Machine learning: lecture 9

Tommi S. Jaakkola
MIT AI Lab
tommi@ai.mit.edu

PS2 (9/20)	10/03
In class midterm	10/17
PS3 (10/4)	10/22
Project proposals	10/29
PS4 (10/22)	10/31
PS5 (10/31)	11/14
PS6 (11/14)	11/28
In class final	12/05
Projects	12/11

Topics

- Feature selection
  - information criterion
  - greedy selection
  - selection via regularization
- Combination of methods
  - forward fitting (regression)
  - boosting (classification)

Feature selection

- Various objectives
  - noise reduction
  - additional regularization
  - reduction of computational effort

  etc.

Feature selection example

- Our goal here is to reduce the number of useless word detectors/features

$$\begin{aligned} \phi_k(\mathbf{x}) &= 1 \text{ if word } k \text{ is present and } 0 \text{ otherwise} \\ y &= 0, 1 \quad \text{document label} \end{aligned}$$

- Suppose we have already estimated $P(\phi_k|y, \hat{\theta}_k)$ for each word k and label y. Here the *variable* ϕ_k would take the value $\phi_k(\mathbf{x})$ for any document $\mathbf{x}$.

  To simplify notation we define (unregularized estimates)

$$\begin{aligned} \hat{P}(y) & \quad \text{(estimated prior class freq.)} \\ \hat{P}(\phi_k, y) &= P(\phi_k|y, \hat{\theta}_k)\hat{P}(y) \\ \hat{P}(\phi_k) &= \sum_{y=0,1} \hat{P}(\phi_k, y) \quad \text{(estimated word freq.)} \end{aligned}$$

Feature selection example cont'd

- We can select features which alone would provide substantial amount of information about the label
- More formally, we can pick features that have a high value of *mutual information* with the labels:

$$I(\phi_k; y) = \sum_{\phi_k=0,1} \sum_{y=0,1} \hat{P}(\phi_k, y) \log_2 \left[\frac{\hat{P}(\phi_k, y)}{\hat{P}(\phi_k)\hat{P}(y)} \right]$$

This is a measure of distance between $\hat{P}(\phi_k, y)$ and $\hat{P}(\phi_k)\hat{P}(y)$, where

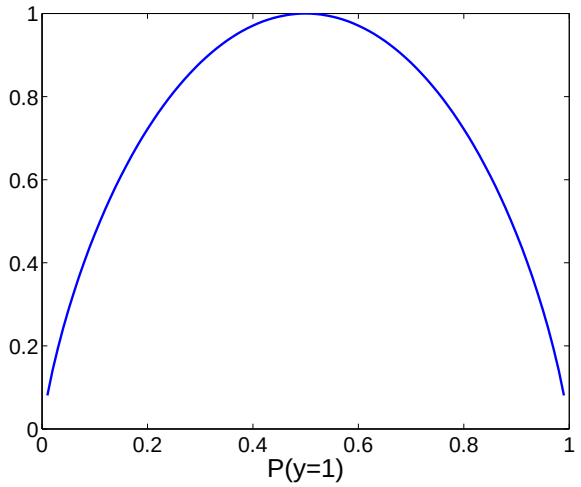
$\hat{P}(\phi_k, y)$ is our best estimate of the relation between the single feature and the label

$\hat{P}(\phi_k)\hat{P}(y)$ would be our estimate if we assumed a priori that features and labels are *independent*

A bit of background

- Entropy (uncertainty) of a binary random variable y

$$H(y) = -\sum_{y=0,1} P(y) \log_2 P(y)$$



Why Shannon entropy?

10101101010100011101101000110101 01...

Background cont'd

- Properties of mutual information:

$$I(\phi_k; y) = \sum_{\phi_k=0,1} \sum_{y=0,1} P(\phi_k, y) \log_2 \frac{P(\phi_k, y)}{P(\phi_k)P(y)}$$

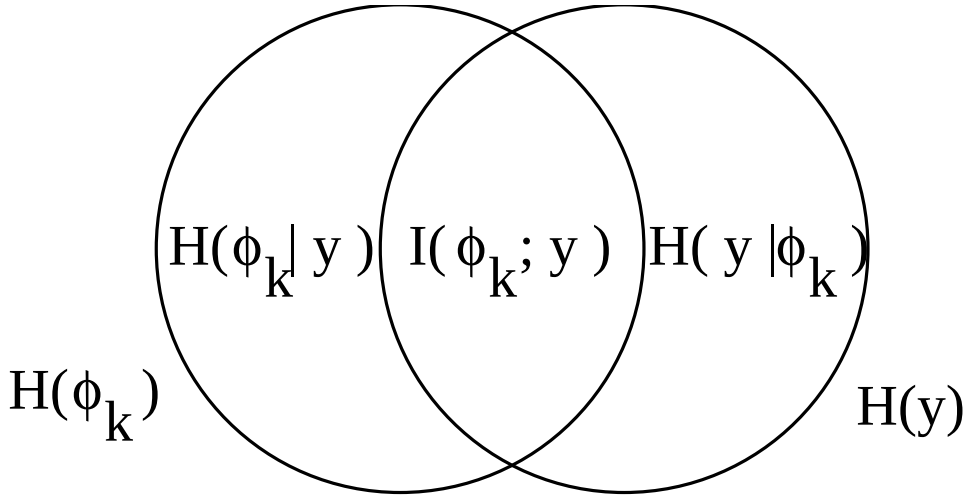
1. $I(\phi_k; y) = I(y; \phi_k)$ (symmetry)
2. If ϕ_k and y are independent, $I(\phi_k; y) = 0$
3. $I(\phi_k; y) \leq H(y)$, $I(\phi_k; y) \leq H(\phi_k)$
4. $I(\phi_k; y) = H(y) - H(y|\phi_k) = H(\phi_k) - H(\phi_k|y)$

where the conditional entropy $H(y|\phi_k)$ is defined as

$$H(y|\phi_k) = \sum_{\phi_k=0,1} P(\phi_k) \left[- \sum_{y=0,1} P(y|\phi_k) \log_2 P(y|\phi_k) \right]$$

Background cont'd

- Venn diagram



$$I(\phi_k; y) = H(y) - H(y|\phi_k) = H(\phi_k) - H(\phi_k|y)$$

Feature selection: the criterion

- We try to remove useless word detectors

$$\phi_k = 0,1 \quad \text{whether } k^{th} \text{ word is present in a document}$$

$$y = 0,1 \quad \text{document label}$$

- We select only features that have a high value of *mutual information* with the labels (i.e., knowing the value of the feature tells us a lot about what the label is):

$$I(\phi_k; y) = \sum_{\phi_k=0,1} \sum_{y=0,1} \hat{P}(\phi_k, y) \log_2 \left[\frac{\hat{P}(\phi_k, y)}{\hat{P}(\phi_k)\hat{P}(y)} \right]$$

  - how many features?
  - redundancy?
  - coordination?

Other ways of selecting features

- Let's try to solve the document classification task with a logistic regression model
  Our predictions based on m (initially 10,000) binary word features $\{\phi_1(\mathbf{x}), \ldots, \phi_m(\mathbf{x})\}$ would be

$$P(y = 1|\mathbf{x}, \mathbf{w}) = g(\, w_0 + w_1\phi_1(\mathbf{x}) + \ldots + w_m\phi_m(\mathbf{x})\,)$$

  where $g(\cdot)$ is the logistic function.

- We'd like to find a small subset of the features that lead to good (better) classification

- Alternatives (in addition to the one presented earlier):
  1. greedily add features
  2. find relevant features using regularization

Greedy selection of features

1. Find k for which

$$P(y = 1|\mathbf{x}, \mathbf{w}) = g(\, w_0 + w_k \phi_k(\mathbf{x}) \,)$$

   yields the best classifier
2. Find k' for which

$$P(y = 1|\mathbf{x}, \mathbf{w}) = g(\, w_0 + w_k \phi_k(\mathbf{x}) + w_{k'} \phi_{k'}(\mathbf{x}) \,)$$

   yields the best classifier. Here all the parameters w_0, w_k and $w_{k'}$ should be reoptimized when trying to add each k'
3. ...

- When/how do we stop?

Feature selection via regularization

$$P(y = 1|\mathbf{x}, \mathbf{w}) = g(\, w_0 + w_1\phi_1(\mathbf{x}) + \ldots + w_m\phi_m(\mathbf{x})\,)$$

- We can introduce a regularization penalty that tries to set the weights to zero unless they are "useful"

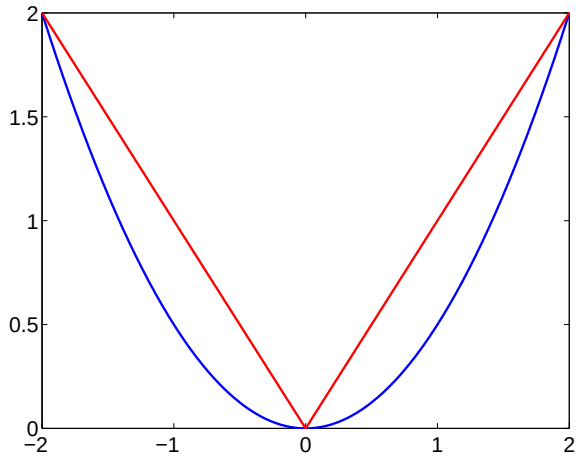
$$J(\mathbf{w}; C) = \sum_{t=1}^{n} \log P(y_t|\mathbf{x}_t, \mathbf{w}) - C \sum_{i=1}^{m} |w_i|$$

  where $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ is our training set. Note that w_0 is not penalized.
- The selection of non-zero weights here is carried out *jointly*, not individually

- Why should this regularization penalty work at all?

Feature selection via regularization cont'd

- The effect of the regularization penalty on feature selection depends on its derivative at $w \approx 0$



$w^2/2$ versus $|w|$

$$J(\mathbf{w}; C) = \sum_{t=1}^{n} \log P(y_t | \mathbf{x}_t, \mathbf{w}) - C \sum_{i=1}^{m} |w_i|$$

- How do we deal with redundant features?

Combination of methods

- Similarly to feature selection we can select simple "weak" classification or regression methods and combine them into a single "strong" method
- Example techniques
  - forward fitting (regression)
  - boosting (classification)

Combination of regression methods

- We want to combine multiple "weak" regression methods into a single "strong" method
- Suppose we are given a family simple regression methods

$$f(\mathbf{x};\theta) = w\,\phi_k(\mathbf{x})$$

  where $\theta = \{k, w\}$ (the parameters specify a single basis function as well as the associated weight)
- *Forward-fitting*: sequentially introduce new simple regression methods to reduce the remaining prediction error

Forward fitting cont'd

Simple family: $f(\mathbf{x};\theta) = w\phi_k(\mathbf{x}), \;\; \theta = \{k, w\}$

- We can fit each new component to reduce the prediction error; in each iteration we solve the same type of estimation problem

  Step 1: $$\hat{\theta}_1 \leftarrow \arg\min_{\theta} \sum_{i=1}^{n} (y_i - f(\mathbf{x}_i;\theta))^2$$

  Step 2: $$\hat{\theta}_2 \leftarrow \arg\min_{\theta} \sum_{i=1}^{n} (\underbrace{y_i - f(\mathbf{x}_i;\hat{\theta}_1)}_{\text{error}} - f(\mathbf{x}_i;\theta))^2$$

  Step 3: $\ldots$

- The resulting combined regression method

$$\hat{f}(\mathbf{x}) = f(\mathbf{x};\hat{\theta}_1) + \ldots + f(\mathbf{x};\hat{\theta}_m)$$

  has much lower (training) error.

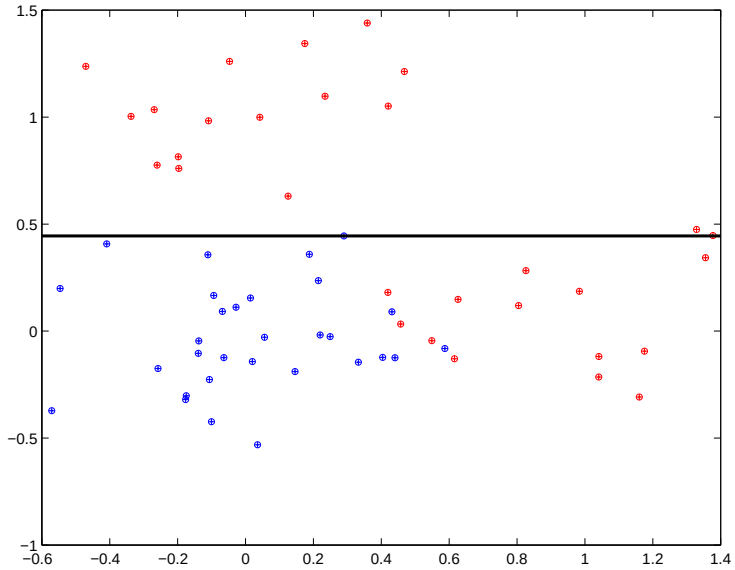
Combination of classifiers

- Suppose we have a family of component classifiers (generating ± 1 labels) such as *decision stumps*:

$$h(\mathbf{x};\theta) = \text{sign}(\, w_1\, x_k - w_0\,)$$

where $\theta = \{k, w_1, w_0\}$.

Each decision stump pays attention to only a single component of the input vector



Combination of classifiers con'd

- We'd like to combine simple classifiers in a manner similar to the regression models, i.e., construct the final classifier as the sign of

$$\hat{h}_m(\mathbf{x}) = \hat{\alpha}_1 \, h(\mathbf{x}; \hat{\theta}_1) + \ldots + \hat{\alpha}_m \, h(\mathbf{x}; \hat{\theta}_m)$$

  where the "votes" α are used to emphasize components that are more reliable than others

- Surely any new component classifier that we add should concentrate on the training examples that seem hard to classify.

  Part of the problem here is to estimate the new components (and votes) in a modular fashion