Introduction à l'apprentissage automatique - Polytech
Empirical risk minimization

Caio Corro

Université Paris-Saclay, CNRS, Laboratoire Interdisciplinaire des Sciences du Numérique, 91400, Orsay, France

Theory

Problem

Train/test objective mismatch

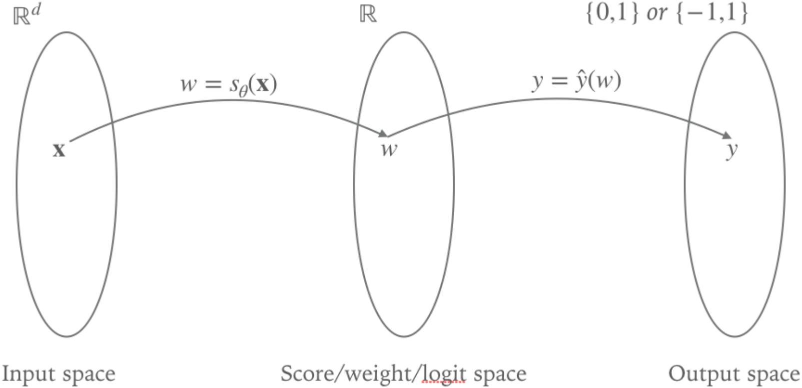
- At training time we minimize a loss function, *e.g.* the hinge loss
- At test time we evaluate the model using a different function, *e.g.* classification error / classification accuracy

There is a mismatch between the two objective!

Goal

Can we prove that minimizing a given loss function also minimizes the test time objective?

Assumptions



- We assume the class of scoring functions S is "rich enough" (*i.e.* set of all measurable mappings)
- We assume we have access to an infinite number of training datapoints

0-1 loss function

Definition of the 0-1 loss function

Count the number of classification errors

$\Rightarrow$ exactly what we aim to minimize at test time

Binary classification

- If $Y = \{0, 1\}$: $\quad \ell_{0-1}(y, w) = \mathbb{1}[(2y-1)w < 0] = [\,\mathrm{sign}(w) \neq 2y - 1]$
- If $Y = \{-1, 1\}$: $\quad \ell_{0-1}(y, w) = \mathbb{1}[y \times w < 0] = [\,\mathrm{sign}(w) \neq y]$

Multiclass classification

- If $Y = E(k)$:
$$\ell_{0-1}(\boldsymbol{y}, \boldsymbol{w}) = \begin{cases} 0 & \text{if } \boldsymbol{y} = \arg\max_{\boldsymbol{y}' \in E(k)} \langle \boldsymbol{w}, \boldsymbol{y}' \rangle, \\ 1 & \text{otherwise.} \end{cases}$$
- If $Y = \{1, ..., k\}$:
$$\ell_{0-1}(y, \boldsymbol{w}) = \mathbb{1}[y \neq \underset{y' \in \{1,\dots,k\}}{\arg\max}\ w_{y'}]$$

Bayes risk

Notations

- $\mathbf{x}$: random variable representing inputs in $\mathbb{R}^d$
- $\mathbf{y}$: random variable representing outputs, problem dependent
- $p(\mathbf{x}, \mathbf{y})$: data distribution, unknown in practice but we can still use it for theory
- S: set of scoring functions

WARNING

The input/output mapping in the data distribution is not necessarily deterministic, an input $\boldsymbol{x} \in \mathbb{R}^d$ may be associated with several outputs with a non null probability.

Bayes risk

Notations

- $\mathbf{x}$: random variable representing inputs in $\mathbb{R}^d$
- $\mathbf{y}$: random variable representing outputs, problem dependent
- $p(\mathbf{x}, \mathbf{y})$: data distribution, unknown in practice but we can still use it for theory
- S: set of scoring functions

WARNING

The input/output mapping in the data distribution is not necessarily deterministic, an input $\boldsymbol{x} \in \mathbb{R}^d$ may be associated with several outputs with a non null probability.

Risk of a scoring function

The risk of a given scoring function $s \in S$ is denoted:

$$r(s) = \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \ell_{0-1}(\mathbf{y}, s(\mathbf{x}))\ ]$$

or, in other words, it is the classification error probability for classifier based on s.

Minimum Bayes risk

Risk of a scoring function

The risk of a given scoring function $s \in S$ is denoted:

$$r(s) = \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \ell_{0-1}(\mathbf{y}, s(\mathbf{x}))\ ]$$

or, in other words, it is the classification error probability for classifier based on s.

Minimum Bayes risk

It seems that it is a good idea to aim for a model of minimum Bayes risk:

$$\begin{aligned} r^* = \inf_{s \in S} r(s) &= \inf_{s \in S} \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \ell_{0-1}(\mathbf{y}, s(\mathbf{x}))\ ] \\ &= \inf_{s \in S} \mathbb{E}_{\mathbf{x}}\left[\ \mathbb{E}_{\mathbf{y}|\mathbf{x}}[\ \ell_{0-1}(\mathbf{y}, s(\mathbf{x}))\ ]\ \right] \\ &= \mathbb{E}_{\mathbf{x}}\left[\ 1 - \max_{\boldsymbol{y} \in Y} p(\mathbf{y} = \boldsymbol{y}|\mathbf{x})\ \right] \end{aligned}$$

Surrogate losses

Recall

In practice we cannot use the 0-1 loss:

- non-convex
- partial derivatives are null almost everywhere

It is known that minimizing the 0-1 loss is a NP-hard problem (Ben-David et al., 2003; Feldman et al., 2009)

Main idea

- Replace the 0-1 loss with a surrogate loss
- hope (prove?) that minimizing the surrogate loss leads to a classifier of minimum Bayes risk

Classification calibration

Surrogate risk

Let $\widetilde{\ell}$ be a surrogate loss.
The surrogate risk $\widetilde{r}(s)$ of scoring function $s \in S$ is defined as:

$$\widetilde{r}(s) = \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \widetilde{\ell}(\mathbf{y}, s(\mathbf{x}))\ ],$$

and the optimal surrogate risk $\widetilde{r}^*$ is defined as:

$$\widetilde{r}^* = \inf_{s \in S} \widetilde{r}(s) = \inf_{s \in S} \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \widetilde{\ell}(\mathbf{y}, s(\mathbf{x}))\ ]$$

Classification calibration

Surrogate risk

Let $\widetilde{\ell}$ be a surrogate loss.
The surrogate risk $\widetilde{r}(s)$ of scoring function $s \in S$ is defined as:

$$\widetilde{r}(s) = \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \widetilde{\ell}(\mathbf{y}, s(\mathbf{x}))\ ],$$

and the optimal surrogate risk $\widetilde{r}^*$ is defined as:

$$\widetilde{r}^* = \inf_{s \in S} \widetilde{r}(s) = \inf_{s \in S} \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \widetilde{\ell}(\mathbf{y}, s(\mathbf{x}))\ ]$$

Definition

A surrogate loss $\widetilde{\ell}$ is said to be **classification calibrated** if and only if:

$$s^* \in \arg\min_{s \in S} \widetilde{r}(s) \implies r(s^*) = r^*,$$

i.e. minimizing the surrrogate risk leads to a prediction model of optimal Bayes risk.
This property is also called Bayes consistency and Fisher consistency.

Pointwise analysis 1/2

Assumption

We assume the class of scoring function S is "rich enough"
(*i.e.* set of all measurable mappings)

$$
\begin{aligned}
r^* = \inf_{s \in S} r(s) &= \inf_{s \in S} \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \ell_{0-1}(s(\mathbf{x}), \mathbf{y})\ ] \\
&= \inf_{\forall \boldsymbol{x} \in X:\ \boldsymbol{w}^{(x)} \in \mathbb{R}^k} \mathbb{E}_{\mathbf{x}}\left[\ \mathbb{E}_{\mathbf{y}|\mathbf{x}}[\ \ell_{0-1}(\mathbf{y}, \boldsymbol{w}^{(\mathbf{x})})\ ]\ \right] \\
&= \mathbb{E}_{\mathbf{x}} \underbrace{\left[\ \inf_{\boldsymbol{w}^{(\mathbf{x})} \in \mathbb{R}^k} \mathbb{E}_{\mathbf{y}|\mathbf{x}}\left[\ \ell_{0-1}(\mathbf{y}, \boldsymbol{w}^{(\mathbf{x})})\ \right]\ \right]}_{\text{We can focus on this part only}}
\end{aligned}
$$

Pointwise analysis 1/2

Assumption

We assume the class of scoring function S is "rich enough"
(*i.e.* set of all measurable mappings)

Pointwise risk

Let $\boldsymbol{x} \in X$ such that $p(\mathbf{x} = \boldsymbol{x}) > 0$. We redefine the concept of (optimal) surrogate/Bayes risk as follows:

$$r(\boldsymbol{w}) = \mathbb{E}_{\mathbf{y}|\mathbf{x}=\boldsymbol{x}}[\ \ell_{0-1}(\mathbf{y}, \boldsymbol{w})\ ] \qquad \widetilde{r}(\boldsymbol{w}) = \mathbb{E}_{\mathbf{y}|\mathbf{x}=\boldsymbol{x}}[\ \widetilde{\ell}(\mathbf{y}, \boldsymbol{w})\ ]$$

$$r^* = \inf_{\boldsymbol{w} \in \mathbb{R}^k} r(\boldsymbol{w}) \qquad \widetilde{r}^* = \inf_{\boldsymbol{w} \in \mathbb{R}^k} \widetilde{r}(\boldsymbol{w})$$

where $\boldsymbol{w} \in \mathbb{R}^k$ should be interpreted as the output of the scoring function, *i.e.* $\boldsymbol{w} = s(\boldsymbol{x})$.

Classification calibration for binary classification

Surrogate losses

Let $Y = \{-1, 1\}$.

- Perceptron loss: $\widetilde{\ell}_{\text{perceptron}}(y, w) = \max(0, -y \times w)$
- Hinge loss: $\widetilde{\ell}_{\text{hinge}}(y, w) = \max(0, 1 - y \times w)$
- Squared hinge loss: $\widetilde{\ell}_{\text{s. hinge}}(y, w) = \max(0, 1 - y \times w)^2$
- Exponential loss: $\widetilde{\ell}_{\exp}(y, w) = \exp(-y \times w)$
- Negative log-likelihood (NLL): $\widetilde{\ell}_{\text{nll}}(y, w) = \log(1 + \exp(-y \times w))$
- Quadratic error (or squared error): $\widetilde{\ell}_{\text{quad.}}(y, w) = (y \times w - 1)^2$

Classification calibration

All these surrogate losses are classification calibrated **except the perceptron loss**.

Classification calibration for binary classification

Surrogate risk

$$
\begin{aligned}
\widetilde{r}(w) &= \mathbb{E}_{\mathbf{y}|\mathbf{x}=\boldsymbol{x}}[\ \widetilde{\ell}(\mathbf{y}, w)\ ] \\
&= p(\mathbf{y}=1|\mathbf{x}=\boldsymbol{x}) \times \widetilde{\ell}(1, w) \quad + \quad p(\mathbf{y}=-1|\mathbf{x}=\boldsymbol{x}) \times \widetilde{\ell}(-1, w) \\
&= \mu \times \widetilde{\ell}(1, w) \quad + \quad (1-\mu) \times \widetilde{\ell}(-1, w)
\end{aligned}
$$

where $\mu = p(\mathbf{y}=1|\mathbf{x}=\boldsymbol{x})$.
The optimal surrogate risk is:

$$
\widetilde{r}^* = \inf_{w \in \mathbb{R}} \widetilde{r}(w) = \inf_{w \in \mathbb{R}} \ \mu \times \widetilde{\ell}(1, w) + (1-\mu) \times \widetilde{\ell}(-1, w)
$$

Classification calibration for binary classification

Surrogate risk

$$\begin{aligned}\widetilde{r}(w) &= \mathbb{E}_{\mathbf{y}|\mathbf{x}=\boldsymbol{x}}[\ \widetilde{\ell}(\mathbf{y}, w)\ ] \\ &= p(\mathbf{y}=1|\mathbf{x}=\boldsymbol{x}) \times \widetilde{\ell}(1, w) \quad + \quad p(\mathbf{y}=-1|\mathbf{x}=\boldsymbol{x}) \times \widetilde{\ell}(-1, w) \\ &= \mu \times \widetilde{\ell}(1, w) \quad + \quad (1-\mu) \times \widetilde{\ell}(-1, w)\end{aligned}$$

where $\mu = p(\mathbf{y}=1|\mathbf{x}=\boldsymbol{x})$.
The optimal surrogate risk is:

$$\widetilde{r}^* = \inf_{w\in\mathbb{R}} \widetilde{r}(w) = \inf_{w\in\mathbb{R}} \ \mu \times \widetilde{\ell}(1, w) + (1-\mu) \times \widetilde{\ell}(-1, w)$$

Intuition

The surrogate loss $\widetilde{\ell}$ is classification calibrated if the minimzer w^* satisfies:

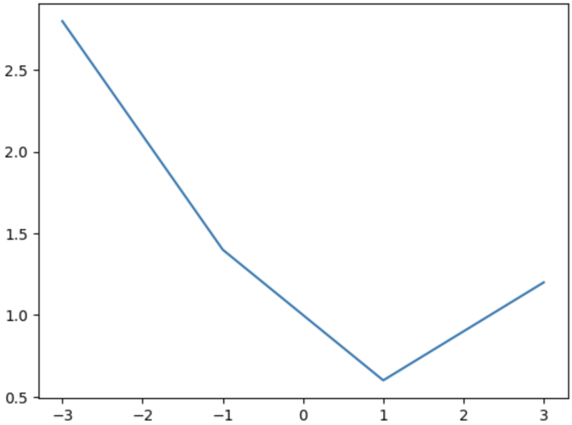
- $\mu > 0.5 \implies w^* \geq 0$ (the class 1 is the most probable class for input $\boldsymbol{x}$)
- $\mu < 0.5 \implies w^* < 0$ (the class 1 is the most probable class for input $\boldsymbol{x}$)
- $\mu = 0.5$: this case is not important.

[4]:
```python
def hinge(y, w):
    return max(0, 1 - y * w)

mu = 0.7
ws = np.linspace(-3, 3, 100)
ls = [mu * hinge(1, w) + (1 - mu) * hinge(-1, w) for w in ws]

plt.plot(ws, ls)
```

[4]:
```
[<matplotlib.lines.Line2D at 0x10f41f0a0>]
```

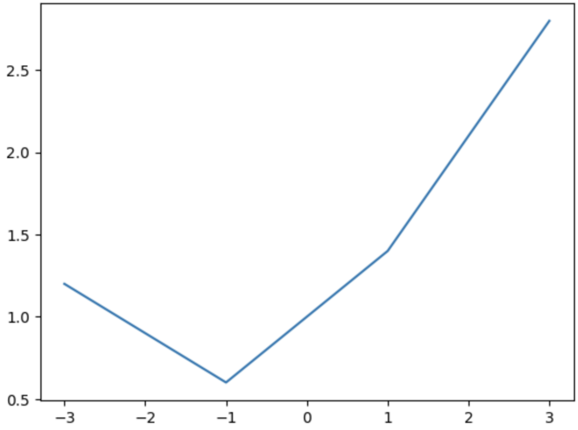


[5]:
```python
def hinge(y, w):
    return max(0, 1 - y * w)

mu = 0.3
ws = np.linspace(-3, 3, 100)
ls = [mu * hinge(1, w) + (1 - mu) * hinge(-1, w) for w in ws]

plt.plot(ws, ls)
```

[5]: [<matplotlib.lines.Line2D at 0x10f2313c0>]



[6]:

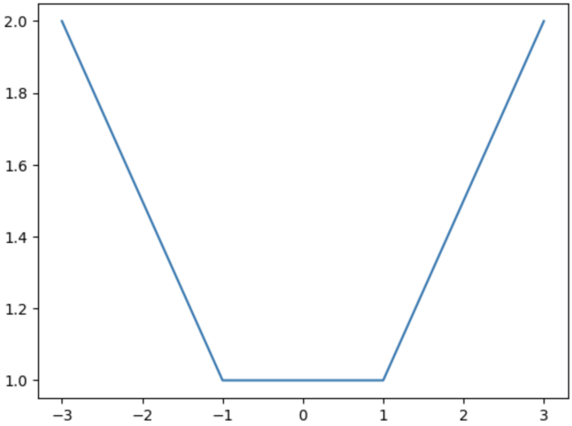
```python
def hinge(y, w):
    return max(0, 1 - y * w)

mu = 0.5
ws = np.linspace(-3, 3, 100)
ls = [mu * hinge(1, w) + (1 - mu) * hinge(-1, w) for w in ws]

plt.plot(ws, ls)
```

[6]:

```
[<matplotlib.lines.Line2D at 0x10ff3c820>]
```



[7]:

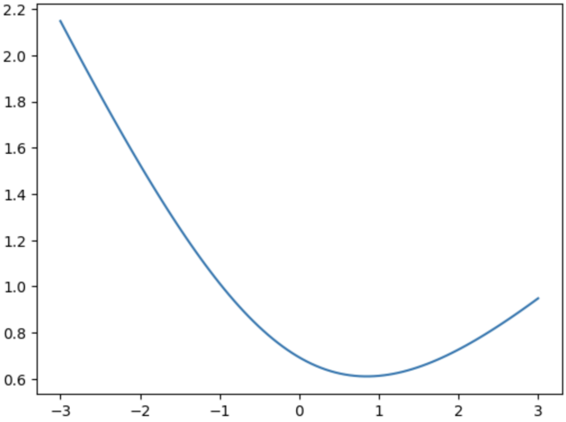
```
def nll(y, w):
    return np.log(1 + np.exp(- y * w))

mu = 0.7
ws = np.linspace(-3, 3, 100)
ls = [mu * nll(1, w) + (1 - mu) * nll(-1, w) for w in ws]

plt.plot(ws, ls)
```

[7]:

```
[<matplotlib.lines.Line2D at 0x10ff9e290>]
```

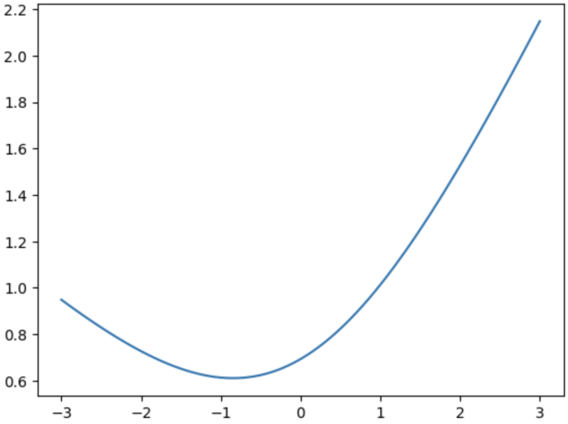


```
[8]: def nll(y, w):
         return np.log(1 + np.exp(- y * w))

     mu = 0.3
     ws = np.linspace(-3, 3, 100)
     ls = [mu * nll(1, w) + (1 - mu) * nll(-1, w) for w in ws]

     plt.plot(ws, ls)
```

```
[8]: [<matplotlib.lines.Line2D at 0x11002ded0>]
```

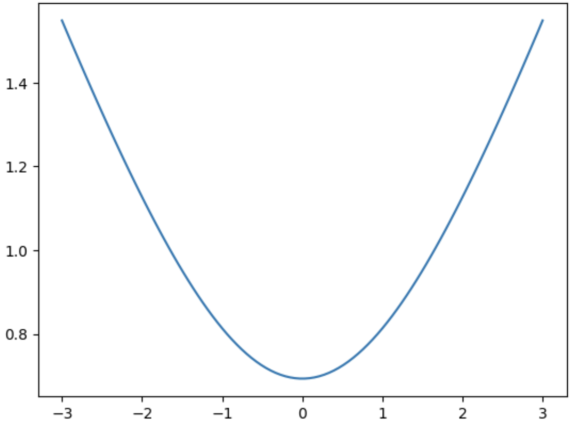


[9]:
```python
def nll(y, w):
    return np.log(1 + np.exp(- y * w))

mu = 0.5
ws = np.linspace(-3, 3, 100)
ls = [mu * nll(1, w) + (1 - mu) * nll(-1, w) for w in ws]

plt.plot(ws, ls)
```

[9]: `[<matplotlib.lines.Line2D at 0x1100b94b0>]`

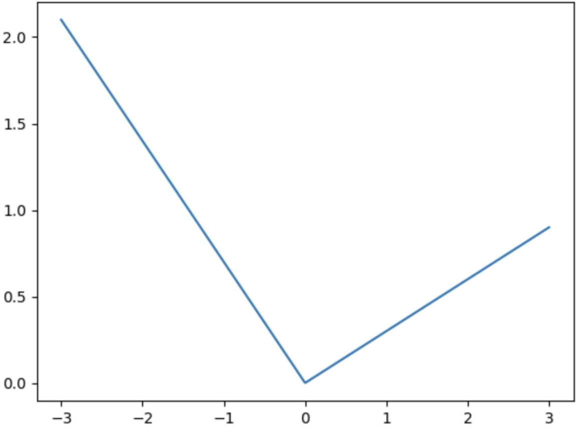


```
[14]: def perceptron(y, w):
          return max(0, - y * w)

      mu = 0.7
      ws = np.linspace(-3, 3, 1000)
      ls = [mu * perceptron(1, w) + (1 - mu) * perceptron(-1, w) for w in ws

      plt.plot(ws, ls)
```

```
[14]: [<matplotlib.lines.Line2D at 0x110325a80>]
```



[15]:

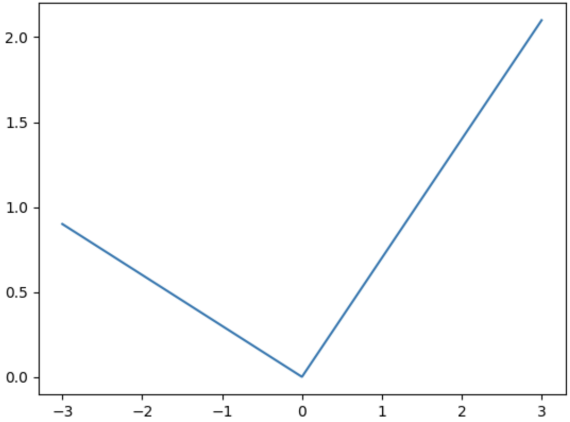
```
def perceptron(y, w):
    return max(0, - y * w)

mu = 0.3
ws = np.linspace(-3, 3, 1000)
ls = [mu * perceptron(1, w) + (1 - mu) * perceptron(-1, w) for w in ws]

plt.plot(ws, ls)
```

[15]:

```
[<matplotlib.lines.Line2D at 0x11039ad70>]
```

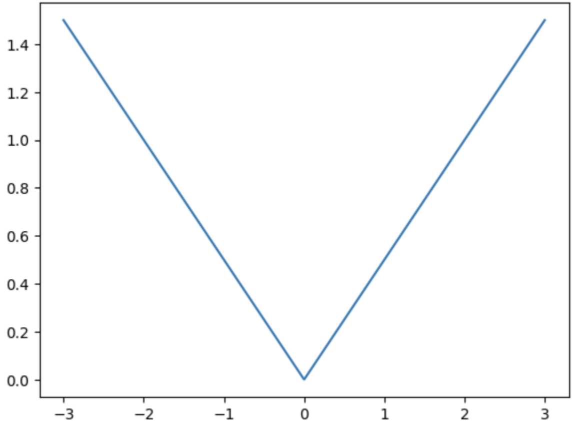


[16]:
```python
def perceptron(y, w):
    return max(0, - y * w)

mu = 0.5
ws = np.linspace(-3, 3, 1000)
ls = [mu * perceptron(1, w) + (1 - mu) * perceptron(-1, w) for w in ws]

plt.plot(ws, ls)
```

[16]:
```
[<matplotlib.lines.Line2D at 0x1104156f0>]
```



Sufficient conditions in the binary case

Let $\widetilde{\ell} : \{-1, 1\} \times \mathbb{R} \to \mathbb{R}_+$ be a surrogate loss function that can be rewrittent as:

$$\widetilde{\ell}(y, w) = \phi(yw)$$

where $\phi : \mathbb{R} \to \mathbb{R}_+$ is function.
$\Rightarrow$ all previously presented binary loss function can be rewritten under this form.

Theorem

If ϕ is convex and differentiable at 0 with $\phi'(0) < 0$,
then $\widetilde{\ell}$ is classification calibrated

Proof

See (Lin, 2004).

Classification calibration for multiclass classification

Surrogate losses

Let $Y = E(k)$.

- Hinge loss: $\widetilde{\ell}_{\text{hinge}}(\boldsymbol{y}, \boldsymbol{w}) = \max\left(0, 1 - \langle \boldsymbol{y}, \boldsymbol{w} \rangle + \max_{\boldsymbol{y}' \in E(k) \setminus \{\boldsymbol{y}\}} \langle \boldsymbol{y}', \boldsymbol{w} \rangle\right)$
- NLL: $\widetilde{\ell}_{\text{nll}}(\boldsymbol{y}, \boldsymbol{w}) = -\langle \boldsymbol{y}, \boldsymbol{w} \rangle + \log \sum_i \exp(w_i)$

Properties without proof

In the multiclass classification case:

- The hinge loss **is not** classification calibrated, see (Liu, 2007)
- The NLL loss **is** classification calibrated, see exercises

Strictly proper losses 1/2

Probabilistic prediction model

- We saw that we can learn models that predict a probability distribution over outputs, *e.g.* $p_s(\mathbf{y}|\mathbf{x})$, where the s emphasize that this is the learned model distribution, parameterized by the scoring function s.
- Classification calibration means the the most probable output in the data distribution will also be the most probable output in the model distribution
- We may want a stronger property: that the two distributions are equal

Strictly proper losses 1/2

Probabilistic prediction model

- We saw that we can learn models that predict a probability distribution over outputs, *e.g.* $p_s(\mathbf{y}|\mathbf{x})$, where the s emphasize that this is the learned model distribution, parameterized by the scoring function s.
- Classification calibration means the the most probable output in the data distribution will also be the most probable output in the model distribution
- We may want a stronger property: that the two distributions are equal

Definition

Let $p(\mathbf{y}, \mathbf{x})$ be the data distribution and $p_s(\mathbf{y}|\mathbf{x})$ the model distribution. In the pointwise setting, a surrogate loss $\widetilde{\ell}$ is **classification calibrated** if and only if the scoring function s^* that minimizes the surrogate risk leads to a model distribution equal to the data distribution:

$$\forall \boldsymbol{y} \in Y : \quad p_{s^*}(\mathbf{y} = \boldsymbol{y}|\mathbf{x} = \boldsymbol{x}) = p(\mathbf{y} = \boldsymbol{y}|\mathbf{x} = \boldsymbol{x}).$$

Strictly proper losses 1/2

Remarks

- Strict properness implies classification calibration
- The support of the data distribution must be “representable” by the model distribution
- The NLL loss is strictly proper for models whose probability parameters are computed using the sigmoid/softmax function **if the conditional data distribution has full support**.

Risk minimization decomposition

Practical issues

- We only have access to a finite training dataset
- The set of function S is not the set of all measurable mapping
- The learning algorithm may not find the optimal classifier $s \in S$, *i.e.* the following problem may be solved approximately

$$s^* \in \arg\min_{s \in S} \frac{1}{|D|} \sum_{(\boldsymbol{x},\boldsymbol{y}) \in D} \ell(\boldsymbol{y}, s(\boldsymbol{x})),$$

or in other words, in practice the computed s^* is not a minimizer.

Risk minimization decomposition

Practical issues

- We only have access to a finite training dataset
- The set of function S is not the set of all measurable mapping
- The learning algorithm may not find the optimal classifier $s \in S$, *i.e.* the following problem may be solved approximately

$$s^* \in \arg\min_{s \in S} \frac{1}{|D|} \sum_{(\boldsymbol{x},\boldsymbol{y}) \in D} \ell(\boldsymbol{y}, s(\boldsymbol{x})),$$

or in other words, in practice the computed s^* is not a minimizer.

Risk decomposition

Excess risk: $$r(s^*) - r^* \geq 0$$

Excess risk decomposition:

$$r(s^*) - r^* = \underbrace{r(s^*) - \inf_{s \in S} r(s)}_{\text{Estimation error}} + \underbrace{\inf_{s \in S} r(s) - r^*}_{\text{Approximation error}}$$