average.
- For each data point, its contribution:
  - to s_w can be described as a (squared) distance from this data point to the average, and
  - to s_b is a (squared) distance between the average of this class and the overall average.

8. Class Separability Measure (CSM): cont-d

- In the SVM approach, each data point $x^{(i)}$ is represented by the vector $\phi(x^{(i)})$.
- For each class S_c, $c = 1, 2, \ldots, C$, let n_c denote the number of data points classified into this class.
- Let ϕ_c denote the average of all $\phi(x^{(i)})$ from S_c.
- Let ϕ denote the average of all n vectors $\phi(x^{(i)})$.
- Then, we can define the *within-class scatter* s_w as

$$s_w \stackrel{\text{def}}{=} \sum_{c=1}^{C} \sum_{i \in S_c} \|\phi(x^{(i)}) - \phi_c\|^2.$$

- *Between-classes scatter* is

$$s_b \stackrel{\text{def}}{=} \sum_{c=1}^{C} n_c \cdot \|\phi_c - \phi\|^2.$$

- A classification is of good quality if $s_w \ll s_b$.

9. CSM in terms of the kernel matrix

- *Case:* two classes, with n^+ and n^- examples.
- First, for every i, we compute

$$a_i^+ = \frac{1}{n^+} \sum_{j:y_j=1} k_{ij}; \quad a_i^- = \frac{1}{n^-} \sum_{j:y_j=-1} k_{ij}.$$

- Second, we compute

$$a^{++} = \frac{1}{n^+} \sum_{j:y_j=1} a_i^+, \quad a^{+-} = \frac{1}{n^-} \sum_{j:y_j=-1} a_i^+,$$

$$a^{-+} = \frac{1}{n^+} \sum_{j:y_j=1} a_i^-, \quad a^{--} = \frac{1}{n^+} \sum_{j:y_j=-1} a_i^-,$$

and $s_b = a^{++} - a^{+-} - a^{-+} + a^{--}$.

- Then, we compute

$$s_w = \sum_{i=1}^{n} k_{ii} - n^+ \cdot a^{++} - n^- \cdot a^{--}.$$

10. Need for an alternative quality measure

- In many practical examples, KTA and CSM provides a reasonable estimate for the quality of fit.
- However, there are examples when KTA and CSM are counter-intuitive.
- *Example:* for some coordinate $\phi_p(x)$, we have
  - $\phi_p(x^{(i)}) = 1$ for all positive examples and
  - $\phi_p(x^{(i)}) = -1$ for all negative examples.
- *Intuitively:* we have a perfect classification.
- However, the values $\phi_q(x^{(i)})$ for $q \neq p$ may be widely scattered.
- *Result:* we can have a huge value of the within-class scatter $s_w \gg s_b$.

11. Feature-Space Measure (FSM)

- *Main idea:* take into account only the scatter in the direction between the centers ϕ^- and ϕ^+.
- First, we compute:
  - the average ϕ^+ of the values $\phi(x^{(i)})$ for all the positive examples and
  - the average ϕ^- of the values $\phi(x^{(i)})$ for all the negative examples.
- In the ideal case, as we have mentioned, we should have $\phi^+ = e$ and $\phi^- = -e$ for some unit vector e.
- Then, we estimate the vector e as the unit vector in the direction of the difference $\phi^+ - \phi^-$, i.e., as

$$e = \frac{\phi^+ - \phi^-}{\|\phi^+ - \phi^-\|}.$$

12. Feature-Space Measure (FSM): cont-d

- Next, for each example i, we compute the projection $p_i = \langle \phi(x^{(i)}), e \rangle$ of the vector $\phi^{(i)}$ to the direction e.
- We compute the population means

$$p^+ = \frac{1}{n^+} \cdot \sum_{i:y_i=1} p_i; \quad p^- = \frac{1}{n^-} \cdot \sum_{i:y_i=-1} p_i.$$

- We compute population variances

$$V^+ = \frac{1}{n^+ - 1} \cdot \sum_{i:y_i=1} (p_i - p^+)^2; \quad V^- = \frac{1}{n^- - 1} \cdot \sum_{i:y_i=-1} (p_i - p^-)^2.$$

- Then, we compute the desired value

$$\frac{\sqrt{V^+} + \sqrt{V^-}}{\|\phi^+ - \phi^-\|}.$$

13. Need to take into account probabilistic and interval uncertainty

- *We assumed:* that all the values $x_k^{(i)}$ are known exactly.
- *In practice:* these values comes from measurements.
- *Fact:* measurement are never 100% accurate: there is always a measurement error

$$\Delta x_k^{(i)} \stackrel{\text{def}}{=} \widetilde{x}_k^{(i)} - x_k^{(i)} \neq 0.$$

- *Question:* how these measurement errors affect different measures Q of quality of fit?
- *Two possibilities:*
  - *engineering approach:* $\Delta x_k^{(i)}$ are normally distributed with 0 mean and known standard deviation $\sigma_k^{(i)}$;
  - *realistic approach:* we only know upper bounds $\Delta_k^{(i)}$ on the measurement errors: $|\Delta x_k^{(i)}| \le \Delta_k^{(i)}$.

14. Practical results

- *Practical case:* measurements are reasonably accurate.
- *Conclusion:* we can ignore terms quadratic in $\Delta x_k^{(i)}$ and get

$$\Delta Q = \sum_{i,k} c_{ik} \cdot \Delta x_k^{(i)}, \text{ where } c_{ik} \stackrel{\text{def}}{=} \frac{\partial Q}{\partial x_k^{(i)}}.$$

- *Probabilistic case:* ΔQ is normally distributed, with 0 mean and standard deviation

$$\sigma^2 = \sum_{i,k} c_{ik}^2 \cdot \sigma_{ik}^2.$$

- *Interval case:* the largest possible value of ΔQ is

$$\Delta = \sum_{i,k} |c_{ik}| \cdot \Delta_{ik}.$$

- *What we did:* for the specific quality metrics, provided easier-to-compute versions of these formulas.

15. General case: theoretical results

- We consider situations in which measurements are accurate – and terms quadratic in $\Delta x_k^{(i)}$ can be ignored.
- *Natural question:* what if measurements are not that accurate – and quadratic terms cannot be ignored?
- *Interval case:* we want to compute the exact range of Q when

$$x_k^{(i)} \in \mathbf{x}_k^{(i)} \stackrel{\text{def}}{=} [\widetilde{x}_k^{(i)} - \Delta_k^{(i)}, \widetilde{x}_k^{(i)} + \Delta_k^{(i)}],$$

  i.e., the range:

$$[\underline{Q}, \overline{Q}] = \{Q(\{x_k^{(i)}\}) : x_k^{(i)} \in \mathbf{x}_k^{(i)}, 1 \le i \le n, 1 \le k \le d\}.$$

- *Theoretical results:* computing the exact range of Q is NP-hard for all three quality metrics.
- *Intuitive meaning of NP-hardness:* no efficient algorithm always correctly computes $[\underline{Q}, \overline{Q}]$.

16. Acknowledgments

This work was supported in part by:

- by the Japan Advanced Institute of Science and Technology (JAIST) International Joint Research Grant 2006-08,
- by Texas Department of Transportation contract No. 0-5453,
- by National Science Foundation grants HRD-0734825, EAR-0225670, and EIA-0080940, and
- and by the Max Planck Institut für Mathematik.