CSCE 978 Lecture 3: Risk and Loss Functions*

Stephen D. Scott

January 24, 2006

Introduction

- In Lecture 1 we mentioned our desire to infer a "good" classifier
- What does this mean?!?!
- There are many ways to define "goodness", even for binary classification

Outline

- Loss functions
  - Binary classification
  - Regression
- Expected risk
- Sections 1.3, 3.1–3.2 (also read Section 3.5)

Loss Functions

D3.1 Let $(x, y, f(x)) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Y}$ be the pattern x, its true label y and a prediction $f(x)$ of y. A loss function is a mapping $c : \mathcal{X} \times \mathcal{Y} \times \mathcal{Y} \to [0, \infty)$ with the property $c(x, y, y) = 0$ for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$

- c is always ≥ 0 so we can't use good predictions to "undo" bad ones

- It is always possible to get 0 loss on pattern x by predicting correctly

- Our choice of loss function will depend on considerations of computational complexity and statistical properties

Loss Functions
Binary Classification

- Count number of misclassifications:

$$c(x, y, f(x)) = \begin{cases} 0 & \text{if } y = f(x) \\ 1 & \text{otherwise} \end{cases}$$

- Same as above, but penalty is input-dependent:

$$c(x, y, f(x)) = \begin{cases} 0 & \text{if } y = f(x) \\ \tilde{c}(x) & \text{otherwise} \end{cases}$$

  – E.g. if $y \in \{\text{rocks}, \text{diamonds}\}$ then penalty for "false diamond" classification depends on x's weight

- Can also have different values for false positive ($y = -1$) and false negative ($y = +1$) errors

  – If $y \in \{\text{cancer}, \neg\text{cancer}\}$ then FP results in unnecessary treatment, but FN can be fatal

Loss Functions
Binary Classification
(cont'd)

- If $f(x)$ is real-valued and $y \in \{-1, +1\}$, can think of sign$(f(x))$ as prediction and $|f(x)|$ as a confidence. Then a highly confident incorrect prediction can be penalized more, as can low-confidence correct predictions:
  - Soft margin loss:

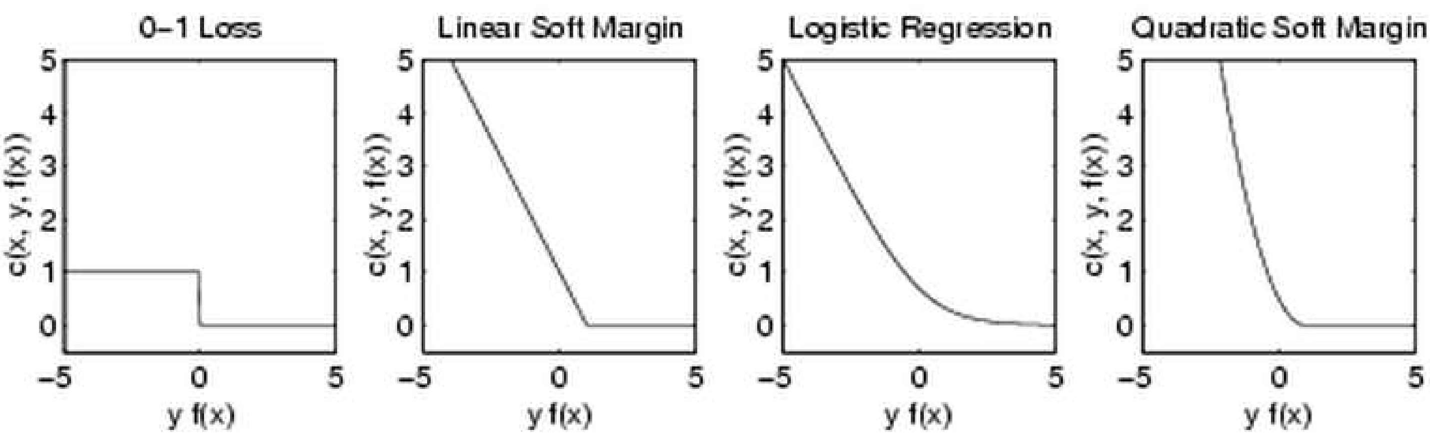
$$\begin{aligned} c(x, y, f(x)) &= \max(0, 1 - yf(x)) \\ &= \begin{cases} 0 & \text{if } yf(x) \geq 1 \\ 1 - yf(x) & \text{otherwise} \end{cases} \end{aligned}$$

  - Logistic loss:

$$c(x, y, f(x)) = \ln\left(1 + \exp(-yf(x))\right)$$

  - Both penalize a lot for confident, incorrect predictions, penalize a little for low confidence, and don't penalize much or at all for confident, correct predictions

Loss Functions
Binary Classification
(cont'd)



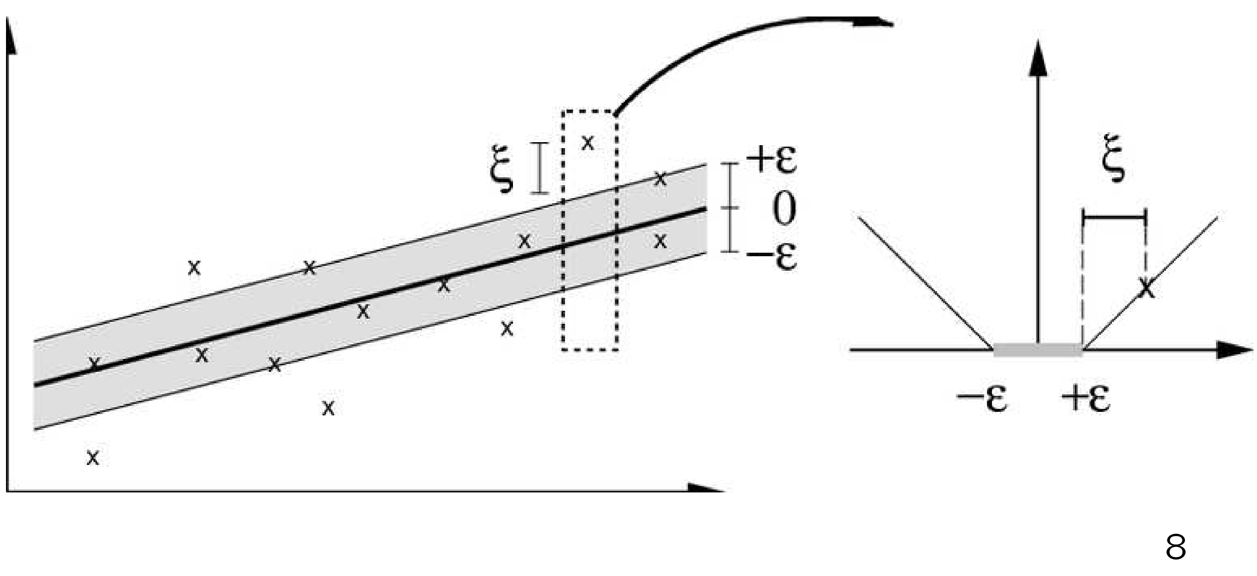
Loss Functions
Regression

- In regression, $\mathcal{Y} \subseteq \mathbb{R}$ rather than $\mathcal{Y} = \{-1, +1\}$
- Thus we're interested in how far off our prediction $f(x)$ is
- Squared loss (very popular):

$$c(x, y, f(x)) = (f(x) - y)^2$$

- Can extend soft margin loss to ϵ-insensitive loss, which doesn't penalize for close predictions:

$$c(x, y, f(x)) = |f(x) - y|_\epsilon = \max(|f(x) - y| - \epsilon, 0)$$



Loss Functions
Practical Considerations

- Want loss function to be:
  - Cheap to compute
  - Have few discontinuities in first derivative
  - Convex (to ensure unique global optimum)
  - Yield computationally efficient solutions for learning
  - Resistant to outliers/noise

Risk

- A loss function measures error on individual examples

- Our ultimate goal is to minimize loss on new (yet unseen) examples

- How do we measure this?
  - Without making certain assumptions, this is very difficult or even impossible
  - Assume that there is a probability distribution $\mathsf{P}(x, y)$ on $\mathcal{X} \times \mathcal{Y}$ that governs generation of patterns and labels
    * Assume the pairs (x, y) are drawn iid (independent and identically distributed) according to $\mathsf{P}(x, y)$
    * Generally, we won't make specific assumptions about the nature of $\mathsf{P}(x, y)$
  - $\mathsf{P}(y \mid x) =$ conditional probability of getting label y given that x is the pattern (so x could have a different label on each draw)

Risk
Definitions

- For now, assume we know all the new patterns we'll ever classify; call these the test patterns $x'_1, \ldots, x'_{m'}$ (note we do not know the labels until after we make predictions)

D3.2 When test set $x'_1, \ldots, x'_{m'}$ already known, goal is to minimize the expected error on the test set:

$$R_{\text{test}}[f] := \frac{1}{m'} \sum_{i=1}^{m'} \int_{\mathcal{Y}} c(x'_i, y, f(x'_i))\, d\mathsf{P}(y \mid x'_i)$$

- Often, minimizing $R_{\text{test}}[f]$ not realistic since typically don't know test set a priori
  - One exception: querying fixed collection of images, biological sequences, etc.

D3.3 The expected risk (expected loss) wrt P & c:

$$\begin{aligned} R[f] &:= \mathbf{E}\left[R_{\text{test}}[f]\right] = \mathbf{E}\left[c(x, y, f(x))\right] \\ &= \int_{\mathcal{X}\times\mathcal{Y}} c(x, y, f(x))\, d\mathsf{P}(x, y) \end{aligned}$$

- Not realistic since we don't know $\mathsf{P}(x, y)$

Risk
Definitions
(cont'd)

- To get a handle on $\mathsf{P}(x, y)$, assume it's the same one that generated the training set

- Now use the training patterns to estimate $\mathsf{P}(x, y)$

D3.4 The empirical risk is

$$R_{\text{emp}}[f] := \int_{\mathcal{X}\times Y} c(x, y, f(x))\, p_{\text{emp}}(x, y)\, dx\, dy = \frac{1}{m}\sum_{i=1}^{m} c(x_i, y_i, f(x_i))$$

- Easy to compute and generally straightforward to minimize (depending on c)

- So now all we have to do is find an f that minimizes $R_{\text{emp}}[f]$, use that as our predictor, and we're done, right?

(Can we go home now?)

NO!

- We have to appropriately restrict the set of functions $\mathcal{F}$ from which we choose f
  - Otherwise, $R_{\text{emp}}[f]$ won't approximate $R[f]$, which is what we want to minimize

- E.g. what if $\mathcal{F}$ is the set of all functions from $\mathcal{X}$ to $\mathcal{Y}$?
  - Then our learning algorithm could get $R_{\text{emp}}[f] = 0$ by simply storing the (x, y) pairs in a table (i.e. memorization)
  - Is this learning? Will it generalize well?

- Restricting $\mathcal{F}$ has been looked from many perspectives: e.g. VC dimension, bias, structural risk minimization

- Our approach (called regularization) will quantify the "power" ("expressiveness") of each f and minimize a sum of this and $R_{\text{emp}}[f]$
  - Special case: minimum description length principle

Topic summary (over Lectures 2 and 3) due in 1 week!