Lecture 2: Supervised Learning: Classification: KNN, Perceptron, Support Vector Machines, and Logistic Regression

Qinfeng (Javen) Shi

6 August 2015

Intro. to Stats. Machine Learning
COMP SCI 4401/7401

Table of Contents I

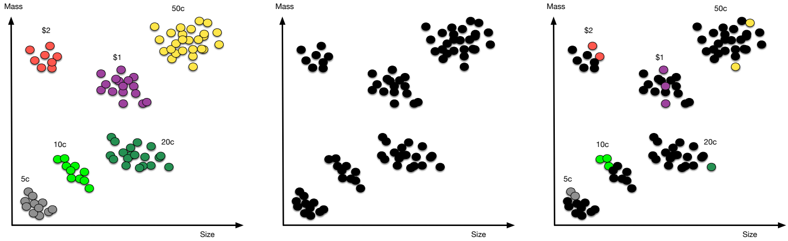
Table of Contents II

Recap

- What's machine learning?
- All you care is the testing error (not the training error).
- Train too well is not good (overfitting).
- The simplest model that fits the data is also the most plausible (Occam's Razor).

Recap continues

3 types of learning:



(a) Supervised (b) Unsupervised (c) Semi-supervised

Figure : Recognising coins by the features of their mass and size

Supervised Learning

We have (input, correct output) in the training data, *i.e.* Input-output data pairs $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$.

Based on the output y_i (not the input), Supervised Learning has 3 main types:

- Classification (discrete output)
- Novelty detection (discrete output)
- Regression (continuous output)

Classification

Discrete output y

1. Binary classification $y \in \{-1, 1\}$
2. Multi-class $y \in \{1, 2, \cdots, c\}$ for c classes
3. Multi-label $y = (y^{(1)}, \cdots, y^{(i)}, \cdots y^{(L)})$, where $y^{(i)} \in \{1, 2, \cdots, c_i\}$ assuming L labels and c_i classes for the i-th label.
4. Structured output. Complex objects with examples to show later.

Predict Annual Income (Binary classification)

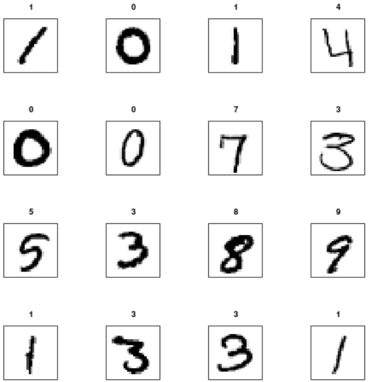
Predict whether income exceeds $50K/yr based on census data.
$y \in \{-1, 1\}$. 1 means $> 50K/yr$, -1 means $\leq 50K/yr$.

Input x from the UCI Adult Dataset

age: continuous.
workclass: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
fnlwgt: continuous.
education: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.
education-num: continuous.
marital-status: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.
occupation: Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.
relationship: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.
race: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.
sex: Female, Male.
capital-gain: continuous.
capital-loss: continuous.
hours-per-week: continuous.

Handwritten Digits Recognition (Multi-class)

$y \in \{0, 1, \cdots, 9\}$



Predict Articles' Topics (Multi-label)

$y =$ (religion, politics, science)

article	religion	politics	science
1	No	Yes	Yes
2	No	No	Yes
3	Yes	No	No
...			

Examples for structured outputs will be given in future lectures about probabilistic graphical models.

Novelty detection

Motivation: data from one class are easy to collect, and data from the rest class(es) are hard (or disastrous) to collect, or too few to be statistical meaningful.

Novelty detection

Motivation: data from one class are easy to collect, and data from the rest class(es) are hard (or disastrous) to collect, or too few to be statistical meaningful.

Example:

- Operational status of a nuclear plant as “normal”

Novelty detection

Motivation: data from one class are easy to collect, and data from the rest class(es) are hard (or disastrous) to collect, or too few to be statistical meaningful.

Example:

- Operational status of a nuclear plant as “normal”
- Seeing a baby elephant ⇒ elephants are small?

Novelty detection

- Only "normal data" in your training dataset (thus seen all as 1-class).

Novelty detection

- Only "normal data" in your training dataset (thus seen all as 1-class).
- for a testing data point, to predict if it's "normal" (*i.e.* belong to that class or not).

Novelty detection

Q: Since belonging to one class or not, why not a binary classification problem?

Novelty detection

Q: Since belonging to one class or not, why not a binary classification problem?

A: In novelty detection there are no "abnormal" data (*i.e.* 2nd class data) in the training dataset for you to train on.

Novelty detection

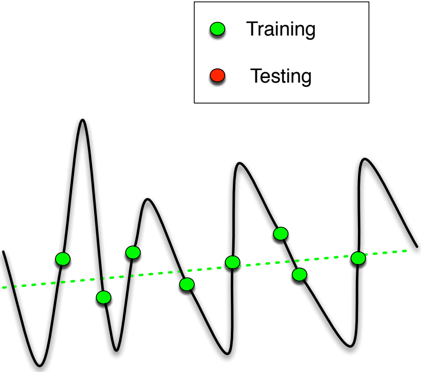
Q: Since belonging to one class or not, why not a binary classification problem?

A: In novelty detection there are no "abnormal" data (*i.e.* 2nd class data) in the training dataset for you to train on.

Other names: one-class classification, unary classification, outlier detection, anomaly detection

Regression

Continuous output y (to be covered in Lecture 4).



Gradient and Sub-Gradient

Refer to the video on the forum.

Illustrate using the whiteboard or the document camera if needed.

Convexity

- Convexity for a function
- Convexity for a set

Illustrate using the whiteboard or the document camera.

Lagrange multipliers and function

To solve a convex minimisation problem,

$$\begin{aligned} &\min_{\mathbf{x}} f_0(\mathbf{x}) \\ \text{s.t.} \quad & f_i(\mathbf{x}) \leq 0, i = 1, \cdots, m, \qquad \text{(Primal)} \end{aligned}$$

where f_0 is convex, and the feasible set (let's call it A) is convex (equivalent to all f_0, f_i are convex). $\mathbf{x}$ are called primal variables.

Lagrange function:

$$L(\mathbf{x}, \alpha) = f_0(\mathbf{x}) + \sum_{i=1}^{m} \alpha_i f_i(\mathbf{x}),$$

where $\alpha_i \geq 0$ are called Lagrange multipliers also known as (a.k.a) dual variables.

Dual problem

$L(\mathbf{x}, \alpha)$ produces the primal objective:

$$f_0(\mathbf{x}) = \max_{\alpha \geq 0} L(\mathbf{x}, \alpha).$$

$L(\mathbf{x}, \alpha)$ produces the dual objective:

$$D(\alpha) = \min_{\mathbf{x} \in A} L(\mathbf{x}, \alpha).$$

The following problem is called the (Lagrangian) dual problem,

$$\begin{aligned} &\max_{\alpha} D(\alpha) \\ \text{s.t.} \quad &\alpha_i \geq 0, i = 1, \cdots, m. \qquad \text{(Dual)} \end{aligned}$$

Primal and Dual relation

In general:

$$\min_{\mathbf{x}\in A} f_0(\mathbf{x}) = \min_{\mathbf{x}\in A}(\max_{\alpha\geq 0} L(\mathbf{x},\alpha)) \geq \max_{\alpha\geq 0}(\min_{\mathbf{x}\in A} L(\mathbf{x},\alpha)) = \max_{\alpha\geq 0} D(\alpha).$$

Since $L(\mathbf{x},\alpha)$ is convex w.r.t. $\mathbf{x}$, and concave w.r.t. α, we have

$$\min_{\mathbf{x}\in A} f_0(\mathbf{x}) = \min_{\mathbf{x}\in A}(\max_{\alpha\geq 0} L(\mathbf{x},\alpha)) = \max_{\alpha\geq 0}(\min_{\mathbf{x}\in A} L(\mathbf{x},\alpha)) = \max_{\alpha\geq 0} D(\alpha).$$

To solve the primal $\min_{\mathbf{x}\in A} f_0(\mathbf{x})$, one can solve the dual $\max_{\alpha\geq 0} D(\alpha)$.

Duality

The following always holds
$D(\alpha) \leq f_0(\mathbf{x}), \ \forall \, \mathbf{x}, \alpha$ (so called weak duality)

Sometimes (not always) below holds
$\max_\alpha D(\alpha) = \min_{\mathbf{x}} f_0(\mathbf{x})$ (so called strong duality)
Strong duality holds for SVM.

How to do it?

Given a problem, how to get its dual form?

1. transform the problem to a standard form
2. write down the Lagrange function
3. use optimality conditions to get equations
    - 1st order condition
    - complementarity conditions
4. remove the primal variables.

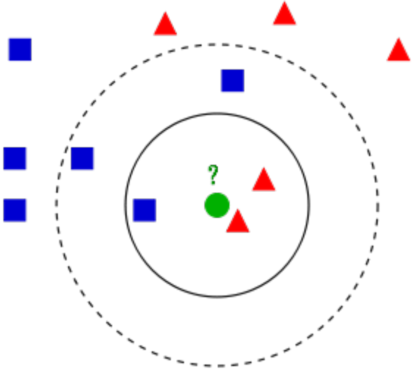
Examples.

Break

Take a break ...

1st glance at a classification algorithm

K Nearest Neighbour (KNN):



KNN: majority vote of the k Nearest Neighbours of the test point (green). If $k = 3$, the test point is predicted as red, if $k = 5$, the test point is predicted as blue. Picture courtesy of wikipedia

Questions

Thousands of classification algorithms out there. How can we possibly study they all?

Many algorithms come out every year, how do we keep up with them?

Answers

Learning theory analyses sets of algorithms' behaviour (will be covered in later lectures)

Many algorithms can be formulated in a unified framework called Empirical Risk Minimisation (ERM).

Risks

Given a loss $\ell(\mathbf{x}, y, \mathbf{w})$,

(True) Risk

$$R(\mathbf{w}, \ell) = \mathbb{E}_{(\mathbf{x},y)\sim p}\, \ell(\mathbf{x}, y, \mathbf{w})$$

Empirical Risk

$$R_n(\mathbf{w}, \ell) = \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{x}_i, y_i, \mathbf{w})$$

For example:

(SVM) Hinge loss $\ell_H(\mathbf{x}, y, \mathbf{w}) = \max\{0, 1 - y(\langle \mathbf{x}, \mathbf{w}\rangle)\}$.

Perceptron loss $\ell_{pern}(\mathbf{x}, y, \mathbf{w}) = \max\{0, -y \langle \mathbf{x}, \mathbf{w}\rangle\}$.

Zero-one loss $\ell_{0/1}(\mathbf{x}, y, \mathbf{w}) = \mathbf{1}_{\{g(\mathbf{x})\neq y\}}$. Here $\mathbf{1}_{\{a\}}$ is an indicator function which $= 1$ when a is true, $= 0$ otherwise.

Generalisation error

Generalisation error is the error rate over all possible testing data from the distribution P, that is the risk w.r.t. zero loss,

$$R(g) = \mathbb{E}_{(\mathbf{x},y)\sim p}[\mathbf{1}_{\{g(\mathbf{x})\neq y\}}] = P(g(\mathbf{x}) \neq y)$$

Empirical risk for zero-one loss is

$$R_n(g) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{\{g(\mathbf{x}_i)\neq y_i\}},$$

which is in fact the training error.

Regularised ERM

Regularised Empirical Risk Minimisation

$$g_n = \operatorname*{argmin}_{g \in \mathcal{G}} R_n(g) + \lambda \Omega(g),$$

where $\Omega(g)$ is the regulariser, *e.g.* $\Omega(g) = \|g\|^2$. $\mathcal{G}$ is the hypothesis set. Unfortunately, above is not convex. It turns out that one can optimise

$$\mathbf{w}_n = \operatorname*{argmin}_{\mathbf{w} \in \mathcal{W}} R_n(\mathbf{w}, \ell) + \lambda \Omega(\mathbf{w}),$$

as long as ℓ is a surrogate loss (brief def here) of the zero-one loss.

Decision functions (Recall from Lecture 1)

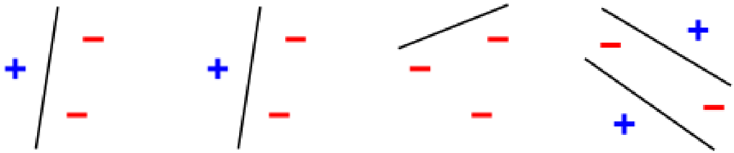
Linear decision function $g(\mathbf{x}; \mathbf{w}) = \text{sign}(\langle \mathbf{x}, \mathbf{w} \rangle + b)$ are often used (sign here is for binary classification). Here $\mathbf{x}, \mathbf{w} \in \mathbb{R}^d$, $b \in \mathbb{R}$. Since $\langle \mathbf{x}, \mathbf{w} \rangle + b = \langle [\mathbf{x}; 1], [\mathbf{w}; b] \rangle$, for simplicity one often write

$$\text{Binary} \quad g(\mathbf{x}; \mathbf{w}) = \text{sign}(\langle \mathbf{x}, \mathbf{w} \rangle).$$

$$\text{Multi-class} \quad g(\mathbf{x}; \mathbf{w}) = \operatorname*{argmax}_{y}(\langle \mathbf{x}, \mathbf{w}_y \rangle).$$

Separability

Not all data are linearly separable (*e.g.* the 4-th one).



Picture courtesy of wikipedia

To deal with linearly non-separable case, non-linear decision functions are needed (often used in kernel methods).

Perceptron Algorithm

Assume $g(\mathbf{x};\mathbf{w}) = \text{sign}(\langle \mathbf{x}, \mathbf{w} \rangle)$, where $\mathbf{x}, \mathbf{w} \in \mathbb{R}^d$, $y \in \{-1, 1\}$.

```
**Input:** training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, step size $\eta$, #iter $T$
Initialise $w_1 = \mathbf{0}$
for $t = 1$ to $T$ do
    $$\mathbf{w}_{t+1} = \mathbf{w}_t + \eta \sum_{i=1}^{n} (y_i \, \mathbf{x}_i \, \mathbf{1}_{\{y_i \langle \mathbf{x}_i, \mathbf{w}_t \rangle < 0\}}) \quad (1)$$
end for
**Output:** $\mathbf{w}^* = \mathbf{w}_T$
```

The class of $\mathbf{x}$ is predicted via

$$y^* = \text{sign}(\langle \mathbf{x}, \mathbf{w}^* \rangle)$$

View it in ERM

$$\min_{\mathbf{w},\xi} \frac{1}{n}\sum_{i=1}^{n} \xi_i, \quad \text{s.t.} \quad y_i \langle \mathbf{x}_i, \mathbf{w} \rangle \geq -\xi_i, \xi_i \geq 0$$

whose unconstrained form is

$$\min_{\mathbf{w}} \frac{1}{n}\sum_{i=1}^{n}[-y_i \langle \mathbf{x}_i, \mathbf{w} \rangle]_+ \Leftrightarrow \min_{\mathbf{w}} R_n(\mathbf{w}, \ell_{pern})$$

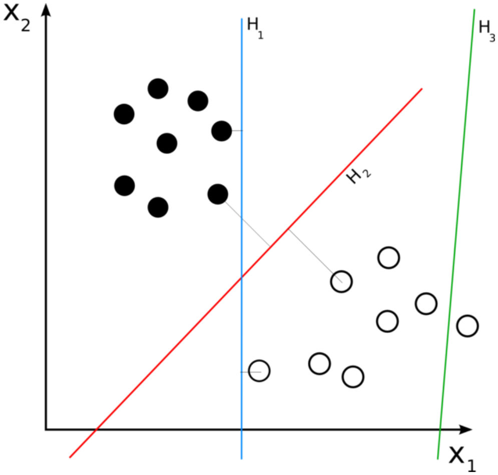
with Loss $\ell_{pern}(\mathbf{x}, y, \mathbf{w}) = \max\{0, -y \langle \mathbf{x}, \mathbf{w} \rangle\}$ and
Empirical Risk $R_n(\mathbf{w}, \ell_{pern}) = \frac{1}{n}\sum_{i=1}^{n} \ell_{pern}(\mathbf{x}_i, y_i, \mathbf{w})$.

$$\text{Sub-gradient} \quad \frac{\partial R_n(\mathbf{w}, \ell_{pern})}{\partial \mathbf{w}} = -\frac{1}{n}\sum_{i=1}^{n}(y_i \, \mathbf{x}_i \, \mathbf{1}_{\{y_i(\langle \mathbf{x}_i, \mathbf{w}_t \rangle) < 0\}}).$$

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta' \frac{\partial R_n(\mathbf{w}, \ell_{pern})}{\partial \mathbf{w}} = \mathbf{w}_t + \eta' \frac{1}{n}\sum_{i=1}^{n}(y_i \, \mathbf{x}_i \, \mathbf{1}_{\{y_i(\langle \mathbf{x}_i, \mathbf{w}_t \rangle) < 0\}})$$

Letting $\eta = \eta' \frac{1}{n}$ recovers the equation (1).

Max Margin



Picture courtesy of wikipedia

Max Margin Formulation

One form of soft margin binary Support Vector Machines (SVMs) (a primal form) is

$$\min_{\mathbf{w},b,\gamma,\xi} -\gamma + C\sum_{i=1}^{n}\xi_i \tag{2}$$
$$\text{s.t. } y_i(\langle \mathbf{x}_i, \mathbf{w}\rangle + b) \geq \gamma - \xi_i, \xi_i \geq 0, \| \mathbf{w} \|^2 = 1$$

For a testing $\mathbf{x}'$, given the learnt $\mathbf{w}^*$, b^*, the predicted label

$$y^* = g(\mathbf{x}'; \mathbf{w}^*) = \text{sign}(\langle \mathbf{x}', \mathbf{w}^*\rangle + b^*).$$

Primal

A more popular version is (still a primal form)

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\| \mathbf{w} \|^2 + C\sum_{i=1}^{n} \xi_i,$$
$$\text{s.t.} \quad y_i(\langle \mathbf{x}_i, \mathbf{w}\rangle + b) \geq 1 - \xi_i, \xi_i \geq 0, i = 1, \cdots, n,$$

This is equivalent to the previous form and $\gamma = 1/\| \mathbf{w} \|$.

View in in ERM hinge loss $\ell_H(\mathbf{x}, y, \mathbf{w}) = \max\{0, 1 - y(\langle \mathbf{x}, \mathbf{w}\rangle + b)\}$, and $\Omega(\mathbf{w}) = \frac{1}{2}\| \mathbf{w} \|^2$ with a proper λ.

It is often solved by using Lagrange multipliers and duality.

Lagrangian function

$$\begin{aligned} L(\mathbf{w}, b, \xi, \alpha, \beta) = {} & \frac{1}{2} \| \mathbf{w} \|^2 + C \sum_{i=1}^{n} \xi_i \\ & + \sum_{i=1}^{n} \alpha_i [1 - \xi_i - y_i(\langle \mathbf{x}_i, \mathbf{w} \rangle + b)] + \sum_{i=1}^{n} \beta_i(-\xi_i) \end{aligned}$$

Optimise Lagrangian function — 1st order condition

To get $\inf_{\mathbf{w},b,\xi}\{L(\mathbf{w},b,\xi,\alpha,\beta)\}$, by 1st order condition

$$\frac{\partial L(\mathbf{w},b,\xi,\alpha,\beta)}{\partial\,\mathbf{w}} = 0 \Rightarrow \mathbf{w}^* - \sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i = 0 \tag{3}$$

$$\frac{\partial L(\mathbf{w},b,\xi,\alpha,\beta)}{\partial \xi_i} = 0 \Rightarrow C - \alpha_i - \beta_i = 0 \tag{4}$$

$$\frac{\partial L(\mathbf{w},b,\xi,\alpha,\beta)}{\partial b} = 0 \Rightarrow \sum_{i=1}^{n}\alpha_i y_i = 0 \tag{5}$$

Optimise Lagrangian function — Complementarity conditions

Complementarity conditions

$$\alpha_i[1 - \xi_i - y_i(\langle \mathbf{x}_i, \mathbf{w} \rangle + b)] = 0, \forall i \tag{6}$$

$$\beta_i \xi_i = 0, \forall i \tag{7}$$

Dual

$$
\begin{aligned}
& L(\mathbf{w}^*, b^*, \xi^*, \alpha, \beta) \\
&= \frac{1}{2}\langle \mathbf{w}^*, \mathbf{w}^* \rangle + \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \alpha_i y_i \langle \mathbf{x}_i, \mathbf{w}^* \rangle \\
&+ \sum_{i=1}^{n} \xi_i^* (C - \alpha_i - \beta_i) + b(\sum_{i=1}^{n} \alpha_i y_i) \\
&= \frac{1}{2}\langle \mathbf{w}^*, \mathbf{w}^* \rangle + \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \alpha_i y_i \langle \mathbf{x}_i, \mathbf{w}^* \rangle \quad \text{via eq(4) and eq(5)} \\
&= \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle + \sum_{i=1}^{n} \alpha_i - \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \;\; \text{via eq(3)} \\
&= \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle
\end{aligned}
$$

Dual

$\max_\alpha \inf_{\mathbf{w},b,\xi}\{L(\mathbf{w},b,\xi,\alpha,\beta)\}$ gives the dual form:

$$\begin{aligned}\max_{\alpha} &\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j\rangle \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq C, i = 1, \cdots, n, \quad \text{(via eq(4))} \\ & \sum_{i=1}^{n}\alpha_i y_i = 0\end{aligned}$$

Let α^* be the solution.

From dual to primal variables

How to compute $\mathbf{w}^*, b^*$ from α^*?
Via eq(3), we have

$$\mathbf{w}^* = \sum_{i=1}^{n} \alpha_i^* y_i \mathbf{x}_i. \tag{8}$$

Via comp condition eq(6), we have $\alpha_i[1 - \xi_i - y_i(\langle \mathbf{x}_i, \mathbf{w} \rangle + b)] = 0, \forall i$.
When $\alpha_i > 0$, we know $1 - \xi_i - y_i(\langle \mathbf{x}_i, \mathbf{w} \rangle + b) = 0$. It will be great if $\xi_i = 0$ too. When will it happen? $\beta_i > 0 \Rightarrow \xi_i$ because of comp condition eq(7). Since $C - \alpha_i - \beta_i = 0$ (4), $\beta_i > 0$ means $\alpha < C$.
For any i, s.t. $0 < \alpha_i < C$, $1 - y_i(\langle \mathbf{x}_i, \mathbf{w} \rangle + b) = 0$, so (multiple y_i on both sides, and the fact that $y_i^2 = 1$)

$$b^* = y_i - \langle \mathbf{x}_i, \mathbf{w}^* \rangle \tag{9}$$

Numerically wiser to take the average over all such training points (Burges tutorial).

Support Vectors

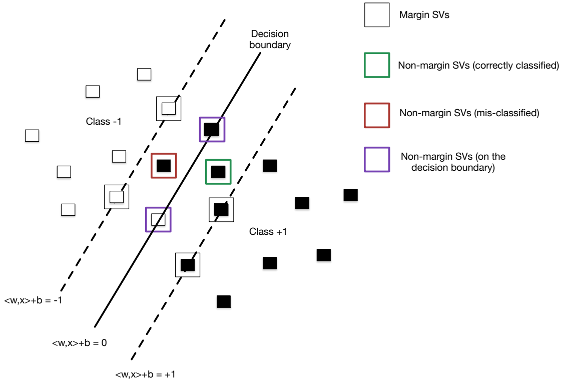
$y^* = \mathrm{sign}(\langle x, \mathbf{w}^* \rangle + b^*) = \mathrm{sign}(\sum_{i=1}^n \alpha_i^* y_i \langle \mathbf{x}_i, x \rangle + b^*)$.

It turns out many $\alpha_i^* = 0$. Those $\mathbf{x}_j$ with $\alpha_j^* > 0$ are called support vectors. Let $S = \{j : \alpha_j^* > 0\}$

$$y^* = \mathrm{sign}(\sum_{j \in S} \alpha_j^* y_j \langle \mathbf{x}_j, \mathbf{x} \rangle + b^*)$$

Note now y can be predicted without explicitly expressing $\mathbf{w}$ as long as the support vectors are stored.

Support Vectors


Decision boundary
Class -1
Class +1
<w,x>+b = -1
<w,x>+b = 0
<w,x>+b = +1
Margin SVs
Non-margin SVs (correctly classified)
Non-margin SVs (mis-classified)
Non-margin SVs (on the decision boundary)

Two types of SVs:

- Margin SVs: $0 < \alpha_i < C$ ($\xi_i = 0$, on the dash lines)
- Non-margin SVs: $\alpha_i = C$ ($\xi_i > 0$, thus violating the margin. More specifically, when $1 > \xi_i > 0$, correctly classified; when $\xi_i > 1$, it's mis-classified; when $\xi_i = 1$, on the decision boundary)

Dual

All derivation holds if one replaces $\mathbf{x}_j$ with $\phi(\mathbf{x}_j)$ and let kernel function $\kappa(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle$. This gives

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \kappa(\mathbf{x}_i, \mathbf{x}_j)$$

$$\text{s.t.} \quad 0 \leq \alpha_i \leq C, i = 1, \cdots, n$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

$$y^* = \text{sign}[\sum_{j \in S} \alpha_j^* y_j \kappa(\mathbf{x}_j, \mathbf{x}) + b^*].$$

This leads to non-linear SVM and more generally kernel methods (will be covered in later lectures).

Theoretical justification

An example of generalisation bounds is below (just to give you an intuition, no need to fully understand it for now).

Theorem (VC bound)

Denote h *as the VC dimension, for all* $n \geq h$*, for any* $\delta \in (0, 1)$*, with probability at least* $1 - \delta$*,* $\forall g \in \mathcal{G}$

$$R(g) \leq R_n(g) + 2\sqrt{2\frac{h \log \frac{2en}{h} + \log(\frac{2}{\delta})}{n}}.$$

Margin $\gamma = 1/\| \mathbf{w} \|$, $h \leq \min\{D, \lceil \frac{4R^2}{\gamma^2} \rceil\}$, where the radius $R^2 = \max_{i=1}^{n} \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_i) \rangle$ (assuming data are already centered)

Theoretical justification

Other tighter bounds such as Rademacher bounds, PAC-Bayes bounds *etc.* (Generalisation bounds will be covered in the final lecture).

Logistic Regression for Binary Classification

For binary LR, one can assume

$$P(y=+1|\,\mathbf{x};\mathbf{w}) = \frac{1}{1+e^{-\langle \mathbf{w},\mathbf{x}\rangle}}$$

Thus

$$\begin{aligned} P(y=-1|\,\mathbf{x};\mathbf{w}) &= 1 - P(y=+1|\,\mathbf{x};\mathbf{w}) \\ &= \frac{e^{-\langle \mathbf{w},\mathbf{x}\rangle}}{1+e^{-\langle \mathbf{w},\mathbf{x}\rangle}} = \frac{1}{1+e^{\langle \mathbf{w},\mathbf{x}\rangle}} \end{aligned}$$

Above means

$$P(y|\,\mathbf{x};\mathbf{w}) = \frac{1}{1+e^{-y\langle \mathbf{w},\mathbf{x}\rangle}} \tag{10}$$

Alternative formulation

Alternatively if let $y \in \{0, 1\}$, one assumes

$$P(y = +1 | \mathbf{x}; \mathbf{w}) = \frac{1}{1 + e^{-\langle \mathbf{w}, \mathbf{x} \rangle}}$$
$$P(y = 0 | \mathbf{x}; \mathbf{w}) = \frac{1}{1 + e^{\langle \mathbf{w}, \mathbf{x} \rangle}},$$

which means

$$P(y | \mathbf{x}; \mathbf{w}) = (\frac{1}{1 + e^{-\langle \mathbf{w}, \mathbf{x} \rangle}})^{y} (\frac{1}{1 + e^{\langle \mathbf{w}, \mathbf{x} \rangle}})^{(1-y)} \tag{11}$$

Because eq(11) is not as neat as eq(10), we will use eq(10) with $y \in \{-1, 1\}$.

Maximum Likelihood and Log loss

Maximum Likelihood

$$
\begin{aligned}
&\operatorname*{argmax}_{\mathbf{w}} \prod_{i=1}^{n} P(y_i | \mathbf{x}_i; \mathbf{w}) \\
&= \operatorname*{argmin}_{\mathbf{w}} -\log \left(\prod_{i=1}^{n} P(y_i | \mathbf{x}_i; \mathbf{w}) \right) \\
&= \operatorname*{argmin}_{\mathbf{w}} -\sum_{i=1}^{n} \log P(y_i | \mathbf{x}_i; \mathbf{w}) \quad \text{(log loss in ERM)}
\end{aligned}
$$

Gradient for binary class

Let

$$L(\mathbf{w}|X,Y) = -\sum_{i=1}^{n} \log P(y_i|\,\mathbf{x}_i;\mathbf{w})$$

$$\begin{aligned}\frac{\partial L(\mathbf{w}|X,Y)}{\partial\,\mathbf{w}} &= \frac{\partial \sum_{i=1}^{n} \log\left(1+e^{-y_i\langle \mathbf{w},\mathbf{x}_i\rangle}\right)}{\partial\,\mathbf{w}} \quad \text{via eq(10)}\\ &= \sum_{i=1}^{n} \frac{e^{-y_i\langle \mathbf{w},\mathbf{x}_i\rangle}}{1+e^{-y_i\langle \mathbf{w},\mathbf{x}_i\rangle}}(-y_i\,\mathbf{x}_i)\\ &= \sum_{i=1}^{n}(-y_i\,\mathbf{x}_i)(1-P(y_i|\,\mathbf{x}_i;\mathbf{w}))\end{aligned}$$

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta\frac{\partial L(\mathbf{w}|X,Y)}{\partial\,\mathbf{w}}$$

Multi-class

For multi-class LR, let c be the number of classes. Let $\mathbf{w} = (\mathbf{w}_{y'})_{y' \in \mathcal{Y}}$, where $\mathbf{w}_{y'} \in \mathbb{R}^d$, thus $\mathbf{w} \in \mathbb{R}^{dc}$. One assumes

$$P(y|\,\mathbf{x};\mathbf{w}) = \frac{e^{\left\langle \mathbf{w}_y, \mathbf{x} \right\rangle}}{\sum_{y' \in \mathcal{Y}} e^{\left\langle \mathbf{w}_{y'}, \mathbf{x} \right\rangle}} \tag{12}$$

Note: the multi-class form can recover the binary form despite different appearance.

$$\begin{aligned} L(\mathbf{w}\,|X, Y) &= -\sum_{i=1}^{n} \log P(y_i|\,\mathbf{x}_i;\mathbf{w}) \\ &= \sum_{i=1}^{n} \log(\sum_{y' \in \mathcal{Y}} e^{\left\langle \mathbf{w}_{y'}, \mathbf{x}_i \right\rangle}) - \left\langle \mathbf{w}_y, \mathbf{x}_i \right\rangle \end{aligned}$$

Gradient for Multi-class

$$\begin{aligned}\frac{\partial L(\mathbf{w}\,|X,Y)}{\partial\,\mathbf{w}_y} &= \sum_{i=1}^{n}\Big(\frac{e^{\langle \mathbf{w}_y,\mathbf{x}_i\rangle}}{\sum_{y'\in\mathcal{Y}} e^{\left\langle \mathbf{w}_{y'},\mathbf{x}_i\right\rangle}}\,\mathbf{x}_i - \mathbf{x}_i\Big)\\ &= \sum_{i=1}^{n}\mathbf{x}_i(P(y_i|\,\mathbf{x}_i;\mathbf{w}) - 1)\end{aligned}$$

$$\frac{\partial L(\mathbf{w}\,|X,Y)}{\partial\,\mathbf{w}} = \Big(\frac{\partial L(\mathbf{w}\,|X,Y)}{\partial\,\mathbf{w}_y}\Big)_{y\in\mathcal{Y}}$$

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta\frac{\partial L(\mathbf{w}\,|X,Y)}{\partial\,\mathbf{w}}$$

Supervised Learning Definition Revisit

Definition (Lecture 1)

Given input-output data pairs $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ sampled from an unknown but fixed distribution $p(\mathbf{x}, y)$, the goal is to learn $g : \mathcal{X} \to \mathcal{Y}$, $g \in \mathcal{G}$ s.t. $p(g(\mathbf{x}) \neq y)$ is small.

$p(g(\mathbf{x}) \neq y)$ is mainly for classification, not very suitable for regression.

Supervised Learning definition revisit

A more general definition for SL (than the previous lecture):

Definition (revisit)

Given input-output data pairs $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ sampled from an unknown but fixed distribution $p(\mathbf{x}, y)$, the goal is to learn $g : \mathcal{X} \to \mathcal{Y}$, $g \in \mathcal{G}$ s.t. ~~$p(g(\mathbf{x}) \neq y)$~~ the risk $\mathbb{E}_{(\mathbf{x},y)\sim p}[\ell(g(\mathbf{x}), y)]$ is small w.r.t. certain loss ℓ.

$\mathbb{E}_{(\mathbf{x},y)\sim p}[\ell(g(\mathbf{x}), y)]$ is more general (applicable to classification, and regression).

Extension to structured output

SVMs are extended to what known as Structured SVM.

Logistic Regression is extended to what later known as Conditional Random Fields.

Structured case will be covered in later lectures.

That's all

Thanks!