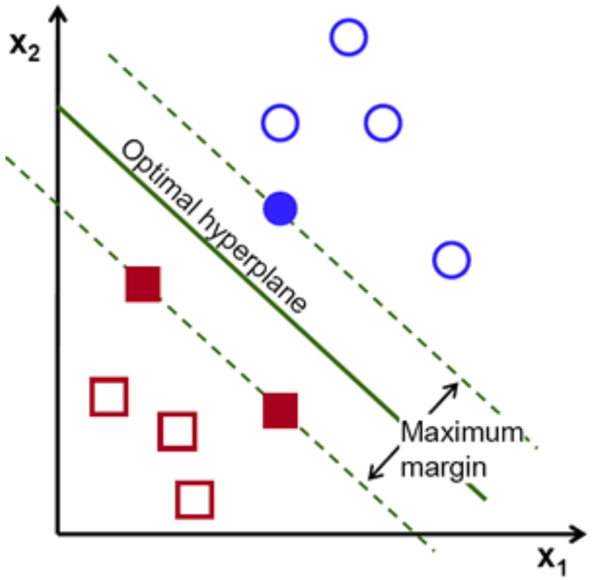
Machine Learning Basics

Jian Tang
HEC Montreal
Mila-Quebec AI Institute
Email: jian.tang@hec.ca

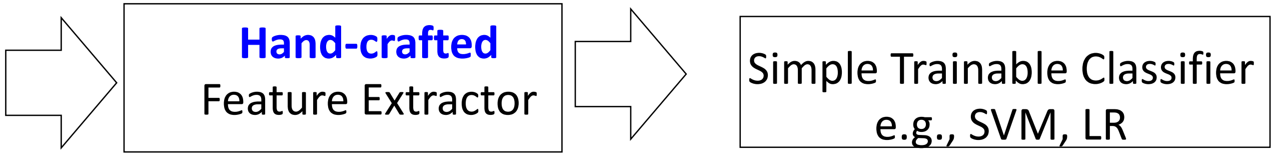
Machine Learning

- “**Machine learning** is a field of computer science that uses statistical techniques to give computer systems the ability to "learn" (i.e., progressively improve performance on a specific task) with data, without being explicitly programmed.”

-Wikipedia



Support vector machines



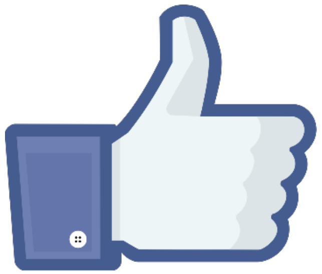
Domain experts

Learning Algorithms

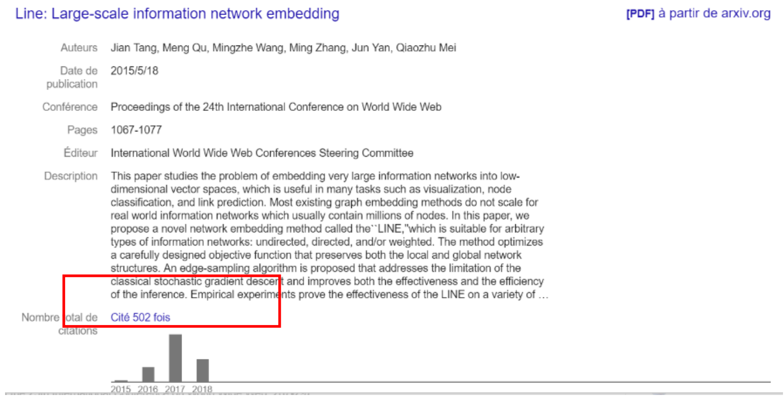
- A machine learning algorithm is an algorithm that is able to **learn** from data (or experience).
- Learning: "A Computer Program is said to learn from experience **E** with respect to some class of tasks **T** and performance measure **P**, if its performance at tasks in **T**, as measured by **P**, improve with experience **E**."– Mitchell (1997)
- Experience **E**: a set of examples. Each example *x* is represented as a high-dimensional *feature* vector $x \in R^n$
  - E.g., an example could be an image represented by the pixels in the image

Task: Regression

- Map a feature vector to a continuous value $f: x \in R^d \rightarrow R$
- The goal is to accurately predict the target values



X: (user features, message features)
Y: the number of likes


Line: Large-scale information network embedding — [PDF] à partir de arxiv.org

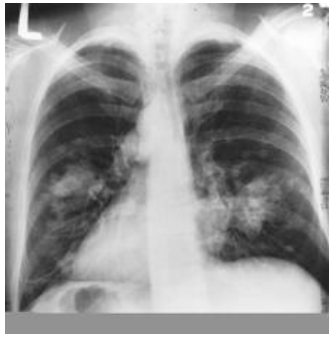
Auteurs	Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, Qiaozhu Mei
Date de publication	2015/5/18
Conférence	Proceedings of the 24th International Conference on World Wide Web
Pages	1067-1077
Éditeur	International World Wide Web Conferences Steering Committee
Description	This paper studies the problem of embedding very large information networks into low-dimensional vector spaces, which is useful in many tasks such as visualization, node classification, and link prediction. Most existing graph embedding methods do not scale for real world information networks which usually contain millions of nodes. In this paper, we propose a novel network embedding method called the``LINE,"which is suitable for arbitrary types of information networks: undirected, directed, and/or weighted. The method optimizes a carefully designed objective function that preserves both the local and global network structures. An edge-sampling algorithm is proposed that addresses the limitation of the classical stochastic gradient descent and improves both the effectiveness and the efficiency of the inference. Empirical experiments prove the effectiveness of the LINE on a variety of …
Nombre total de citations	Cité 502 fois

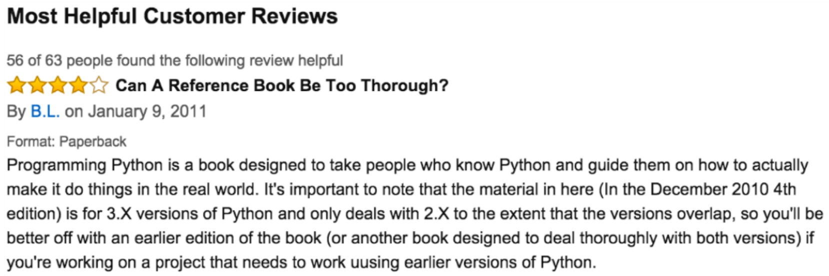
2015 2016 2017 2018

X: (author features, paper features)
Y: the number of citations

Task: Classification

- Assign an input real-valued vector x into K discrete classes $C = \{C_k\}_{k=1,\ldots,K}, i.e., f_\theta: x \in R^d \to C$





Most Helpful Customer Reviews

56 of 63 people found the following review helpful

Can A Reference Book Be Too Thorough?

By B.L. on January 9, 2011

Format: Paperback

Programming Python is a book designed to take people who know Python and guide them on how to actually make it do things in the real world. It's important to note that the material in here (In the December 2010 4th edition) is for 3.X versions of Python and only deals with 2.X to the extent that the versions overlap, so you'll be better off with an earlier edition of the book (or another book designed to deal thoroughly with both versions) if you're working on a project that needs to work uusing earlier versions of Python.

X: set of pixel intensities
Y: cancer present/cancer absent

X: reviews
Y: positive/neutral/negative

Task: Density Estimation

- Map a feature vector to a continuous value $p_{model}: x \in R^d \rightarrow R$, where $p_{model}(x)$ is the probability density function
  - Data imputation, estimate $p(x_i|x_{-i})$
  - Data generation

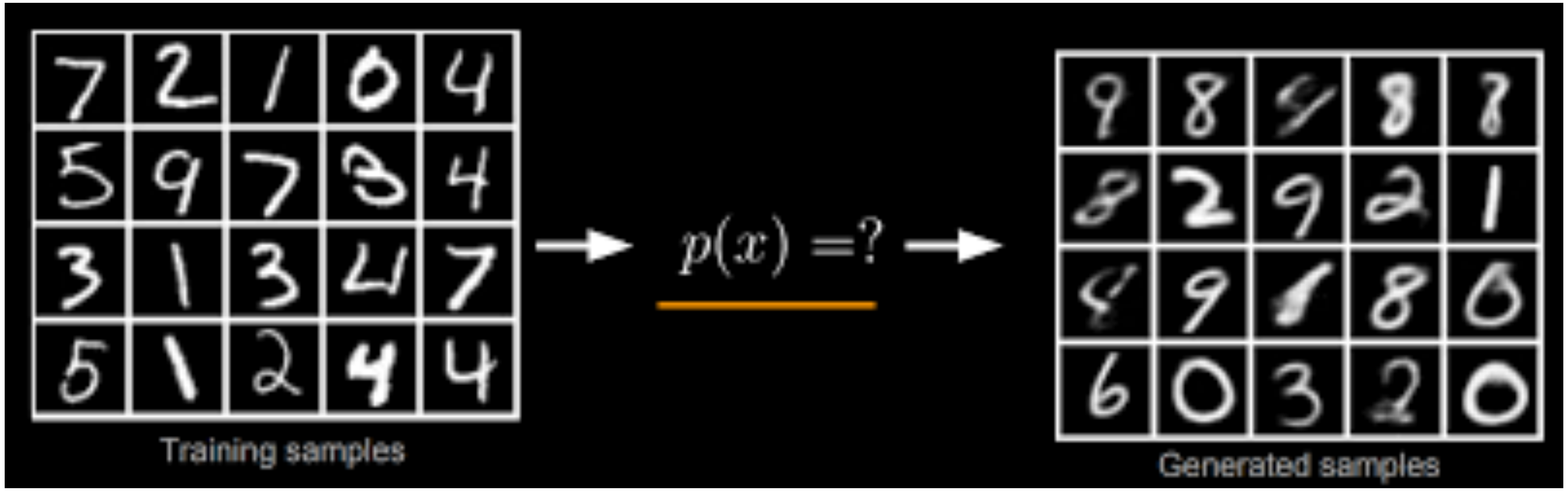


Image from Internet

Performance Measure: P

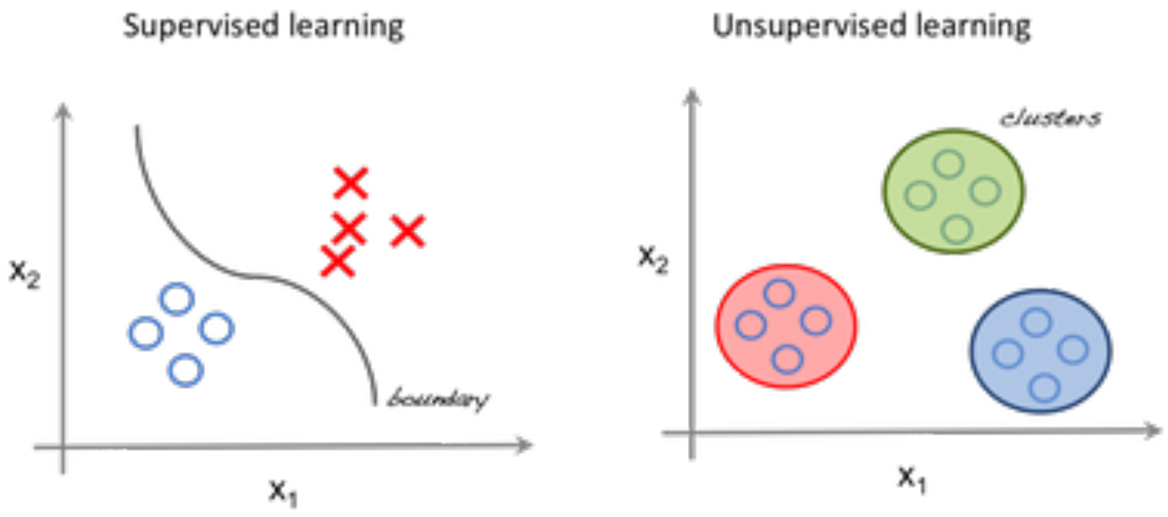
- A quantitative measure P must be designed to evaluate the abilities of a machine learning algorithm
  - Task specific
- E.g. Classification: Accuracy
  - The percentage of examples that are correctly classified
- The performance is usually evaluated on an unseen data set (test data set).

Experience: E

- Machine learning Algorithms:
  - Supervised
  - Unsupervised
- Supervised: each example is associated with a label or a target
  - E.g. classification or regression
- Unsupervised: no label or target is given
  - E.g., density estimation, clustering, dimension reduction
- Not covered: reinforcement learning
  - The experience are not fixed but dynamically generated by interacting with an environment

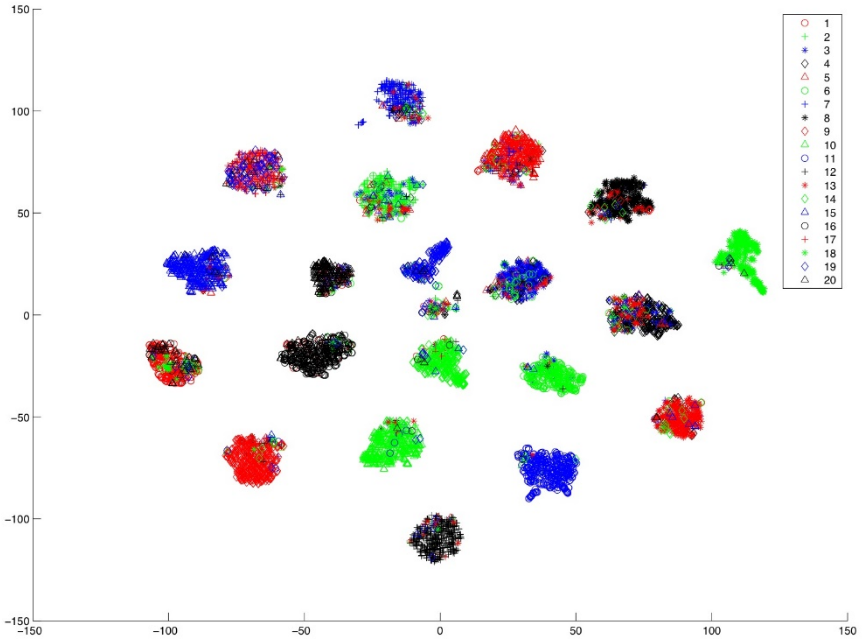
Supervised Learning v.s. Unsupervised Learning

- Supervised learning: labels are given to the algorithms
  - E.g., classification or regression
- Unsupervised learning: no supervision are provided
  - E.g., clustering, dimension reduction, data generation



Dimension Reduction

- Reduce high-dimensional to low-dimensional (e.g., 2D or 3D)
  - E.g. Map data with hundreds or thousands dimensions to 2D/3D.
  - PCA, ICA, t-SNE, LargeVis.



Example: Logistic Regression (K = 2)

- For binary classification, the posterior probability of class $\mathcal{C}_1$ can be written as sigmoid function

$$\mathrm{p}(\mathcal{C}_1|\boldsymbol{x}) = \frac{1}{1 + \exp(-\boldsymbol{x}^T\boldsymbol{w} - b)} = \sigma(\boldsymbol{x}^T\boldsymbol{w} + b)$$

- where $\boldsymbol{w}$ are the weights of the features, b is the bias term. The probability of the other class is defined as:

$$p(\mathcal{C}_2|\mathbf{x}) = 1 - p(\mathcal{C}_1|\mathbf{x}),$$

Maximum Likelihood for Logistic Regression

- We observed a training dataset $\{\mathbf{x}_n, t_n\},\ n = 1, .., N;\ t_n \in \{0, 1\}$.
- Maximize the probability of getting the label right, so the likelihood function takes form:

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}) = \prod_{n=1}^{N} \left[y_n^{t_n} (1 - y_n)^{1-t_n} \right], \quad y_n = \sigma(\boldsymbol{x}_n^T \boldsymbol{w})$$

Cross-Entropy Error Function

- Taking the negative log of the likelihood, we can define the cross-entropy error function (that we want to minimize):

$$E(\mathbf{w}) = -\ln p(\mathbf{t}|\mathbf{X}, \mathbf{w}) = -\sum_{n=1}^{N} \left[t_n \ln y_n + (1 - t_n) \ln(1 - y_n) \right] = \sum_{n=1}^{N} E_n.$$

- Here $E_n = -t_n \log y_n - (1 - t_n) \log(1 - y_n)$ is the cross entropy between the two binary distributions $\mathrm{P}_{\mathrm{data}} = (t_n, 1 - t_n)$ and $\mathrm{P}_{\mathrm{model}} = (y_n, 1 - y_n)$

Multi-class (K > 2) with Softmax Function

- Define a linear function for each class:

Class 1:		$\mathbf{x}^{\mathrm{T}}\mathbf{w}_{1}$
Class 2:		$\mathbf{x}^{\mathrm{T}}\mathbf{w}_{2}$
	….	
Class K:		$\mathbf{x}^{\mathrm{T}}\mathbf{w}_{K}$

- Normalize these scores with a softmax function

$$P(\mathcal{C}_k \mid \mathbf{x}) = \frac{\exp(\mathbf{x}^{\mathrm{T}}\mathbf{w}_{\boldsymbol{k}})}{\sum_{i=1}^{K} \exp(\mathbf{x}^{\mathrm{T}}\mathbf{w}_{\boldsymbol{i}})}$$

- which defines the probability of belonging to class $\mathcal{C}_k$

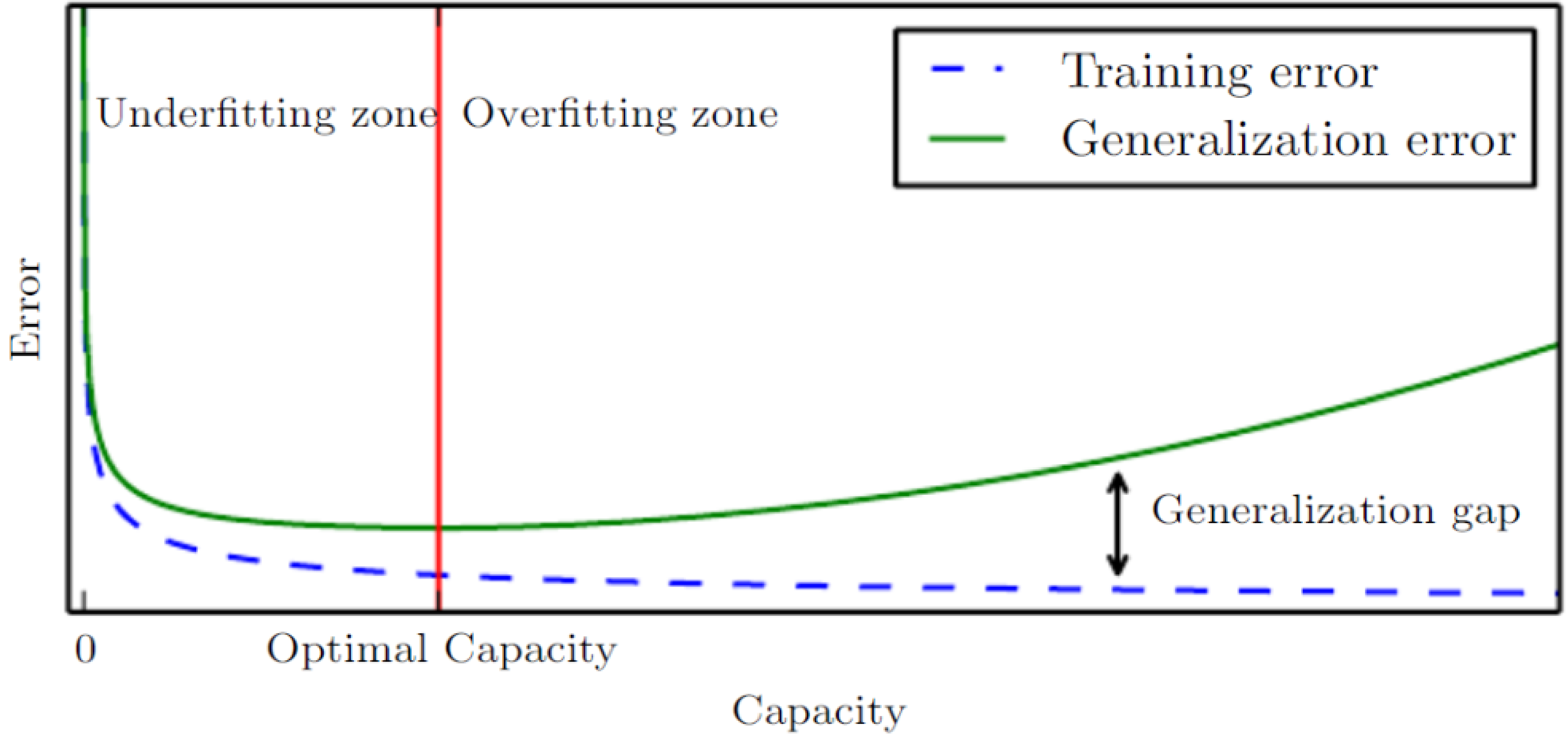
Model Capacity, Underfitting, and Overfitting

- The goal of machine learning model is to maximize the **generalization** ability
  - Perform well on previously unobserved inputs
- Training data => training error
- Test data => test error (generalization error)
- For linear regression:
  - Train the model by minimizing the train error
  - Evaluate the performance of the model according to the test error

Model Capacity, Underfitting, and Overfitting

- Model capacity: the ability to fit a variety of functions
  - Models with more parameters usually have larger capacity
- Underfitting: model is not able to obtain a sufficiently low error value on the training set
- Overfitting: perform wells on training data but not on the test data

Model Capacity v.s. Error



Regularization

- Techniques for avoid overfitting
  - Expressing preferences for different functions
- Regularized Logistic Regression

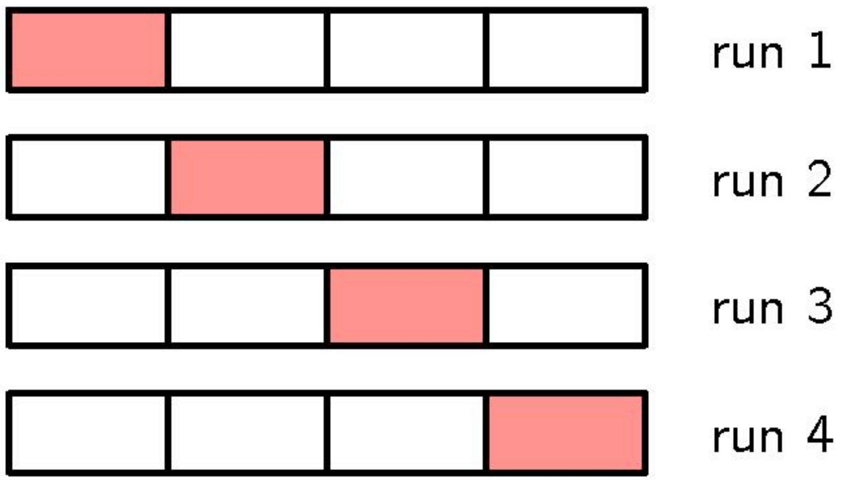
$$\mathrm{O} = -\log \mathrm{p}(\mathbf{T}|\mathbf{X}, \mathbf{w}) + \lambda||\boldsymbol{w}||_2^2$$

- This is also know as L2 regularization or weight decay

Cross Validation

- Divide the data set into three subsets
  - Training: used to learn the model parameters
  - Validation: used to select the model,hyper-parameters (e.g., regularization)
  - Test: evaluate the performance of the models
- K-fold cross validation
  - Use as much training data as possible

S fold cross-validation

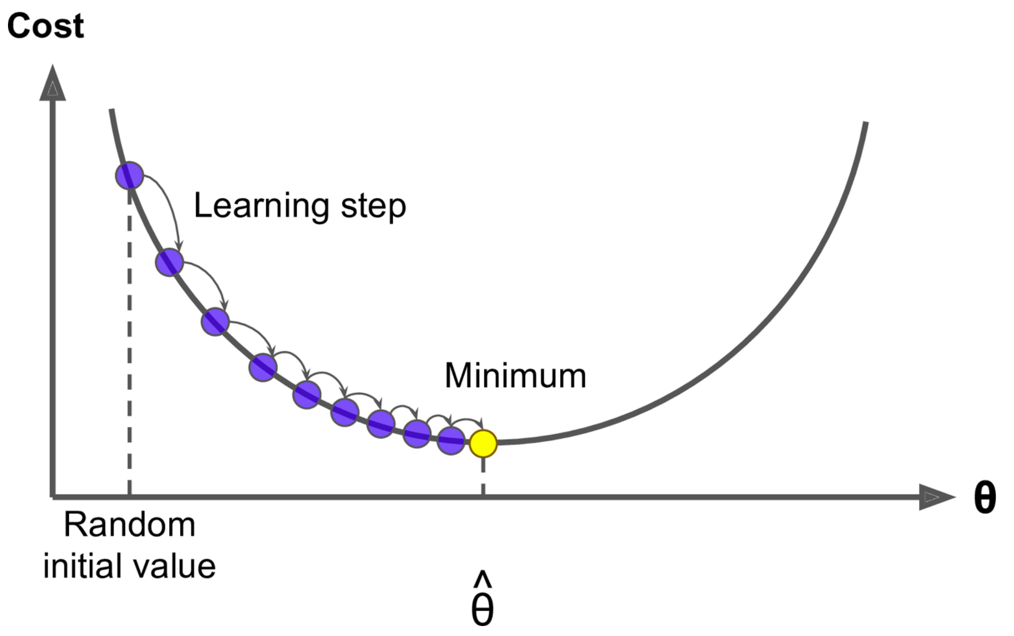


Gradient Descent

- Gradient Descent is an iterative optimization algorithm for finding the minimum of a function (e.g., the negative log likelihood)
- For a function F(x) at a point **a**, F(x) *decreases fastest* if we go in the direction of the negative gradient of **a**

$$\mathbf{a}_{n+1} = \mathbf{a}_n - \gamma \nabla F(\mathbf{a}_n)$$

When the gradient is zero, we arrive at the local minimum



Stochastic Gradient Descent

- For minimizing the cross-entropy error function w.r.t. the parameter w :

$$\nabla_w E(w) = \frac{1}{n}\sum_{i=1}^{n} \nabla_w E_n$$

- However n can be very large, which is too computational expensive
- **Stochastic** Gradient Descent: approximate the gradient with random samples

One sample: $\nabla_w E(w) \approx \nabla_w E_n$

A batch of samples: $\nabla_w E(w) \approx \frac{1}{B}\sum_{i=1}^{B} \nabla_w E_i$

Reading

- Deep Learning Book Chap. 2, 3, 5

Things to Do

- Register your presentation and course project groups
  - The two should be the same