THE UNIVERSITY OF SYDNEY

Advanced Machine Learning

(COMP 5328)

Review

Tongliang Liu

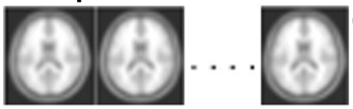
Elements of Machine Learning Algorithms

THE UNIVERSITY OF SYDNEY

Input training data

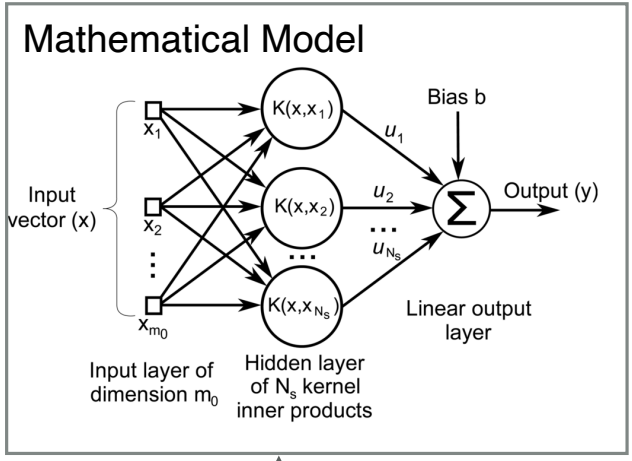
Group 1

Group 2

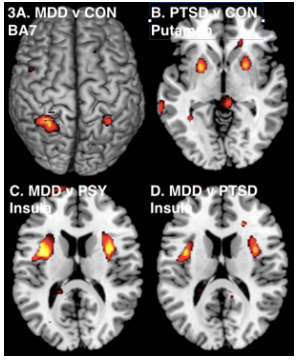


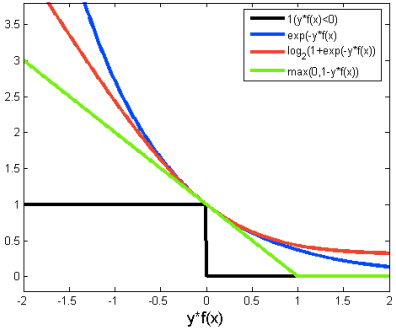
Data

Input predefined function class



Output hypothesis
Predictions/Patterns





Objective function

Optimisation method

Topics

THE UNIVERSITY OF SYDNEY

Week	Topic	Elements of ML
1	Introduction to ML Problems	
2	Loss Functions and Convex Optimisation	I, III, IV
3	Hypothesis Complexity and Generalisation	II, V
4	Dictionary Learning and NMF	I, II, IV, V
5	Sparse Coding and Regularisation	II, III, V
6	Learning with Noisy Data	I, III
7	Domain Adaptation and Transfer Learning	I, II
8	Learning with Noisy Data II: Label Noise	I, III, V
9	Reinforcement Learning	I, III, IV, V
10	Causal Inference	I, III
11	Multi-task Learning	III, V
12	Review	

I. Input training data
II. Predefined hypothesis class
III. Objective function
IV. Optimisation method
V. Output hypothesis

Best classifier (Input data)

THE UNIVERSITY OF SYDNEY

- For a given data point (*X*,*Y*), the classification error for a hypothesis h is measured by the 0-1 loss function:

$$1_{\{Y \neq \text{sign}(h(X))\}} = \begin{cases} 0 & Y = \text{sign}(h(X)) \\ 1 & Y \neq \text{sign}(h(X)) \end{cases}$$

- The best classifier can be mathematically defined as:

$$\arg\min_{h} \frac{1}{|D|} \sum_{i \in D} 1_{\{Y_i \neq \text{sign}(h(X_i))\}}$$

where D is the set of indices of **all possible data** points of the task, and $|D|$ denotes the size of the set D.

The law of large numbers

THE UNIVERSITY OF SYDNEY

LLN describes the result of performing the same experiment a large number of times.

The average of the results obtained from a large number of independent trials should converge to the expected value.

$$\frac{1}{|D|}\sum_{i\in D} 1_{\{Y_i \neq \text{sign}(h(X_i))\}} \xrightarrow{|D|\to\infty} \mathbb{E}\left[1_{\{Y \neq \text{sign}(h(X))\}}\right]$$

Best classifier

THE UNIVERSITY OF SYDNEY

- The best classifier can be mathematically defined as:

$$\arg\min_{h} \mathbb{E}[1_{\{Y \neq \text{sign}(h(X))\}}]$$

- Some problems: 1, the distribution of data is unknown. We cannot calculate the expectation. 2, the objective function is not convex or smooth, hard to optimise. 3, what kind of hypothesis *h* should we employ to fit the data?

Objective function

- Given a classification task, we want to find a classifier such that the following is minimised:

$$\mathbb{E}[1_{\{Y \neq \text{sign}(h(X))\}}]$$

- We don't have the distribution of data. Fortunately, we have some examples (or a training sample) draw from the distribution:

$$S = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$$

- Because of the law of large numbers, we can use

$$\frac{1}{n} \sum_{i=1}^{n} 1_{\{Y \neq \text{sign}(h(X))\}}$$

(unbiased estimator)

to estimate $\mathbb{E}[1_{\{Y \neq \text{sign}(h(X))\}}]$

Objective function

- The estimator is unbiased because

$$\frac{1}{n}\sum_{i=1}^{n} 1_{\{Y_i \neq \text{sign}(h(X_i))\}} \xrightarrow{n\to\infty} \mathbb{E}[1_{\{Y \neq \text{sign}(h(X))\}}]$$

- This also explains why big data is very helpful.

Optimisation method

- How to obtain the hypothesis that minimises the objective function, i.e.,

$$\arg\min_{h} \frac{1}{n} \sum_{i=1}^{n} 1_{\{Y_i \neq \text{sign}(h(X_i))\}}$$

- Pick one from the predefined hypothesis class *H* to minimise the objective, i.e.,

$$\arg\min_{h \in H} \frac{1}{n} \sum_{i=1}^{n} 1_{\{Y_i \neq \text{sign}(h(X_i))\}}$$

Convex optimisation

- Pick one from the predefined hypothesis class *H* to minimise the objective, i.e.,

$$\arg\min_{h \in H} \frac{1}{n} \sum_{i=1}^{n} \ell(X_i, Y_i, h)$$

where the loss function ℓ is a convex surrogate for the 0-1 loss function.

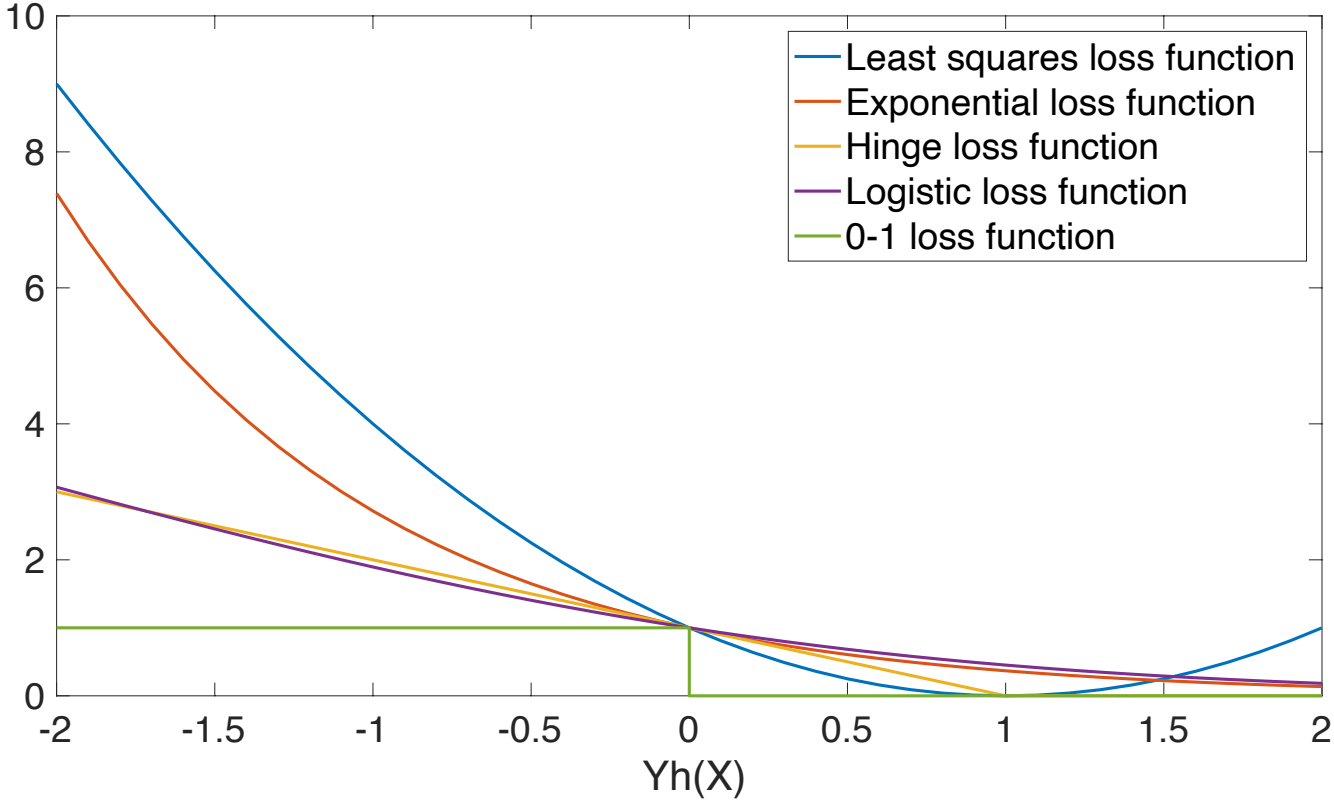
Surrogate loss functions

THE UNIVERSITY OF SYDNEY

- Popular surrogate loss functions:
- Hinge loss: $\ell(X, Y, h) = \max\{0, 1 - Yh(X)\}$
- Logistic loss: $\ell(X, Y, h) = \log_2(1 + \exp(-Yh(X)))$
- Least squares loss: $\ell(X, Y, h) = (Y - h(X))^2$
- Exponential loss: $\ell(X, Y, h) = \exp(-Yh(X))$

Surrogate loss functions

THE UNIVERSITY OF SYDNEY



Surrogate loss functions

THE UNIVERSITY OF SYDNEY

- What are the differences between the 0-1 loss function and the surrogate loss functions?
- Classification-calibrated surrogate loss functions: which will result in the same classifier as the 0-1 loss function if the training data is sufficiently large (an asymptotical property).
- Most (but not all) of the popularly used surrogate loss functions are all classification-calibrated surrogate loss functions.

Bartlett, Peter L., Michael I. Jordan, and Jon D. McAuliffe. "Convexity, classification, and risk bounds." Journal of the American Statistical Association 101.473 (2006): 138-156.

Zhang, Jingwei, Tongliang Liu, and Dacheng Tao. "On the Rates of Convergence from Surrogate Risk Minimizers to the Bayes Optimal Classifier." arXiv preprint arXiv:1802.03688 (2018).

Surrogate loss functions

THE UNIVERSITY OF SYDNEY

- How to check if a given surrogate loss function is a classification-calibrated surrogate loss functions?

Let $\phi(Yh(X)) = \ell(X, Y, h)$.

Given ϕ is convex, the loss function is classification-calibrated
if and only if ϕ is differentiable at 0, and

$$\phi'(0) < 0.$$

Bartlett, Peter L., Michael I. Jordan, and Jon D. McAuliffe. "Convexity, classification, and risk bounds." Journal of the American Statistical Association 101.473 (2006): 138-156.

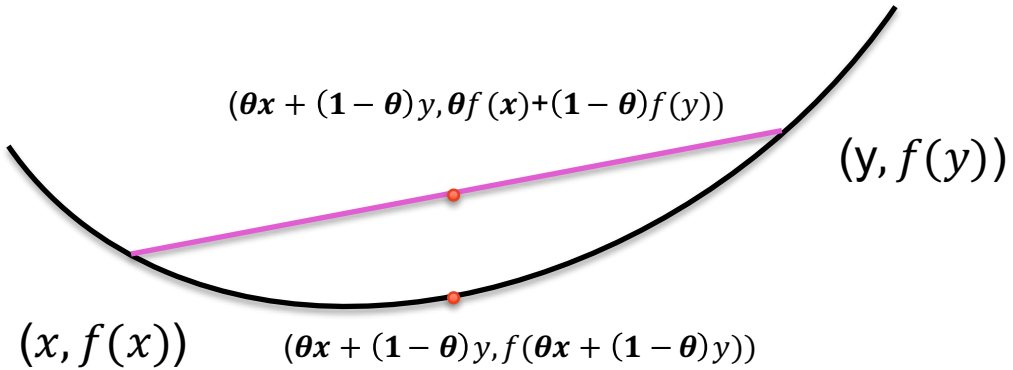
Zhang, Jingwei, Tongliang Liu, and Dacheng Tao. "On the Rates of Convergence from Surrogate Risk Minimizers to the Bayes Optimal Classifier." arXiv preprint arXiv:1802.03688 (2018).

Convex functions

A function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if its domain ($\mathrm{dom}\ f$) is a convex open set and

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta) f(y)$$

for all $x, y \in \mathrm{dom}\ f$, and $0 \leq \theta \leq 1$.



Convex functions

Function f is twice differentiable if the Hessian matrix

$$H_{ij} = \frac{\partial f(x)}{\partial x_i \partial x_j}, \forall x \in \text{domain } f \subseteq \mathbb{R}^d$$

exists.

We now assume that f is twice differentiable, that is, its Hessian matrix exists at each point in the domain of f, which is open. Then f is convex if and only if the Hessian matrix is positive semidefinite for all point in the domain.

Taylor's Theorem

Let $k \geq 1$ be an integer and let the function $f : \mathbb{R} \to \mathbb{R}$ be k times differentiable at the point $a \in \mathbb{R}$. Then there exists a function $h_k : \mathbb{R} \to \mathbb{R}$ such that

$$f(x) = f(a) + f'(a)(x-a) + ... \\ + \frac{f^{(k)}(a)}{k!}(x-a)^k + h_k(x)(x-a)^k$$

and $\lim_{x \to a} h_k(x) = 0$. This means that

$$h_k(x)(x-a)^k = o((x-a)^k).$$

Gradient descent method

Let $$f(h) = \frac{1}{n}\sum_{i=1}^{n} \ell(X_i, Y_i, h)$$

$$h_{k+1} = h_k + \eta d_k \,.$$

By Taylor's theorem, we have

$$f(h_{k+1}) = f(h_k) + \eta \nabla f(h_k)^\top d_k + o(\eta)\,.$$

For positive but sufficiently small η,
$f(h_{k+1})$ is smaller than $f(h_k)$,
if the direction d_k is chosen so that

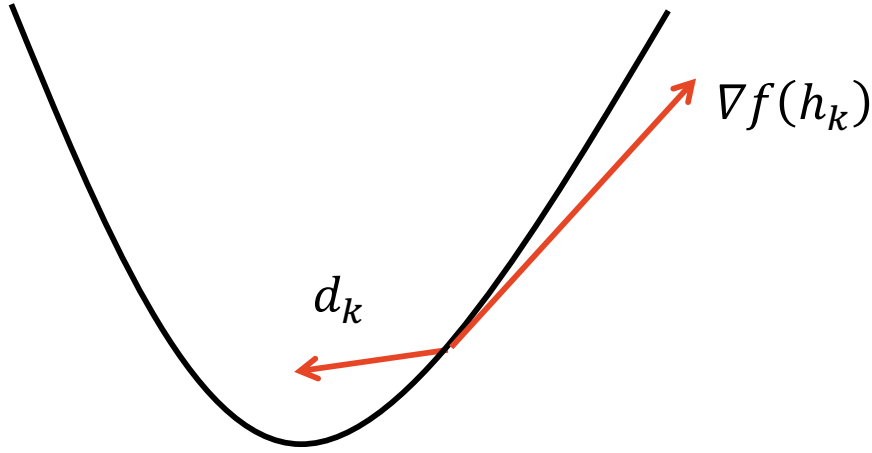
$$\nabla f(h_k)^\top d_k < 0 \quad \text{when} \quad \nabla f(h_k) \neq 0.$$

An iterative updating method

$$f(h_{k+1}) = f(h_k) + \eta \nabla f(h_k)^\top d_k + o(\eta).$$

Two problems:

- How to find d_k?
- How to choose η?



Gradient convergence rate

How many iteration steps do we need to achieve the optimal solution ?

$$h_S = \arg\min_{h \in H} f(h) = \arg\min_{h \in H} \frac{1}{n} \sum_{i=1}^{n} \ell(X_i, Y_i, h)$$

Gradient convergence rate

If the objective function is strongly-convex, and has Lipschitz Gradient, we have a **linear convergence rate**:

$$f(h_{k+1}) - f(h_S) \leq \left(1 - \frac{\mu}{L}\right)^k (f(h_1) - f(h_S)).$$

Gradient descent method

Algorithm	Assumption	Convergence rate
Gradient	Lipshitz Gradient, Convex	$O(1/k)$
Gradient	Lipshitz Gradient, Strongly-Convex	$O(1-\mu/L)^k$
Newton	Lipshitz Gradient, Strongly-convex	$\prod_{i=1}^{k}\rho_k$, $\rho_k \to 0$

Notation

- The best function in the universal function space (target concept):

$$c = \arg\min_{h} R(h).$$

- The best function in the predefined hypothesis class:

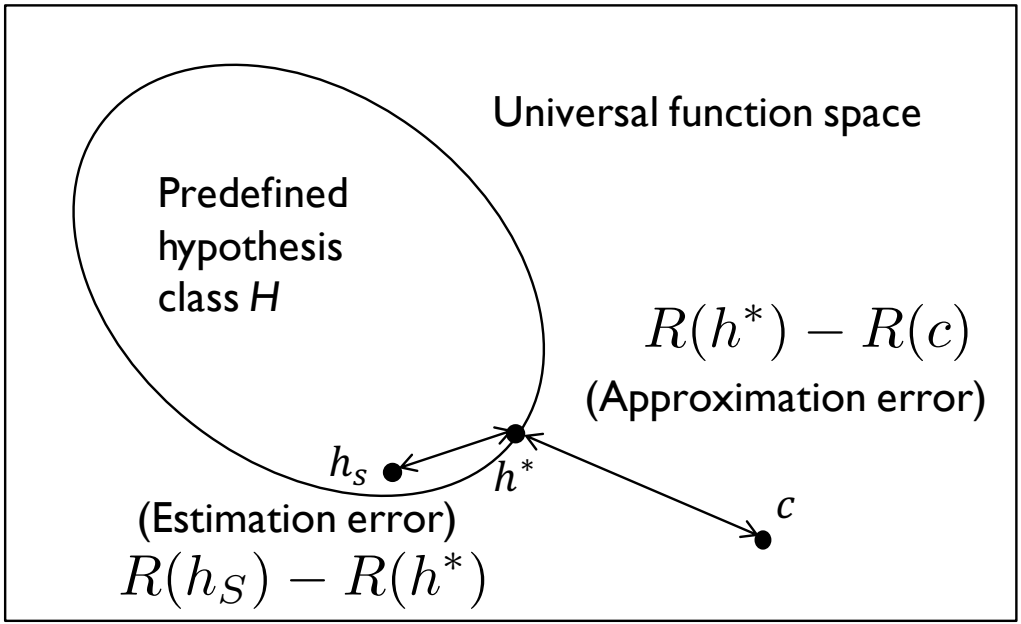
$$h^* = \arg\min_{h \in H} R(h).$$

- The hypothesis we can learn from data:

$$h_S = \arg\min_{h \in H} R_S(h) = \arg\min_{h \in H} \frac{1}{n} \sum_{i=1}^{n} \ell(X_i, Y_i, h).$$

Approximation and estimation error

What are the differences between c, h^*, and h_S?



Niyogi, Partha, and Federico Girosi. "Generalization bounds for function approximation from scattered noisy data." Advances in Computational Mathematics 10.1 (1999): 51-80.

PAC learning framework

Definition:

A hypothesis class H is said to be PAC (probably approximately correct)-learnable if there exists an learning algorithm $\mathcal{A}$ and a polynomial function $poly(\cdot, \cdot)$ such that for any $\epsilon > 0$ and $\delta > 0$, for all distribution D on $X \times Y$, the following holds for any sample of size $n > poly(1/\delta, 1/\epsilon)$ and the hypothesis h_S learned by $\mathcal{A}$:

$$p\left\{R(h_S) - \min_{h \in H} R(h) \leq \epsilon\right\} \geq 1 - \delta.$$

PAC learning framework

learned hypothesis approximately probably

$$p\left\{R(h_S) - \min_{h \in H} R(h) \leq \epsilon\right\} \geq 1 - \delta.$$

If the training sample size is large enough, e.g., $n > poly(1/\delta, 1/\epsilon)$ with a high probability, the learned hypothesis h_S can be an approximation of the best one in the predefined hypothesis class.

PCA-learnable checking: Empirical risk minimisation

ERM algorithm:

$$h_S = \arg\min_{h \in H} R_S(h) = \arg\min_{h \in H} \frac{1}{n} \sum_{i=1}^{n} \ell(X_i, Y_i, h).$$

Check if the following hold when $n > poly(1/\delta, 1/\epsilon)$

$$p\left\{ R(h_S) - \min_{h \in H} R(h) \leq \epsilon \right\} \geq 1 - \delta.$$

$$h^* = \arg\min_{h \in H} R(h)$$

Generalisation error

We have $R_S(h_S) \leq R_S(h^*)$, then

$$
\begin{aligned}
R(h_S) - \min_{h \in H} R(h) &= R(h_S) - R(h^*) \\
&= R(h_S) - R_S(h_S) + R_S(h_S) - R_S(h^*) + R_S(h^*) - R(h^*) \\
&\leq R(h_S) - R_S(h_S) + R_S(h^*) - R(h^*) \\
&\leq |R(h_S) - R_S(h_S)| + |R(h^*) - R_S(h^*)| \\
&\leq \sup_{h \in H} |R(h) - R_S(h)| + \sup_{h \in H} |R(h) - R_S(h)| \\
&= 2 \sup_{h \in H} |R(h) - R_S(h)|.
\end{aligned}
$$

Hypothesis complexity

If A replies B, then $p\{A\} \leq p\{B\}$

Union bound

$$p\left\{\cup_{i=1}^{n} A_i\right\} \leq \sum_{I=1}^{n} p\{A_i\}.$$

$$p\left\{\sup_{h\in H} |R(h) - R_S(h)| \geq \epsilon\right\}$$

$$\leq p\left\{\cup_{h\in H} |R(h) - R_S(h)| \geq \epsilon\right\}$$

$$\leq \sum_{h\in H} p\left\{|R(h) - R_S(h)| \geq \epsilon\right\}$$

$$= |H| p\left\{|R(h) - R_S(h)| \geq \epsilon\right\}$$

$$\leq 2|H| \exp\left(\frac{-2n\epsilon^2}{M^2}\right).$$

Hypothesis complexity

$$p\left\{\sup_{h\in H}|R(h)-R_S(h)|\geq\epsilon\right\}\leq 2|H|\exp\left(\frac{-2n\epsilon^2}{M^2}\right).$$

Let $\delta = 2|H|\exp\left(\frac{-2n\epsilon^2}{M^2}\right)$. We have

$$\epsilon = M\sqrt{\frac{\log|H|+\log 2/\delta}{2n}}.$$

Hypothesis complexity

Thus, with probability at least $1-\delta$, we have

$$\sup_{h\in H}|R(h)-R_S(h)|\leq M\sqrt{\frac{\log|H|+\log 2/\delta}{2n}}.$$

Generalisation bound

Very important inequality:

$$R(h_S) - R_S(h_S) \leq \sup_{h \in H} |R(h) - R_S(h)|.$$

We have

$$R(h_s) \leq R_S(h_S) + \sup_{h \in H} |R(h) - R_S(h)|.$$

Generalisation error bound

PAC learning checking

If the hypothesis class is of finite hypotheses, it is PAC learnable. Because

$$p\left\{R(h_S) - \min_{h\in H} R(h) \leq 2\sup_{h\in H}|R(h) - R_S(h)| \leq 2M\sqrt{\frac{\log|H| + \log(2/\delta))}{2n}}\right\} \geq 1-\delta.$$

Since $\delta = 2|H|\exp\left(\frac{-2n\epsilon^2}{M^2}\right)$. We have

$$n = \frac{M^2}{\epsilon^2}\log\left(\frac{2|H|}{\delta}\right).$$

$n > poly(1/\delta, 1/\epsilon)$

VC dimension

Definition:

VC dimension

The VC dimension of a hypothesis class H is the size of the largest set that can be fully shattered by H:

$$\text{VC dimension}(H) = \max_n \{n : \Pi_H(n) = 2^n\}.$$

Note that VC dimension is designed for binary classification problem. Why? By shattering definition → then should be binary problem

Dictionary learning

Note that

$$\frac{1}{n}\sum_{i=1}^{n}\|x_i - D\alpha_i\|^2 = \frac{1}{n}\|X - DR\|_F^2,$$

where $X = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{d\times n}$,

$R = [\alpha_1, \alpha_2, \ldots, \alpha_n] \in \mathbb{R}^{k\times n}$,

$\|X\|_F = \sqrt{\mathrm{trace}(X^\top X)} = \sqrt{\sum_{i=1}^{d}\sum_{j=1}^{n} X_{i,j}^2}$ is the Frobenius norm of X.

Dictionary learning

Note that

$$\arg \min_{D \in \mathcal{D}, R \in \mathcal{R}} \|X - DR\|_F^2,$$

where $\mathcal{D}$ and $\mathcal{R}$ are some specific domains for D and R.

Dictionary learning

PCA: $A = U\Lambda U^T$

$$A = \underbrace{\begin{bmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_n \\ | & | & & | \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix} \begin{bmatrix} — & \mathbf{u}_1^T & — \\ — & \mathbf{u}_2^T & — \\ & \vdots & \\ — & \mathbf{u}_n^T & — \end{bmatrix}}_{R}$$

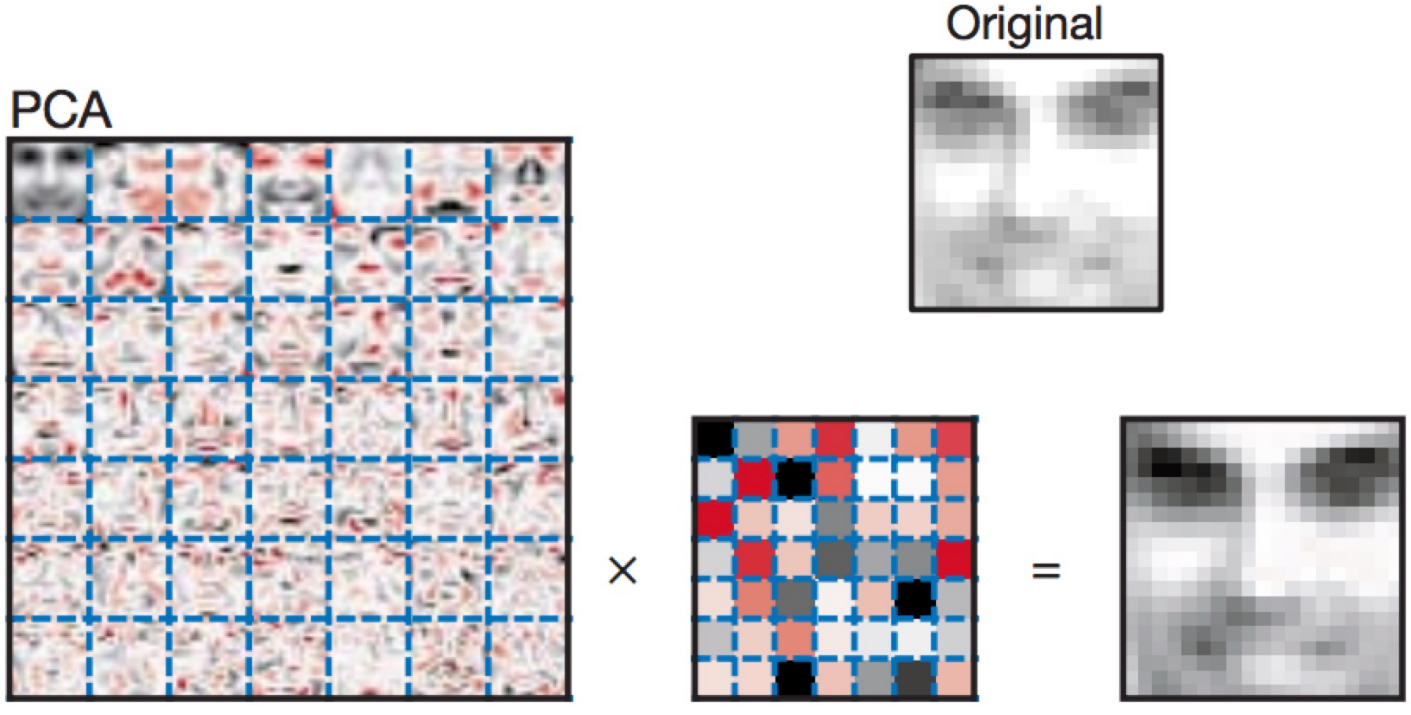
$$\min_{D \in \mathcal{D}, R \in \mathcal{R}} \|X - DR\|_F^2$$

Special requirement: columns of D are orthonormal to each other.

Dictionary learning

PCA: $A = U\Lambda U^T$

$$\alpha^* = \arg\min_{\alpha \in \mathbb{R}^k} \|x - D\alpha\|^2.$$

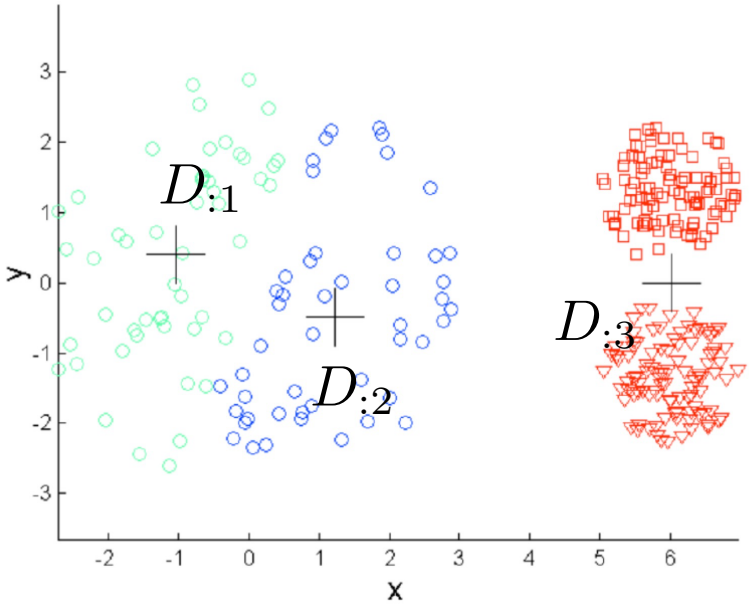


Lee, Daniel D., and H. Sebastian Seung. "Learning the parts of objects by non-negative matrix factorization." Nature 401.6755 (1999): 788.

Dictionary learning

K-means clustering:

$$\min_{D \in \mathcal{D}, R \in \mathcal{R}} \|X - DR\|_F^2$$

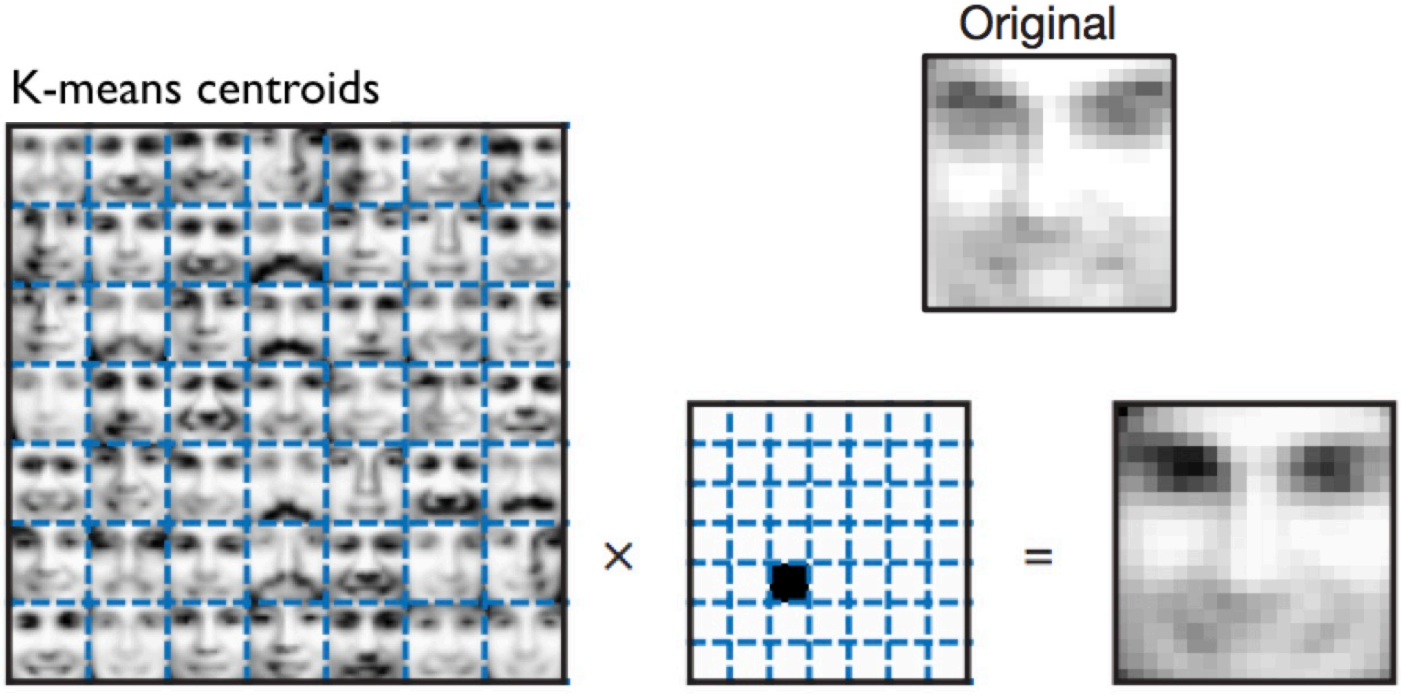


Special requirement: each column of R to be on-hot, i.e., only one have entry equals to one, the other entries are all zeros.

Dictionary learning

K-means clustering:

$$\alpha^* = \arg\min_{\alpha \in \mathbb{R}^k} \|x - D\alpha\|^2.$$



Lee, Daniel D., and H. Sebastian Seung. "Learning the parts of objects by non-negative matrix factorization." Nature 401.6755 (1999): 788.

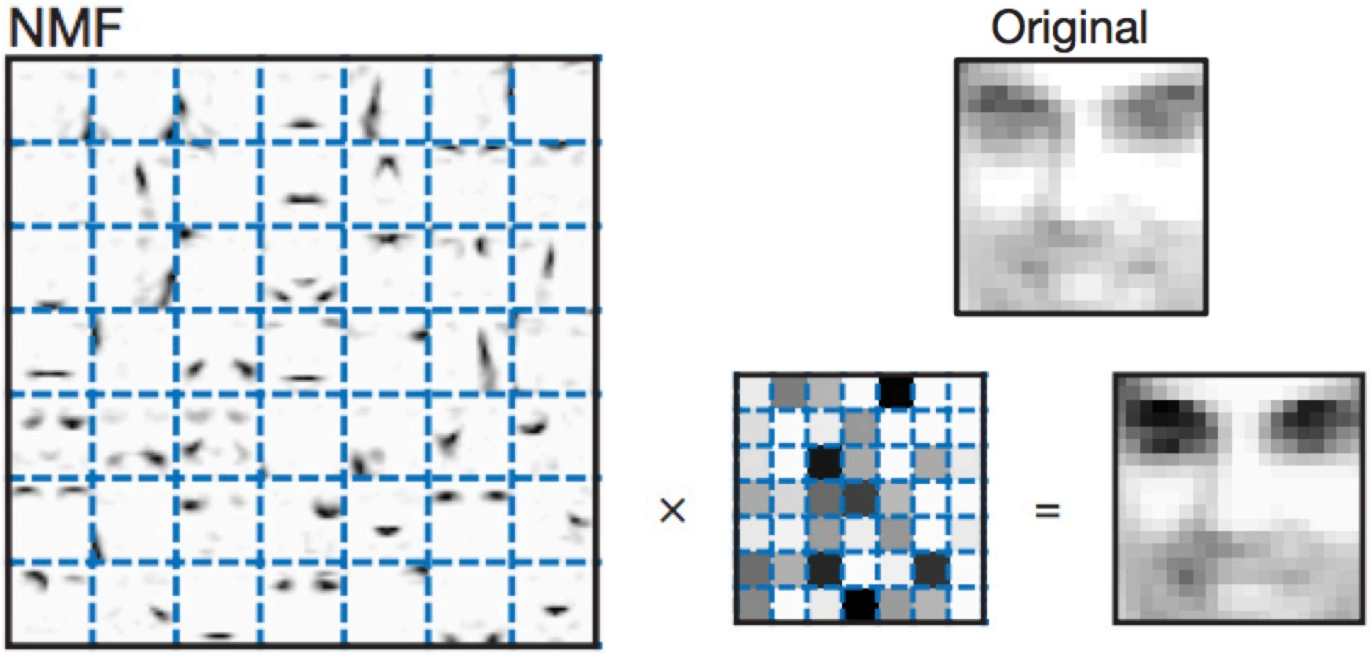
Non-negative matrix factorisation

$$\min_{D \in \mathcal{D}, R \in \mathcal{R}} \|X - DR\|_F^2$$

Special requirement: $\mathcal{D} = \mathbb{R}_+^{d \times k}, \quad \mathcal{R} = \mathbb{R}_+^{k \times n}.$

Non-negative matrix factorisation

$$\alpha^* = \arg\min_{\alpha \in \mathbb{R}^k} \|x - D\alpha\|^2.$$

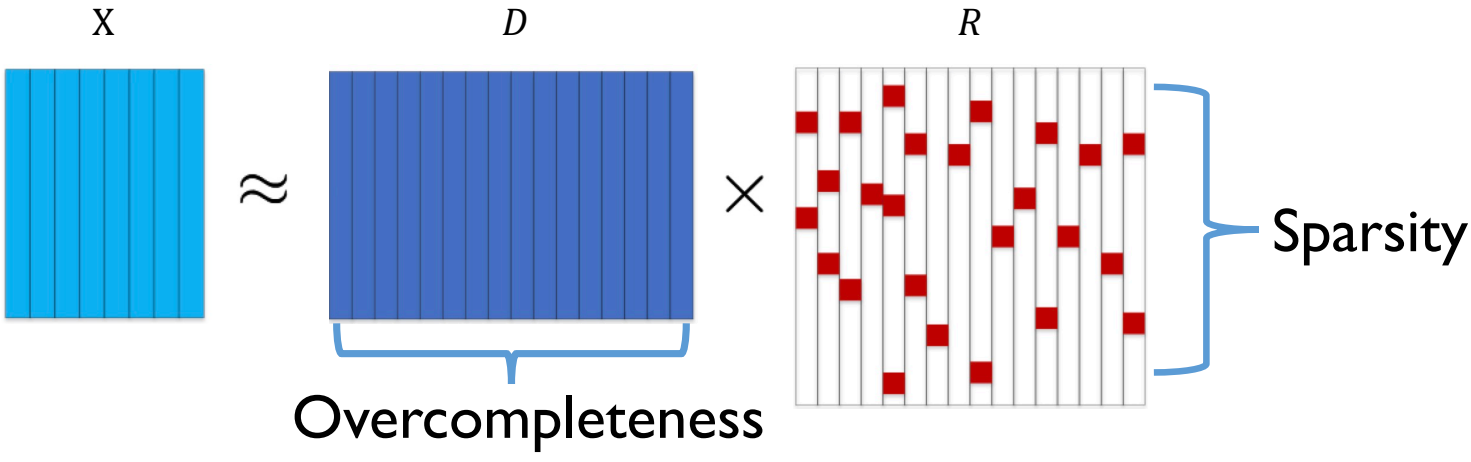


Lee, Daniel D., and H. Sebastian Seung. "Learning the parts of objects by non-negative matrix factorization." Nature 401.6755 (1999): 788.

Sparse coding

Note that

$$\arg \min_{D \in \mathcal{D}, R \in \mathcal{R}} \|X - DR\|_F^2$$



Measure of Sparsity

The objective function:

$$\min_{D \in \mathcal{D}, R \in \mathcal{R}} \underbrace{\|X - DR\|_F^2}_{\text{Data fitting}} + \underbrace{\lambda \psi(R)}_{\text{Sparse regularisation}}$$

Question: how can we design the regularisation to make *R* to be sparse?

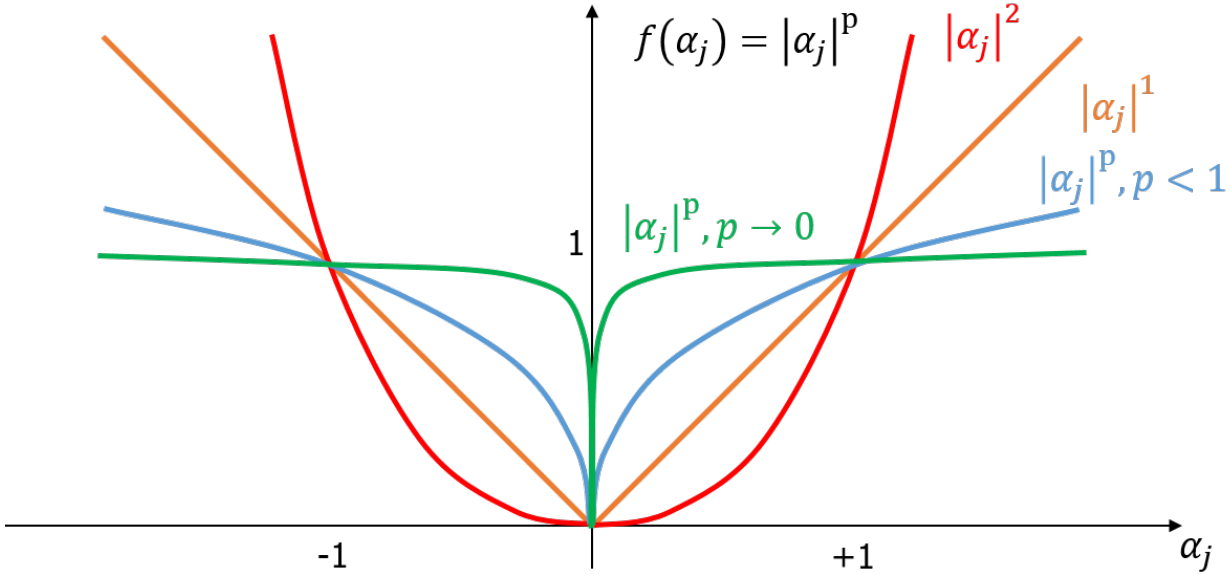
Measure of Sparsity: ℓ_0 norm

ℓ_p norm: $\|\alpha\|_p = \left(\sum_{j=1}^{k} |\alpha_j|^p \right)^{1/p}$, where $\alpha \in \mathbb{R}^k$.

In other words, $\|\alpha\|_p^p = \sum_{j=1}^{k} |\alpha_j|^p$.

Measure of Sparsity: ℓ_0 norm

$$\|\alpha\|_p^p = \sum_{j=1}^{k} |\alpha_j|^p.$$



As $p \to 0$, we get a count of the non-zeros in the vector.

So we can employ $\|\alpha\|_0$ to measure sparsity.

Measure of Sparsity: ℓ_0 norm

As $p \rightarrow 0$, we get a count of the non-zeros in the vector.
So we can employ $\|\alpha\|_0$ to measure sparsity.

However, the ℓ_0 minimisation is not easy. How to do?

Maximum Likelihood Estimation (MLE)

THE UNIVERSITY OF SYDNEY

According to the i.i.d. assumption, the likelihood function is rewritten as

$$p(S|\theta) = \prod_{i=1}^{n} p(x_i, y_i|\theta).$$

Sometimes, we also define the likelihood as follows

$$p(S|\theta) = \prod_{i=1}^{n} p(y_i|x_i, \theta).$$

Maximum Likelihood: Find the value of θ maximising the likelihood $p(S|\theta)$, i.e., it is the value of θ that makes the observed data the "most probable".

Maximum A Posterior (MAP)

THE UNIVERSITY OF SYDNEY

Bayes' rule

$$P(\theta|S) = \frac{P(S|\theta)P(\theta)}{P(S)}$$

$$P(\theta|S) \propto P(S|\theta)P(\theta)$$

$$\arg\max_{\theta} P(\theta|S) = \arg\max_{\theta} P(S|\theta)P(\theta)$$

$$\arg\min_{\theta}(-\log P(\theta|S)) = \arg\min_{\theta}(-\log P(S|\theta) - \log P(\theta))$$

Maximum Posterior $P(\theta|S)$: the observed data makes the value of θ the "most probable".

Maximum A Posterior (MAP)

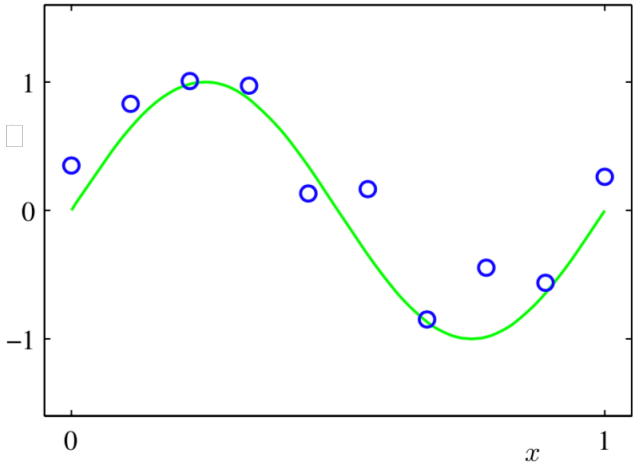
THE UNIVERSITY OF SYDNEY

Because of Bayes' rule

$$\arg\max_{\theta} p(\theta|S) = \arg\max_{\theta} p(S|\theta)p(\theta)$$

$$-\ln(\cdot) = -\ln(\cdot)$$

$$\begin{aligned}
\arg\min_{h}(-\ln p(h|S,\beta^{-1})) &= \arg\min_{h}(-\ln(p(S|X,h,\beta^{-1})p(h))) \\
&= \arg\min_{h}(-\ln p(S|X,h,\beta^{-1}) - \ln p(h)) \\
&= \arg\min_{h}(-\frac{n}{2}\ln\beta + \frac{n}{2}\ln(2\pi) + \frac{\beta}{2}nR_S(h) - \ln p(h))
\end{aligned}$$



$$h(x) = w_0 + w_1 x + \ldots + w_9 x^9$$

Assuming the prior distribution:

$$p(h) = \prod_{i=0}^{9} \sqrt{\frac{\tau}{2\pi}} \exp\left(-\frac{\tau w_i^2}{2}\right)$$

Then, we have

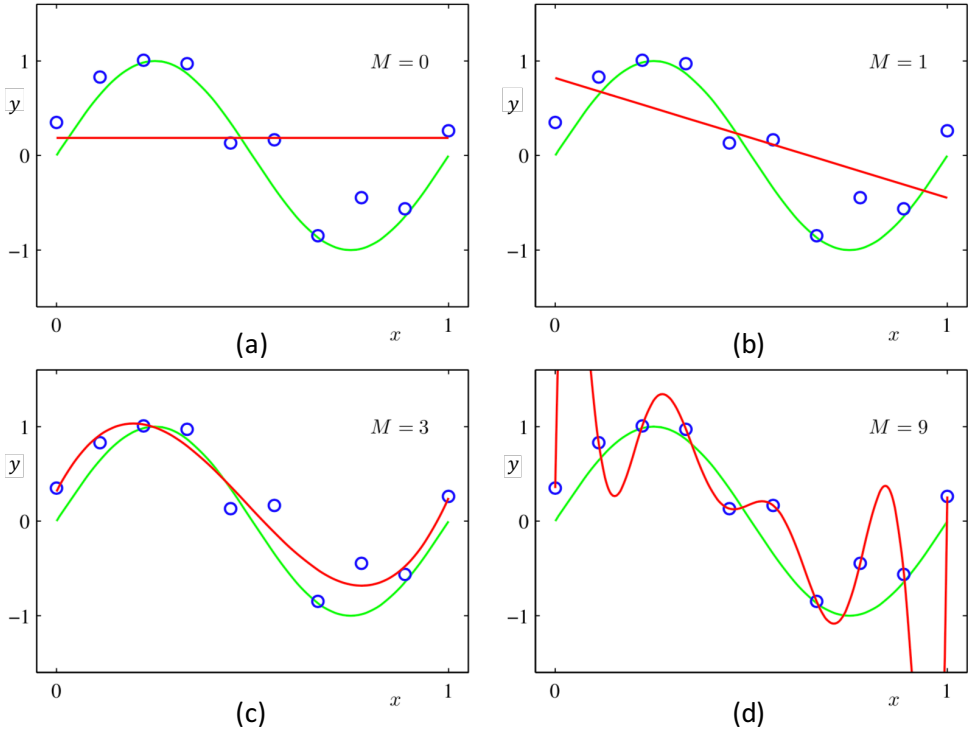
$$\arg\min_h(-\ln p(h|S, \beta^{-1})) = \arg\min_h(-\frac{n}{2}\ln\beta + \frac{n}{2}\ln(2\pi) + \frac{\beta}{2} n R_S(h)$$
$$- 5\ln\tau + 5\ln(2\pi) + \frac{\tau}{2} n \sum_{i=0}^{9} w_i^2)$$

Minimising above equals

$$\min R_S(h) + \lambda \sum_{i=0}^{9} w_i^2 = R_S(h) + \lambda \|w\|_2^2, \quad \lambda = \frac{\tau}{\beta}$$

Bishop's book: "Pattern Recognition and Machine Learning"

Linear regression



- True target: $\sin(2\pi x)$ with small Gaussian noises.
- $h(x) = w_0 + w_1 x + \cdots + w_M x^M$
- $R_S(h) = \frac{1}{n}\sum_{i=1}^{n}\big(y_i - h(x_i)\big)^2$

Bishop's book: "Pattern Recognition and Machine Learning"

Underfitting and overfitting

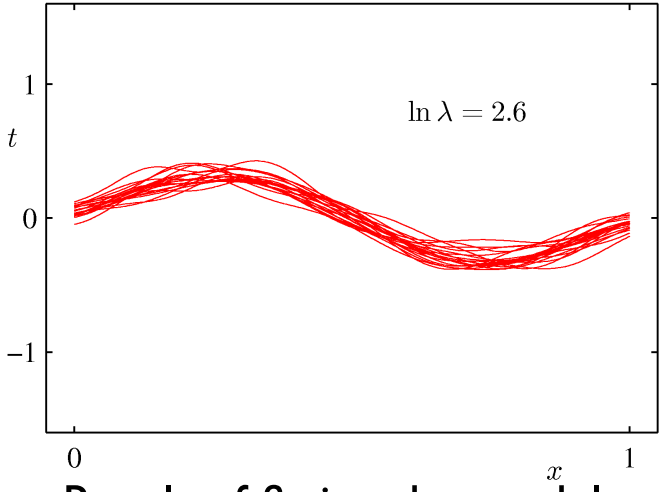
THE UNIVERSITY OF SYDNEY

- **Underfitting** is a phenomenon that the learned model does not fit the training data well; that is, large empirical risk.
- **Overfitting** is a phenomenon that the learned model fits the training data very well but it cannot generalise well to unseen examples drawn from the same distribution; that is, large difference between training and test errors.

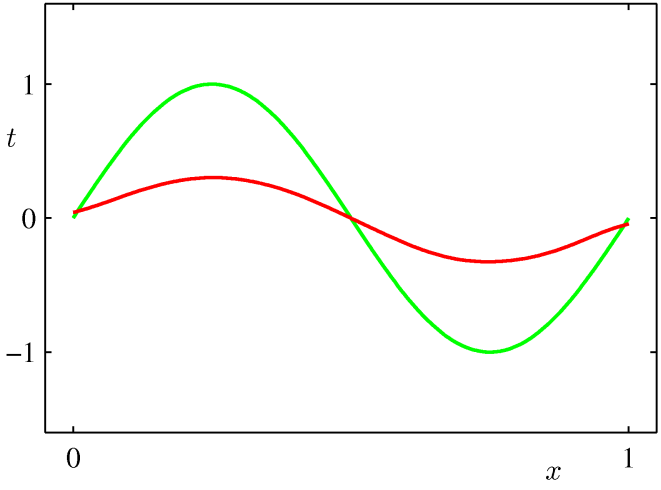
Bias-Variance Visualisation

THE UNIVERSITY OF SYDNEY

20 datasets with varying regularisation parameter.



Result of fitting the model to each dataset.

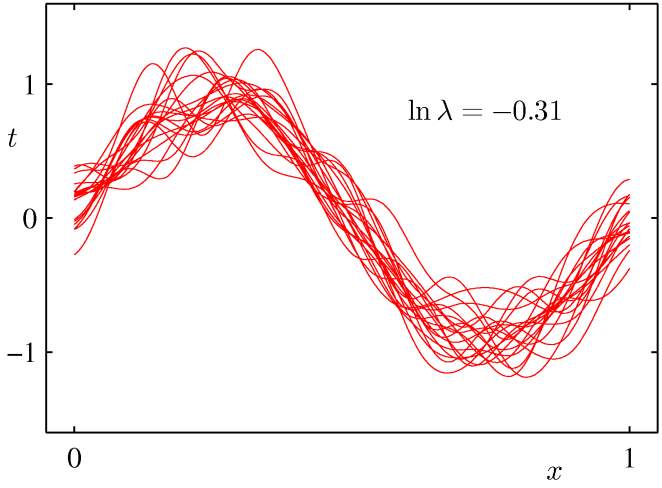


Average of the fits.

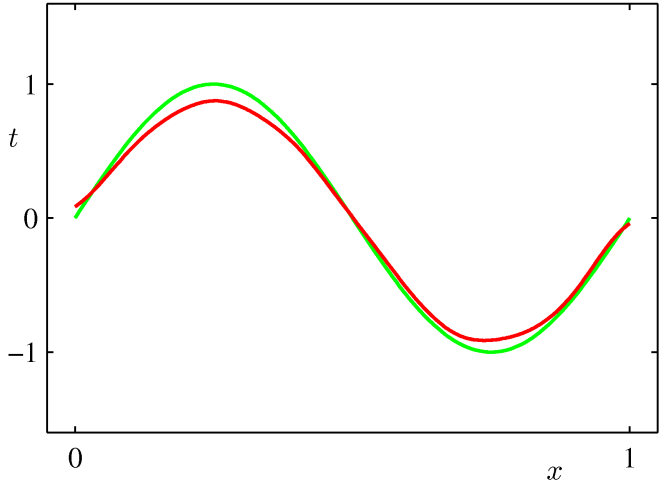
Bias-Variance Visualisation

THE UNIVERSITY OF SYDNEY

20 datasets with varying regularisation parameter.



Result of fitting the model to each dataset.

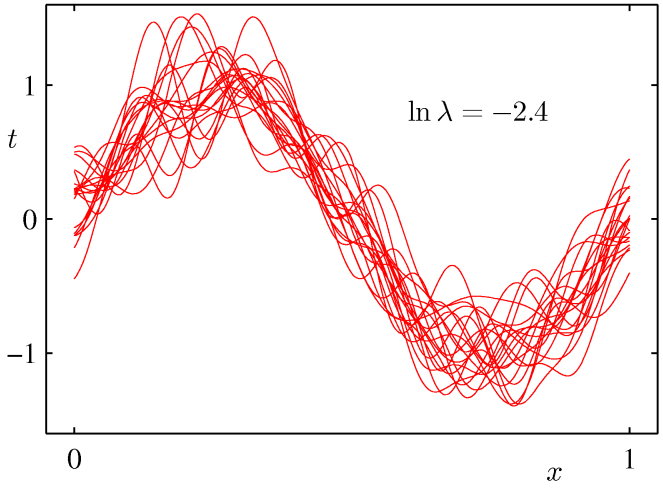


Average of the fits.

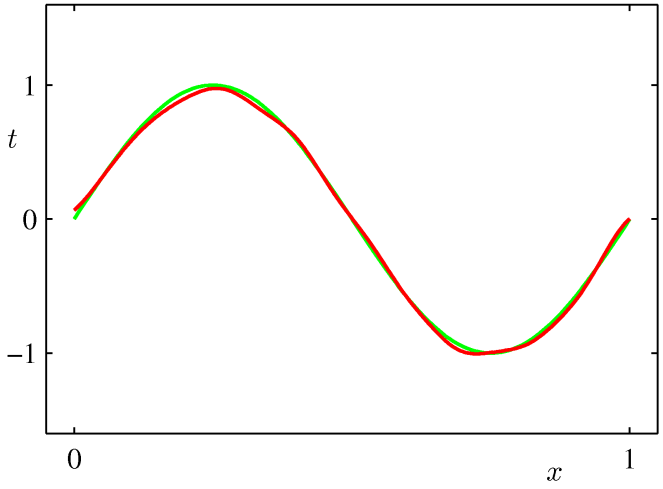
Bias-Variance Visualisation

THE UNIVERSITY OF SYDNEY

20 datasets with varying regularisation parameter.



Result of fitting the model to each dataset.

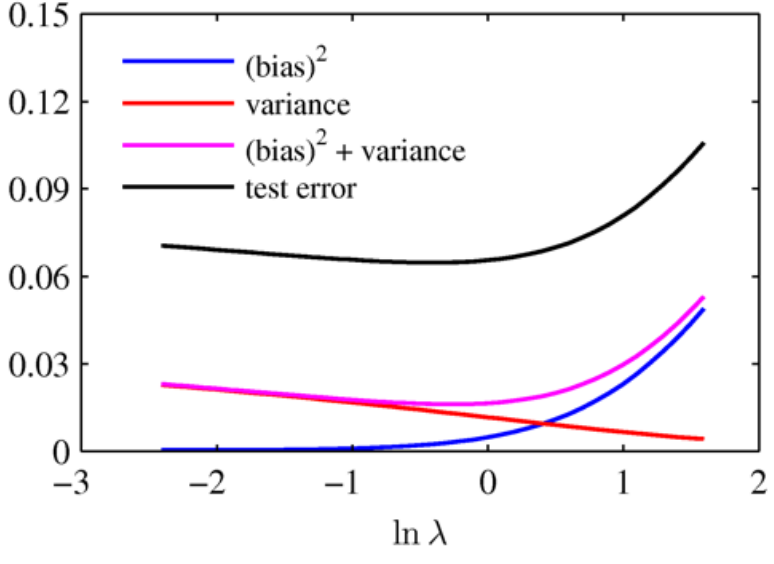


Average of the fits.

The Bias-Variance Trade Off

THE UNIVERSITY OF SYDNEY

From these plots, we note that an over-regularised model (large λ) will have a high bias, while an under-regularised model (small λ) will have a high variance.



Why the above phenomenon happens?

Bias-Variance vs Under-over fitting

- High variance implies overfitting.

- High bias implies underfitting.

Avoid overfitting

THE UNIVERSITY OF SYDNEY

- **Reducing hypothesis complexity**: the hypothesis fits the training data too well because it is too complex.
- **Increasing sample size**: According to the law of large numbers, with more training examples, the empirical risk is closer to the expected risk. Increasing sample size will be helpful to learning the best hypothesis.

Avoid overfitting

- is overfitting caused by data noise?

Optimality criterion

Let $$g(t) = f(th), t \in \mathbb{R}, h \in \mathbb{R}^d.$$

We have $$g'(t) = \nabla f(th)^\top h.$$

$F(x) = g(f(x)),$ then $F'(x) = g'(f(x))f'(x).$

Chain rule for derivatives

If *h* is the minimiser, we have $\nabla f(h) = \mathbf{0}$. Then,

$$g'(1) = 0.$$

In other words, to minimise *f(h)*, we should find an *h* such that

$$g'(1) = 0.$$

Surrogate loss function robustness

- Least square loss: $g'(1) = \frac{1}{n}\sum_{i=1}^{n} 2(y_i - h(x_i))(-h(x_i))$
- Absolute loss: $g'(1) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{|y_i - h(x_i)|}(y_i - h(x_i))(-h(x_i))$
- Cauchy loss:

$$g'(1) = \frac{1}{n}\sum_{i=1}^{n} \frac{2}{\gamma^2 + (y_i - h(x_i))^2}(y_i - h(x_i))(-h(x_i))$$

- Correntropy loss (Welsch loss):

$$g'(1) = \frac{1}{n}\sum_{i=1}^{n} \frac{2}{\sigma^2 \exp\left(\frac{y_i - h(x_i)}{\sigma}\right)^2}(y_i - h(x_i))(-h(x_i))$$

Surrogate loss function

THE UNIVERSITY OF SYDNEY

In other words, to minimise *f(h)*, we should find an *h* such that

$$g'(1) = 0.$$

- All $g'(1)$ w.r.t. the above four loss functions has the term $c_i = (y_i - h(x_i))(-h(x_i))$, we treat them as the bases of contribution to optimising the empirical risks. We can see that different surrogate loss functions assign different weights to the bases. A surrogate loss function is more robust to large noise if it assign a smaller weights to the bases as the error (or noise) is going bigger.

Surrogate loss function robustness

THE UNIVERSITY OF SYDNEY

- Least squares loss: $g'(1) = \frac{1}{n}\sum_{i=1}^{n} 2(y_i - h(x_i))(-h(x_i))$
- Absolute loss: $g'(1) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{|y_i - h(x_i)|}(y_i - h(x_i))(-h(x_i))$
- Cauchy loss:

$$g'(1) = \frac{1}{n}\sum_{i=1}^{n} \frac{2}{\gamma^2 + (y_i - h(x_i))^2}(y_i - h(x_i))(-h(x_i))$$

- Correntropy loss (Welsch loss):

$$g'(1) = \frac{1}{n}\sum_{i=1}^{n} \frac{2}{\sigma^2 \exp\left(\frac{y_i - h(x_i)}{\sigma}\right)^2}(y_i - h(x_i))(-h(x_i))$$

Domain Adaptation and Transfer Learning

THE UNIVERSITY OF SYDNEY

Machine can also find the common knowledge between data and transfer the knowledge from one domain to another one. This relates two terms in machine learning: domain adaptation and transfer learning.

In machine learning, we can exploit training examples drawn from some related domain (the source domain) to improve the performance on a related domain (the target domain).

Importance Reweighting

THE UNIVERSITY OF SYDNEY

Let denote $R^T(h) = \mathbb{E}_{(X,Y)\sim p_t(X,Y)}[\ell(X,Y,h)]$.

We have
$$\begin{aligned} R^T(h) &= \mathbb{E}_{(X,Y)\sim p_t(X,Y)}[\ell(X,Y,h)] \\ &= \int_{(X,Y)} \ell(X,Y,h) p_t(X,Y) dX dY \\ &= \int_{(X,Y)} \ell(X,Y,h) \frac{p_t(X,Y)}{p_s(X,Y)} p_s(X,Y) dX dY \\ &= \mathbb{E}_{(X,Y)\sim p_s(X,Y)}[\frac{p_t(X,Y)}{p_s(X,Y)} \ell(X,Y,h)] \\ &= \mathbb{E}_{(X,Y)\sim p_s(X,Y)}[\beta(X,Y)\ell(X,Y,h)], \end{aligned}$$

where the weights $\beta(X,Y) = p_t(X,Y)/p_s(X,Y)$ represent the changes across domains.

Transfer Learning Models

THE UNIVERSITY OF SYDNEY

Let $\{(x_1^S, y_1^S), \ldots, (x_{n_S}^S, y_{n_S}^S)\}$ be the training sample of source domain. We can approximate $R^T(h)$ by

$$\frac{1}{n}\sum_{i=1}^{n_S}\beta(x_i^S, y_i^S)\ell(x_i^S, y_i^S, h).$$

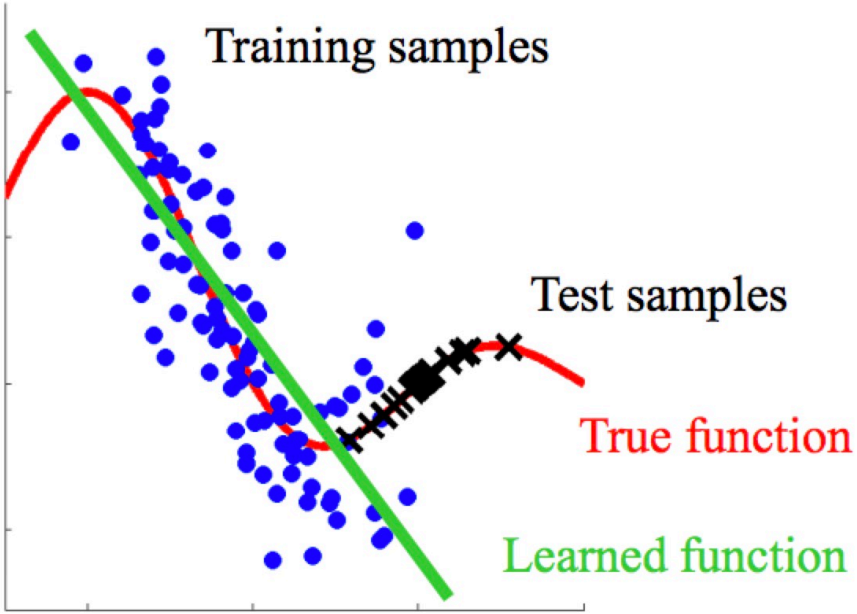
Our target is to learn $\beta(X, Y)$.

We introduce two models, where $\beta(X, Y)$ can be effectively learned.

Covariate Shift Model

THE UNIVERSITY OF SYDNEY

In this model, we assume that $p_t(Y|X) = p_s(Y|X)$ and that $p_s(X) \neq p_t(X)$.



http://iwann.ugr.es/2011/pdf/InvitedTalk-FHerrera-IWANN11.pdf

Covariate Shift Model

THE UNIVERSITY OF SYDNEY

If we assume that $p_t(Y|X) = p_s(Y|X)$, we have

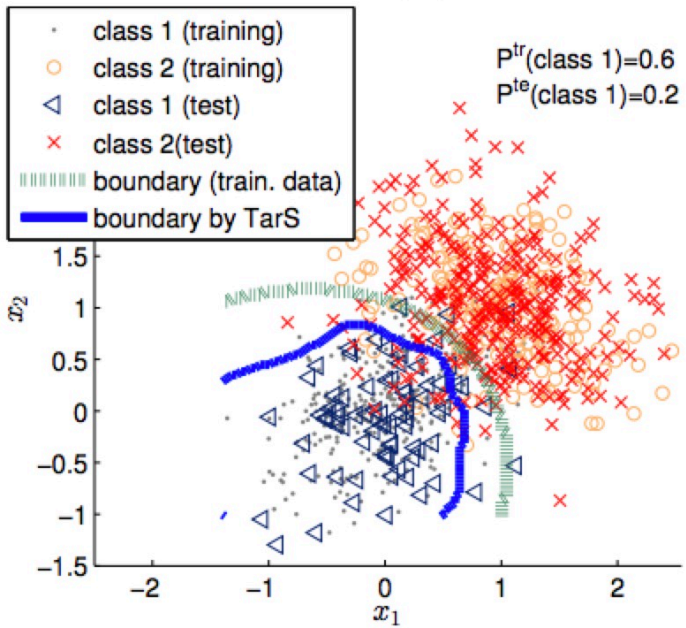
$$\beta(X,Y) = \frac{p_t(X,Y)}{p_s(X,Y)} = \frac{p_t(Y|X)p_t(X)}{p_s(Y|X)p_s(X)} = \frac{p_t(X)}{p_s(X)} = \beta(X).$$

Note that $\beta(X)$ can be learned by kernel mean matching.

Target Shift Model

THE UNIVERSITY OF SYDNEY

In this model, we assume that $p_t(X|Y) = p_s(X|Y)$ and that $p_t(Y) \neq p_s(Y)$.



Zhang, K., Schölkopf, B., Muandet, K., & Wang, Z. (2013, February). Domain adaptation under target and conditional shift. In International Conference on Machine Learning (pp. 819-827).

Target Shift Model

THE UNIVERSITY OF SYDNEY

If we further assume that $p_t(X|Y) = p_s(X|Y)$, we have

$$\beta(X,Y) = \frac{p_t(X,Y)}{p_s(X,Y)} = \frac{p_t(X|Y)p_t(Y)}{p_s(X|Y)p_s(Y)} = \frac{p_t(Y)}{p_s(Y)} = \beta(Y).$$

Note that $\beta(Y)$ is not easy to learn if the target domain does not have any labels. How to deal with this problem?

Kernel Mean Matching

THE UNIVERSITY OF SYDNEY

Denote $\phi : X \to \mathcal{H}$, where $\mathcal{H}$ is a RKHS (Reproducing Kernel Hilbert Space) with the kernel function $K(x_1, x_2) = \langle \phi(x_1), \phi(x_2) \rangle$ and $\langle \cdot, \cdot \rangle$ is the inner product operator.

Let $\mu(p(X)) = \mathbb{E}_{X \sim p(X)}[\phi(X)]$,
where $p(X)$ is a marginal distribution on the feature space.

The expectation μ is a bijective function if K is a universal kernel.

Theorem 1.2 in
Gretton, A., Smola, A. J., Huang, J., Schmittfull, M., Borgwardt, K. M., & Schölkopf, B. (2009). Covariate shift by kernel mean matching. http://www.gatsby.ucl.ac.uk/~gretton/papers/covariateShiftChapter.pdf

Kernel Mean Matching

THE UNIVERSITY OF SYDNEY

If we have

$$\mu(p_t(X)) = \mathbb{E}_{X\sim p_s(X)}[\beta(X)\phi(X)]$$

and $\beta(X) \geq 0$, $\mathbb{E}_{X\sim p_s(X)}[\beta(X)] = 1$,

what is the relationship between $\beta(X)p_s(X)$ and $p_t(X)$?

They are equal to each other. Because

$\mu(\beta(X)p_s(X)) = \mu(p_t(X))$ and thus $\beta(X)p_s(X) = p_t(X)$.

Kernel Mean Matching

THE UNIVERSITY OF SYDNEY

We can learn the weights

$$\min_{\beta} \|\mu(p_t(X)) - \mathbb{E}_{X\sim p_s(X)}[\beta(X)\phi(X)]\|^2$$
$$\text{subject to } \beta(X) \geq 0, \mathbb{E}_{X\sim p_s(X)}[\beta(X)] = 1.$$

If we only have training samples from different domains, e.g., $\{x_1^S, \ldots, x_{n_S}^S\} \sim p_s(X)^{n_S}$ and $\{x_1^T, \ldots, x_{n_T}^T\} \sim p_t(X)^{n_T}$, how can we learn $\beta(X)$?

Kernel Mean Matching

THE UNIVERSITY OF SYDNEY

Since we cannot calculate the expectation, we will use the empirical mean to approximate them, e.g.,

$$\min_{\beta} \left\| \frac{1}{n_T} \sum_{i=1}^{n_T} \phi(x_i^T) - \frac{1}{n_S} \sum_{i=1}^{n_S} \beta(x_i^S)\phi(x_i^S) \right\|^2$$

$$\text{subject to } \beta(x_i^S) \geq 0, \frac{1}{n_S} \sum_{i=1}^{n_S} \beta(x_i^S) = 1.$$

Learning with Noisy Labels

THE UNIVERSITY OF SYDNEY

Problem Setup

- Given the training examples $\{(X_i, \tilde{Y}_i)\}_{1 \le i \le n} \sim D_\rho(X, \tilde{Y})^n$.

- The target is to learn a discriminant function f_n: $\mathcal{X} \to \mathbb{R}$ such that the classifier predicts the correct label y given an observation $\boldsymbol{x}$.

Model Label Noise

THE UNIVERSITY OF SYDNEY

A probabilistic model:

$$\rho_Y(X) = P\left(\tilde{Y}\middle|Y, X\right),$$

where X is the feature, Y is the unobservable true label, and $\tilde{Y}$ is the observed noisy label.

$$\rho_{+1}(X) = P(\tilde{Y} = -1|Y = 1, X); \rho_{-1}(X) = P(\tilde{Y} = 1|Y = -1, X).$$

Note that if there is no label noise, we have

$$P\left(\tilde{Y} = 1\middle|Y = 1, X\right) = P\left(\tilde{Y} = -1\middle|Y = -1, X\right) = 1$$

otherwise

$$P\left(\tilde{Y} = 1\middle|Y = -1, X\right), P\left(\tilde{Y} = -1\middle|Y = 1, X\right) \in (0,1).$$

Class-dependent Label Noise

THE UNIVERSITY OF SYDNEY

Viewing the noisy data and clean data are sampled from two domains, importance reweighting can be applied.

$$R_{D,L}(f) = \mathbb{E}_{(X,Y)\sim D}[L(f(X),Y)] = \int P_D(X,Y)\, L(f(X),Y)\mathrm{dXdY}$$

$$= \int P_{D_\rho}(X,Y)\frac{P_D(X,Y)}{P_{D_\rho}(X,Y)} L(f(X),Y)\mathrm{dXdY}$$

$$= \mathbb{E}_{(X,Y)\sim D_\rho}[\frac{P_D(X,Y)}{P_{D_\rho}(X,Y)} L(f(X),Y)]$$

$$= \mathbb{E}_{(X,Y)\sim D_\rho}[\beta(X,Y)L(f(X),Y)] \quad \text{where } \beta(\boldsymbol{x}, y) = \frac{P_D(X=\boldsymbol{x},Y=y)}{P_{D_\rho}(X=\boldsymbol{x},\widetilde{Y}=y)}.$$

Liu, Tongliang, and Dacheng Tao. "Classification with noisy labels by importance reweighting." IEEE Transactions on pattern analysis and machine intelligence 38.3 (2016): 447-461.

Class-dependent Label Noise

THE UNIVERSITY OF SYDNEY

Viewing the noisy data and clean data are sampled from two domains, importance reweighting can be applied.

Recall that

$$P_{D_\rho}(\widetilde{Y} = y|X = \boldsymbol{x}) = (1 - \rho_{+1} - \rho_{-1})P_D(Y = y|X = \boldsymbol{x}) + \rho_{-y}$$

Then

$$\beta(\boldsymbol{x}, y) = \frac{P_D(X=\boldsymbol{x},Y=y)}{P_{D_\rho}(X=\boldsymbol{x},\widetilde{Y}=y)} = \frac{P_D(Y=y|X=\boldsymbol{x})}{P_{D_\rho}(\widetilde{Y}=y|X=\boldsymbol{x})} = \frac{P_{D_\rho}(\widetilde{Y}=y|X=\boldsymbol{x})-\rho_{-y}}{(1-\rho_{+1}-\rho_{-1})P_{D_\rho}(\widetilde{Y}=y|X=\boldsymbol{x})}.$$

Liu, Tongliang, and Dacheng Tao. "Classification with noisy labels by importance reweighting." IEEE Transactions on pattern analysis and machine intelligence 38.3 (2016): 447-461.

Learning with Noisy Labels

Let T be the following flip matrix (also called transition matrix), e.g.,

$$T = \begin{bmatrix} P(\tilde{Y}=1|Y=1) & P(\tilde{Y}=1|Y=2) & \dots & P(\tilde{Y}=1|Y=C) \\ P(\tilde{Y}=2|Y=1) & P(\tilde{Y}=2|Y=2) & \dots & P(\tilde{Y}=2|Y=C) \\ \vdots & \vdots & \vdots & \vdots \\ P(\tilde{Y}=C|Y=1) & P(\tilde{Y}=C|Y=2) & \dots & P(\tilde{Y}=C|Y=C) \end{bmatrix}.$$

If we assume that given the clean label, the noisy label is independent with the instance, we have that $P(\tilde{Y}|Y) = P(\tilde{Y}|Y,X)$, and that

Forward

$$[P(\tilde{Y}=1|X),\dots,P(\tilde{Y}=C|X)]^\top = T\,[P(Y=1|X),\dots,P(Y=C|X)]^\top,$$

or $$[P(Y=1|X),\dots,P(Y=C|X)]^\top = T^{-1}[P(\tilde{Y}=1|X),\dots,P(\tilde{Y}=C|X)]^\top.$$

Backward

The above means that we can infer the clean class posterior by employing the noisy class posterior and the inverse transition matrix.

RL Framework (Objective)

Given current state is s_0, we want to find the optimal policy (best strategy) π^* that maximises the expected cumulative reward, i.e.,

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{t\geq 0} \gamma^t r_t | S_0 = s_0, \pi\right],$$

With

$$a_t \sim \pi(A_t | S_t = s_t),\ s_{t+1} \sim P(S_{t+1} | A_t = a_t, S_t = s_t),$$

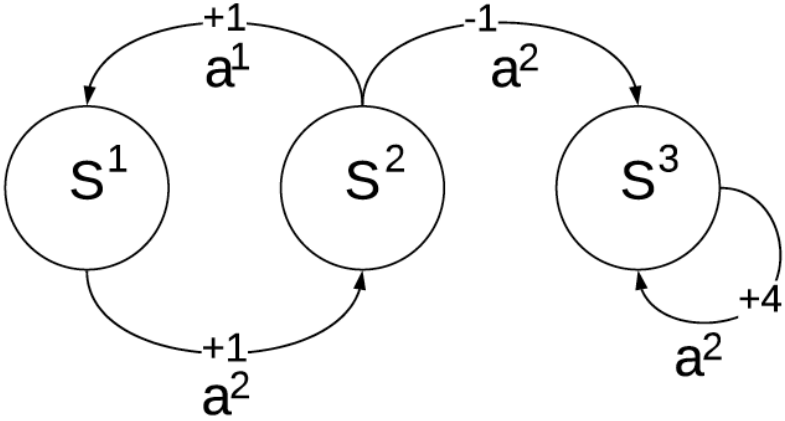
and a discounted factor

$$\gamma \in (0, 1)$$

Diagram of Environment: A Toy RL Problem

THE UNIVERSITY OF SYDNEY

$\mathcal{A} = \{a^1, a^2\}.$
$\mathcal{S} = \{S^1, S^2, S^3\}.$



In this setting,

$$P(S_{t+1} = s^1 | A_t = a^1, S_t = s^2) = 1 \quad P(R_t = -1 | A_t = a^2, S_t = s^2) = 1$$

Diagram of Environment: A Toy RL Problem

Given initial state $s_0 = s^1$ which policy is better?

For all $t \leq 0$, we have

$$\pi^1(A_t = a^2|S_t = s^1) = 1,\ \pi^1(A_t = a^1|S_t = s^2) = 1,\ \pi^1(A_t = a^2|S_t = s^2) = 0,$$
$$\pi^1(A_t = a^2|S_t = s^3) = 1;$$
$$\pi^2(A_t = a^2|S_t = s^1) = 1,\ \pi^2(A_t = a^1|S_t = s^2) = 0,\ \pi^2(A_t = a^2|S_t = s^2) = 1,$$
$$\pi^2(A_t = a^2|S_t = s^3) = 1.$$

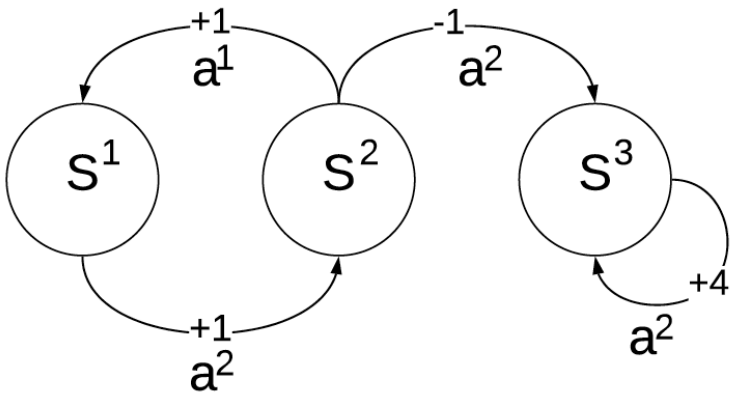
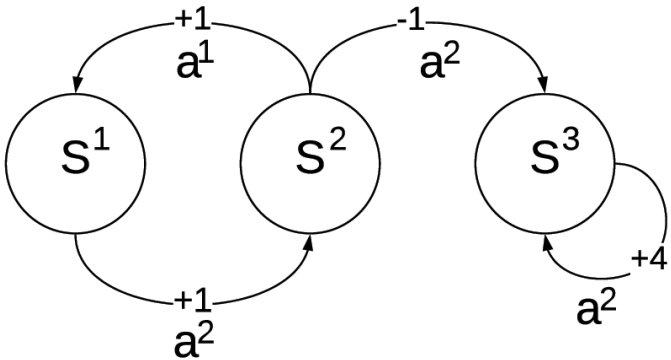


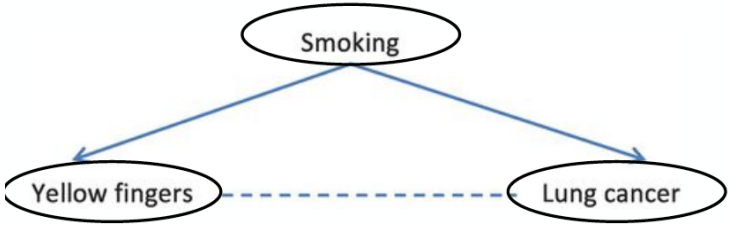
Diagram of Environment: A Toy RL Problem

THE UNIVERSITY OF SYDNEY

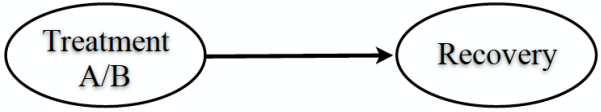
$\pi^*(A_t = a^2|S_t = s^1) = 1$, $\pi^*(A_t = a^1|S_t = s^2) = 0$, $\pi^*(A_t = a^2|S_t = s^2) = 1$, $\pi^*(A_t = a^2|S_t = s^3) = 1$.



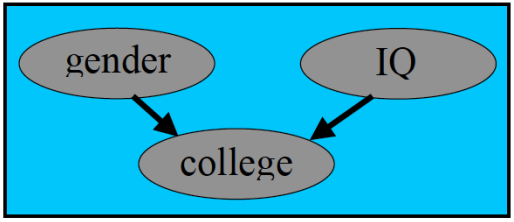
Ways to Produce Dependence



Common Cause



Causal relation



Conditional dependence given common effect

Causal Inference

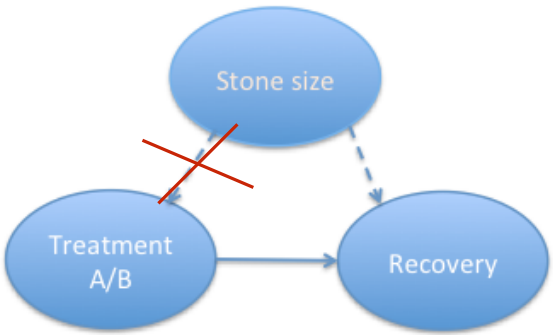
THE UNIVERSITY OF SYDNEY

	Treatment A	Treatment B
Small Stones	*Group 1* **93% (81/87)**	*Group 2* 87% (234/270)
Large Stones	*Group 3* **73% (192/263)**	*Group 4* 69% (55/80)
Both	78% (273/350)	**83% (289/350)**

T - Treatment
R - Recovery

$$P(\mathrm{R} \mid T) = \sum_S P(R \mid T, S) P(S \mid T)$$

$$P(\mathrm{R} \mid do(T)) = \sum_S P(R \mid T, S) P(S)$$



$$P(\text{recovery=yes} \mid do(\text{treatment=A})) = 0.832.$$

$$P(\text{recovery=yes} \mid do(\text{treatment=B})) = 0.7818.$$

Feature- and Parameter-based MTL models

$$\min_{w_0,\Delta W,P} \frac{1}{m}\sum_{i=1}^{m}\frac{1}{n_i}\sum_{j=1}^{n_i}\ell(P^\top x_j^i, y_j^i, w_0+\Delta w^i)+\lambda\|\Delta W\|_F^2,$$

$$s.t.\ PP^\top = I.$$

The above model learns the feature projection map P and the commonly shared parameter w_0 to enhance the relatedness among tasks.

Li, Ya, Xinmei Tian, Tongliang Liu, and Dacheng Tao. "Multi-Task Model and Feature Joint Learning." In IJCAI, pp. 3643-3649. 2015.

Relationship to Transfer Learning

In MTL, there is no distinction among different tasks and the objective is to improve the performance of all the tasks. However, in transfer learning which is to improve the performance of a target task with the help of source tasks, the target task plays a more important role than source tasks.

Exam Details

THE UNIVERSITY OF SYDNEY

- Date: 17 Nov at 5:00pm Sydney time
- Short release take-home (open-book) exam
- Duration: 3hours + 10 minutes reading time + 15 minutes for uploading (do not treat the 15mins as extra writing time)

Exam Details

THE UNIVERSITY OF SYDNEY

- Latex answer template. we have practised!
- Only the generated PDF is allowed to submit.)

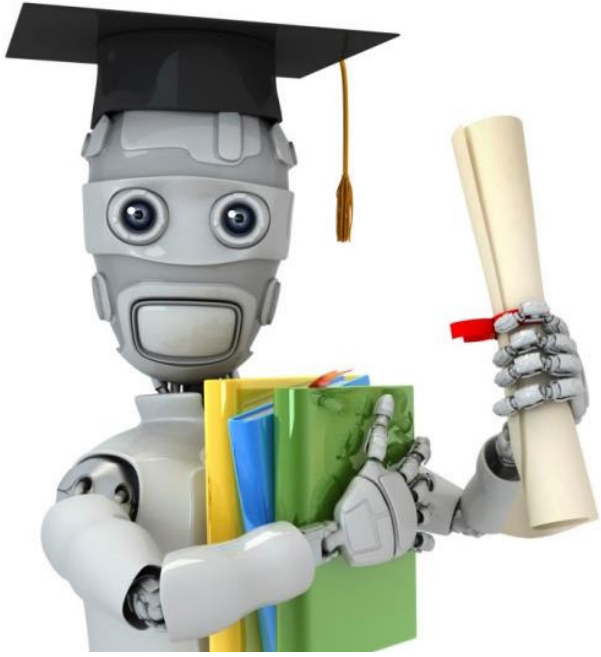
Exam: what to expect?

THE UNIVERSITY OF SYDNEY

- Main concepts and numerical questions
- Theoretical on the main concepts
- Philosophies of different methods
- Pros vs Cons for different methods

THE UNIVERSITY OF SYDNEY

Wish you a successful machine learning and data mining career!



Unit of Study Surveys (USS)

Go to https://student-surveys.sydney.edu.au/students/ and login

Q&A

Thanks!