VB035 - English 1
Textbook

Authors: ***Martin Dvořák, Kateřina Řepová, Ivana Tulajová***

All the texts in this textbook have been taken from ***Encyclopedia of Database Systems*** (Özsu, M. Tamer; Liu, Ling (Eds.) 2009. Approx. 4100 p. 60 illus. ISBN: 978-0-387-49616-0) and abridged.

Tento projekt je spolufinancován Evropským sociálním fondem
a státním rozpočtem České republiky.

EVROPSKÁ UNIE

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

OP Vzdělávání pro konkurenceschopnost

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Phonetics

Vowels

Long Vowels

iː	sheep	**ɑː**	farm	**uː**	coo	**ɔː**	horse	**ɜː**	bird

Short Vowels

ɪ	ship	**æ**	hat	**ʊ**	foot	**ɒ**	sock (UK)	**ʌ**	cup
e	head	**ə**	above	**ɚ**	mother (US)	**ɝ**	worm (US)		

Consonants

Voiced

b	book	**d**	day	**g**	give	**v**	very	**ð**	the
z	zoo	**ʒ**	vision	**dʒ**	jump	**l**	look	**r**	run
j	yes	**w**	we	**m**	moon	**n**	name	**ŋ**	sing

Voiceless

p	pen	**t**	town	**k**	cat	**f**	fish	**θ**	think
s	say	**ʃ**	she	**tʃ**	cheese				

Diphthongs

eɪ	day	**aɪ**	eye	**ɔɪ**	boy	**aʊ**	mouth	**əʊ**	nose (UK)
oʊ	nose (US)	**ɪə**	ear (UK)	**eə**	hair (UK)	**ʊə**	pure (UK)		

Other Symbols

h	[hænd] hand	**ɒ̃**	['kwæs.ɒ̃] croissant (UK)	**i**	['hæp.i] happy
ţ	['bʌţ.ɚ] butter (US)	**u**	[ˌɪn.flu'en.zə] influenza	**!**	['lɪt.!] little

^{ə}**l**, ^{ə}**m**, ^{ə}**n** can be pronounced either: **əl** or **!**, etc.: ['leɪ.b^{ə}l] = ['leɪ.bəl] = ['leɪ.b!]

^{r} linking r is pronounced only before a vowel in British English: [fɔː^{r}] four : [fɔːræp.!z] four apples

' main stress [ˌek.spek'teɪ.ʃ^{ə}n] expectation

ˌ secondary stress [ˌriː'tel] retell

. syllable division ['sɪs.təm] system

Cambridge Dictionary Online [Internet]. c2010 [cited 2010 Feb 24]. Phonetics. Available from: <http://dictionary.cambridge.org/help/phonetics.asp>.

Access Control

ELENA FERRARI
University of Insubria, Varese, Italy

Definition

Access control deals with preventing unauthorized operations on the managed data. Access control is usually performed against a set of *authorizations* stated by Security Administrators (SAs) or users according to the *access control policies* of the organization. Authorizations are then processed by the *access control mechanism* (or *reference monitor)* to decide whether each access request can be authorized or should be denied.

Historical Background

Access control models for DBMSs have been greatly influenced by the models developed for the protection of operating system resources. For instance, the model proposed by Lampson [16] is also known as the *access matrix* model since authorizations are represented as a matrix.

Then, in the 1970s, as research in relational databases began, attention was directed towards access control issues. Around the same time, some early work on multilevel secure database management systems (MLS/DBMSs) was reported. However, it was only after the Air Force Summer Study in 1982 [1] that developments on MLS/DBMSs began. In the 1990s, numerous other developments were made to meet the access control requirements of new applications and environments, such as the World Wide Web, data warehouses, data mining systems, multimedia systems, sensor systems, workflow management systems and collaborative systems. Recently, there have been numerous developments in access control, mainly driven by developments in web data management. For example, standards such as XML (eXtensible Markup Language) and RDF (Resource Description Framework) require proper access control mechanisms [7]. Also, web services and the semantic web are becoming extremely popular and therefore research is currently carried out to address the related access control issues [13]. Access control is currently being examined for new application areas, such as knowledge management [4], data outsourcing, GIS [10], peer-to-peer computing and stream data management [8]. For example, in the case of knowledge management applications, it is important to protect the intellectual property of an organization, whereas when data are outsourced, it is necessary to allow the owner to enforce its access control policies, even if data are managed by a third party.

Foundations

The basic building block on which access control relies is a set of *authorizations:* which state, who can access which resource, and under which mode. Authorizations are specified according to a set of *access control policies,* which define the high-level rules according to which access control must occur. In its basic form, an authorization is, in general, specified on the basis of three components (s,o,p), and specifies that subject s is authorized to exercise privilege p on object o. The three main components of an authorization have the following meaning:

- *Authorization subjects:* They are the ‘‘active’’ entities in the system to which authorizations are granted. Subjects can be further classified into the following, not mutually exclusive, categories: *users,* that is, single individuals connecting to the system; *groups,* that is, sets of users; roles, that is, named collection of privileges needed to perform specific activities within the system; and *processes,* executing programs on behalf of users.
- *Authorization objects:* They are the ‘‘passive’’ components (i.e., resources) of the system to which protection from unauthorized access should be given. The set of objects to be protected clearly depends on the considered environment. For instance, files and directories are examples of objects of an operating system environment, whereas in a relational DBMS, examples of resources to be protected are relations, views and attributes. Authorizations can be specified at different granularity levels, that is, on a whole object or only on some of its components. This is a useful feature when an object (e.g., a relation) contains information (e.g., tuples) of different sensitivity levels and therefore requires a differentiated protection.
- *Authorization privileges:* They state the types of operations (or access modes) that a subject can exercise on the objects in the system. As for objects, the set of privileges also depends on the resources to be protected. For instance, read, write, and execute privileges are typical of an operating system environment, whereas in a relational DBMS privileges refer to SQL commands (e.g., select, insert, update, delete). Moreover, new environments such as digital libraries are characterized by new access modes, for instance, usage or copying access rights.

A basic distinction when dealing with access control is between *discretionary* and *mandatory* access control. Discretionary access control (DAC) governs the access of subjects to objects on the basis of subjects’ identity and a set of explicitly specified authorizations that specify, for each subject, the set of objects that he/she can access in the system and the allowed access modes. When an access request is submitted to the system, the access control mechanism verifies whether or not the access can be authorized according to the specified authorizations. The system is discretionary in the sense that a subject, by proper configuring the set of authorizations, is both able to enforce various access control requirements and to dynamically change them when needed (simply by updating the authorization state). In contrast, mandatory access control (MAC) specifies the access that subjects can exercise on the objects in the system, on the basis of subjects and objects security classification [14].

When mandatory access control is enforced, authorizations are implicitly specified, by assigning subjects and objects proper security classes. The decision on whether or not to grant access depends on the access mode and the relation existing between the classification of the subject requesting the access and that of the requested object. In addition to DAC and MAC, role-based access control (RBAC) has been more recently proposed [12]. In RBAC, permissions are associated with roles, instead of with users, and users acquire permissions through their membership to roles.

Key Applications

Access control techniques are applied in almost all environments that need to grant controlled access to their resources, including, but not limited, to the following:

DBMSs, Data Stream Management Systems, Operating Systems, Workflow Management Systems, Digital Libraries, GIS, Multimedia DBMSs, E-commerce services, Publish-subscribe systems, Data warehouses.

Future Directions

Although access control is a mature area with consolidated results, the evolution of DBMSs and the requirements of new applications and environments pose new challenges to the research community.

Social networks. Web-based social networks (WBSNs) are online communities where participants can establish relationships and share resources across the web with other users. In recent years, several WBSNs have been adopting semantic web technologies, such as FOAF, for representing users' data and relationships, making it possible to enforce information interchange across multiple WBSNs. So far, this issue has been mainly addressed in a very simple way, by some of the available WBSNs, by only allowing users to state whether specific information (e.g., personal data and resources) should be public or accessible only by the users with whom the owner of such information has a direct relationship.

- *Data streams.* In many applications, such as telecommunication, battle field monitoring, network monitoring, financial monitoring, sensor networks, data arrive in the form of high speed data streams. These data typically contain sensitive information (e.g., health information, credit card numbers) and thus unauthorized access should be avoided.
- *Semantic web.* The web is now evolving into the semantic web. The semantic web [5] is a web that is intelligent with machine-readable web pages. The major components of the semantic web include web infrastructures, web databases and services, ontology management and information integration. If the semantic web is to be effective, it is necessary to ensure that the information on the web is protected from unauthorized access and malicious modifications. Also, it must be ensured that individual's privacy is maintained. To cope with these issues, it is necessary to secure all the semantic web related technologies, such as XML, RDF, Agents, Databases, web services, and Ontologies and ensure the secure interoperation of all these technologies [13].

(Abridged)

Recommended Reading

1. Air Force Studies Board, Committee on Multilevel Data Management Security. Multilevel data management security. National Research Council, 1983.

2. Berners-Lee T. et al. The semantic web. Scientific American, 2001.

3. Bertino E., and Sandhu R.S. Database security: concepts, approaches, and challenges. IEEE Trans. Dependable and Secure Computing, 2(1):2–19, 2005.

4. Bertino E., Khan L.R., Sandhu R.S., and Thuraisingham B.M. Secure knowledge management: confidentiality, trust, and privacy. IEEE Trans. Syst. Man Cybern. A, 36(3):429–438, 2006.

5. Carminati B., Ferrari E., and Perego A. Enforcing access control in web-based social networks. ACM trans. Inf. Syst. Secur., to appear.

6. Carminati B., Ferrari E., and Tan K.L. A framework to enforce access control over Data Streams. ACM Trans. Inf. Syst. Secur., to appear.

7. Carminati B., Ferrari E., and Thuraisingham B.M. Access control for web data: models and policy languages. Ann. Telecomm., 61 (3–4):245–266, 2006.

8. Carminati B., Ferrari E., and Bertino E. Securing XML data in third party distribution systems. In Proc. of the ACMFourteenth Conference on Information and Knowledge Management (CIKM05), Bremen, Germany, 2005.

9. Castano S., Fugini M.G., Martella G., and Samarati P. Database security. Addison Wesley, 1995.

10. Damiani M.L. and Bertino E. Access control systems for geo-spatial data and applications. In Modelling and management of geographical data over distributed architectures, A. Belussi, B. Catania, E. Clementini, E. Ferrari (eds.). Springer, 2007.

11. Fagin R. On an authorization mechanism. ACMTrans. Database Syst., 3(3):310–319, 1978.

12. Ferraiolo D.F., Sandhu R.S., Gavrila S.I., Kuhn D.R., and Chandramouli R. Proposed NIST standard for role-based access control. ACM Trans. Inf. Syst. Secur., 4(3):224–274, 2001.

13. Ferrari E. and Thuraisingham B.M. Security and privacy for web databases and services. In Advances in Database Technology, Proc. 9th Int. Conf. on Extending Database Technology, 2004, pp. 17–28.

14. Ferrari E. and Thuraisingham B.M. Secure database systems. In O. Diaz, M. Piattini (eds.). Advanced databases: technology and design. Artech House, 2000.

15. Griffiths P.P. and Wade B.W. An authorization mechanism for a relational database system. ACM Trans. Database Syst., 1 (3):242–255, 1976.

16. Lampson B.W. Protection. Fifth Princeton Symposium on Information Science and Systems, Reprinted in ACM Oper. Sys. Rev., 8(1):18–24, 1974.

Answer the following questions:

1) How would you define *access control*?
2) What is *reference monitor*?
3) What does the abbreviation of DBMS stand for?
4) The article mentions terms such as *data warehouses, data mining systems, workflow management systems,* and *semantic Web*. Do you know what these are?
5) What is *data outsourcing*?
6) Why is access control needed in the area of *knowledge management*?
7) How would you describe *authorizations*?
8) What is the difference between *discretionary* and *mandatory* access control?
9) What is the use of access control in *data streams*?

Match the following terms with their definitions:

1) access control policy
2) authorization subjects
3) authorization objects
4) authorization privileges
5) WBSN

a) entities in the system that authorizations are granted to
b) an online community where participants can establish relationships across the Web
c) entities that are subject to protection
d) a set of rules specifying access control
e) access modes

Mark the following statements as *true* or *false:*

1) The systems developed for the protection of operating system resources greatly influenced the access control models utilized for DBMSs.
2) *Mandatory access control* governs the access of subjects to objects on the basis of subjects' identity and a set of explicitly specified authorizations.
3) *Discretionary access control* is based on assigning subjects and objects proper security classes.
4) *Semantic Web* represents a term used to refer to the technology that is concerned with the form of Web pages rather than their contents.

Vocabulary

attribute ['æt.rɪ.bjuːt] – atribut, rys

authorization [ˌɔː.θᵊr.aɪ'zeɪ.ʃᵊn] US [ˌɑː.θɚ–] – autorizace

building block ['bɪl.dɪŋ] [blɒk] US [blɑːk] – stavební kámen

classify st ['klæs.ɪ.faɪ] – klasifikovat něco

component [kəm'pəʊ.nənt] US [–'poʊ–] – komponent, komponenta

consolidate st [kən'sɒl.ɪ.deɪt] US [–'sɑː.lɪ–] – posílit něco

control [kən'trəʊl] US [–'troʊl] – ovládat něco; ovládání

carry st out ['kær.i] US ['ker–] [aʊt] – uskutečnit něco, provádět něco

data ['deɪ.tə] US [–t̬ ə] – data (data is/are)

discretionary [dɪ'skreʃ.ᵊn.ᵊr.i] US [–er–] – přenechaný volnému uvážení

dynamic [da ɪ'næm.ɪk] – dynamický

explicit [ɪk'splɪs.ɪt] – explicitní, jasně řečený

granularity ['græn.ju.læ.rə.tɪ] – zrnitost, stupně členění

implicit [ɪm'plɪs.ɪt] – implicitní, naznačený

interoperation [in.tə.o.pə'reɪ.ʃᵊn] US [–t̬ ɚ–] – vzájemná spolupráce

issue ['ɪʃ.uː] US ['ɪs.juː] – záležitost, problém

maintenance ['meɪn.tɪ.nən*t*s] – údržba

major ['meɪ.dʒəʳ] US [–dʒɚ] – hlavní

malicious [mə'lɪʃ.əs] – zlovolný, zákeřný

mandatory ['mæn.də.tri] US [–tɔːr.i] – povinný

matrix ['meɪ.trɪks] – matice

mature [mə'tjʊəʳ] US [–'tʊr] – zralý, dospělý

mode [məʊd] US [moʊd] – mód, způsob, režim

model ['mɒd.ᵊl] US ['mɑː.dᵊl] – model; modelovat

modification [ˌmɒd.ɪ.fɪ'keɪ.ʃᵊn] US [ˌmɑː.dɪ–] – modifikace, změna

mutual ['mjuː.tʃu.əl] – vzájemný

numerous ['njuː.mə.rəs] US ['nuː–] – četný, početný

object ['ɒb.dʒɪkt] US ['ɑːb–] – objekt

pose (challenges) [pəʊz] US [poʊz] ['tʃæl.ɪndʒɪz] – představovat, vytvářet (výzvy)

privilege ['prɪv.ᵊl.ɪdʒ] – privilegium, právo

protection [prə'tek.ʃᵊn] – ochrana

privacy ['prɪv.ə.si] US ['praɪ.və–] – soukromí

reference to st ['ref.ᵊr.ᵊn*t*s] US [–ɚ–] – odkaz na něco

relational [rɪ'leɪ.ʃᵊnᵊl] – relační

relational database ['deɪ.tə.beɪs] US [–t̬ ə–] – relační databáze

sensitivity [ˌsent .sɪ'tɪv.ɪ.ti] US [–sə'tɪv.ə.t̬ i] – citlivost

set [set] – množina

third party [θɜːd] US [θɝːd] ['pɑː.ti] US ['pɑrː.ti] – třetí strana

therefore ['ðeə.fɔːʳ] US ['ðer.fɔːr] – tudíž

tuple [tjuːpl, tʌpl] – n–tice

to access st ['æk.ses] – přistupovat k něčemu

to address an issue [ə'dres] US ['æd.res] ['ɪʃ.uː] US ['ɪs.juː] – vzabývat se problémem

to assign st to st [ə'saɪn] – přiřadit něco něčemu

to avoid st [ə'vɔɪd] – vyhnout se něčemu

to be characterized by st ['kær.ɪk.tə.raɪzd] US ['ker.ɪk.tɚ.aɪzd] – být charakterizován něčím

to configure st [kən'fɪg.əʳ] US [kən'fɪg.ɚ] – (na)konfigurovat něco

to consider st [kən'sɪd.əʳ] US [–ɚ] – zvažovat něco

to consider 1 to be 2 – považovat 1 za 2

to cope with st [kəʊp] US [koʊp] – vypořádat se s něčím

to deal with st [dɪəl] – zabývat se něčím

to enforce st [ɪn'fɔːs] US [–'fɔːrs] – prosadit, prosadit si

to ensure st [ɪn'ʃɔːʳ] US [–'ʃʊr] – zajistit něco

to execute st ['ek.sɪ.kjuːt] 🔊 – provést něco, vykonat

to exercise st on st ['ek.sə.saɪz] US [–sɚ–] 🔊 – uplatňovat něco na něčem; (e.g., exercise a strong influence on st)

to maintain st [meɪn'teɪn] 🔊 – udržovat něco

to have access to st [hæv] 🔊 ['æk.ses] 🔊 – mít přístup k něčemu

to object to st [əb'dʒekt] 🔊 – mít námitky vůči něčemu

to propose st [prə'pəʊz] US [–'poʊz] 🔊 – nabídnout něco, navrhnout

to submit st to st [səb'mɪt] 🔊 – předat něco něčemu

to verify st ['ver.ɪ.faɪ] 🔊 – ověřit něco

whereas [weə'ræz] US [wer'æz] 🔊 – kdežto

Phrases

As for ... – Co se týče ...

Both ... and ... – Jak ..., tak ...

In contrast, ... ['kɒn.trɑːst] US ['kɑːn.træst] 🔊 – Naopak, ...

Audio

LIE LU[1], ALAN HANJALIC[2]
[1]Microsoft Research Asia, Beijing, China
[2]Delft University of Technology, Delft,
The Netherlands

Definition

Audio refers to audible sound – the sound perceivable by the human hearing system, or the sound of a frequency belonging to the *audible frequency range* (20-20,000 Hz). Audio can be generated from various sources and perceived as speech, music, voices, noise, or any combinations of these. The perception of an audible sound starts by the sound pressure waves hitting the eardrum of the outer ear. The generated vibrations are transmitted to the cochlea of the inner ear to produce mechanical displacements along the basilar membrane. These displacements are further transduced into electrical activity along the auditory nerve fibers, and finally ''analyzed'' and ''understood'' in the central auditory system [4,7].

Historical Background

The step from the fundamental definition of audio towards the concept of *audio signal* can be seen as a step towards the birth of the modern consumer electronics. An audio signal is a signal that contains audio information in the audible frequency range. The technology for generating, processing, recording, broadcasting and retrieving audio signals, first *analog* and later on *digital* ones, has rapidly grown for over a century, from the pioneering radio broadcasting and telephony systems to advanced mobile communication infrastructures, music players, speech recognition and synthesis tools, and audio content analysis, indexing and retrieval solutions. This growth may have been initiated by the research in the field of signal processing, but it has been maintained and has continuously gained in strength through an extensive interdisciplinary effort involving signal processing, information theory, human-computer interaction, psychoacoustics, psychology, natural language processing, network and wireless technology, and information retrieval.

Foundations

Digital Audio

An audio signal is an analog signal, which can be represented as a one-dimensional function x(t), where t is a continuous variable representing time. To facilitate storage and processing of such signals in computers, they can be transformed into digital signals by *sampling* and *quantization*.

Sampling is the process in which one audio signal value *(sample)* is taken for each time interval *(sampling period) T*. This results in a *discrete audio* signal *x(n) = x(nT)*, where *n* is a numeric sequence. The sampling period T determines the *sampling frequency* that can be defined as *f = 1/T*. Typical sampling frequencies of digital audio are 8, 16, 32, 48, 11.025, 22.05, and 44.1 kHz (Hz represents the number of samples per second). Based on the Nyquist-Shannon sampling theorem, the sampling frequency must be at least 2 times larger than the band limit of the audio signal in order to be able to reconstruct the original analog signal back from its discrete representation. In the next step, each sample in the discrete audio signal is *quantized* with a bit resolution, which makes each sample be represented by a fixed limited number of bits. Common bit resolution is 8-bit or 16-bit per sample. The overall result is a digital representation of the original audio signal, that is referred to as *digital audio signal* or, if it is just considered as a set of bits, for instance for the purpose of storage and compression, as *digital audio data.*

Audio Coding and Compression

The digitization process described above leads to the basic standard of digital audio representation or *coding* named *Pulse Code Modulation (PCM)*, which was developed in 1930-1940s. PCM is also the standard digital audio format in computers and Compact Disc (CD). PCM can be integrated into a widely used WAV format, which consists of the digital audio data and a header specifying the sampling frequency, bits per sample, and the number of audio channels.

As a basic audio coding format, PCM keeps all samples obtained from the original audio signal and all bits representing the samples. This format is therefore also referred to as *raw* or *uncompressed.* While it preserves all the information contained in the original analog signal, it is also rather expensive to store. For example, a one-hour *stereo* (A Cambridge Dictionary definition of stereo: a way of recording or playing sound so that it is separated into two signals and produces more natural sound) audio signal with 44.1 kHz sampling rate and 16 bits per sample requires 635MB of digital storage space. To save storage in computers and improve the efficiency of audio transmission, processing and management, *compression* theory and algorithms can be applied to decrease the size of a digital audio signal while still keeping the quality of the signal and communicated information at the acceptable level.

Starting with the variants of PCM, such as *Differential Pulse Code Modulation* (DPCM) and *Adaptive*

Differential Pulse Code Modulation (ADPCM), a large number of audio compression approaches have been developed [5]. Some most commonly used approaches include MP3/ACC defined in the MPEG-1/2 standard [2,3], Windows Media Audio (WMA) developed by Microsoft, and RealAudio (RA) developed by RealNetworks. These approaches typically lead to a compressed audio signal being about 1/5 to 1/10 of the size of the PCM format.

Audio Content Analysis

Audio content analysis aims at extracting descriptors or metadata related to audio content and allowing content-based search, retrieval, management and other user actions performed on audio data. The research in the field of audio content analysis has built on the synergy of many scientific disciplines, such as signal processing, pattern recognition, machine learning, information retrieval, and information theory, and has been conducted in three main directions, namely *audio representation, audio segmentation,* and *audio classification.*

Audio representation refers to the extraction of audio signal properties, or features, that are representative of the audio signal composition (both in temporal and spectral domain) and audio signal behavior over time. The extracted features then serve as input into audio segmentation and audio classification. Audio segmentation aims at automatically revealing semantically meaningful temporal segments in an audio signal, which can then be grouped together (using e.g. a *clustering* algorithm) to facilitate search and browsing. Finally, an audio classification algorithm classifies a piece of audio signal into a predefined semantic class, and assigns the corresponding label (e.g., ‘‘applause,’’ ‘‘action,’’ ‘‘highlight,’’ ‘‘music’’) to it for the purpose of text-based search and retrieval.

Audio Retrieval

Audio retrieval aims at retrieving sound samples from a large corpus based on their relation to an input query. Here, the query can be of different types and the expected results may vary depending on the application context. For example, in the *content-based retrieval* scenario, a user may use the text term ‘‘applause’’ to search for the audio clips containing the audio effect ‘‘applause.’’ Clearly, the results obtained from audio classification can help annotate the corresponding audio samples, audio segments or audio tracks, and thus facilitate this search and retrieval strategy. However, audio retrieval can also be done by using an audio data stream as a query, i.e., by performing *query-by-example* [6]. For instance, one could aim at retrieving a song and all its variants by simply singing or humming its melody line.

In another retrieval scenario, the user may want to retrieve the exact match to the query or some information related to it. This typically falls into the application domain of *audio fingerprinting* [1]. An audio fingerprint is a highly compact feature-based representation of an audio signal enabling extremely fast search for a match between the signal and a large scale audio database for the purpose of audio signal identification.

(Abridged)

Recommended Reading

1. Haitsma J. and Kalker T. A highly robust audio fingerprinting system with an efficient search strategy. J. New Music Res., 32 (2):211–221, 2003.

2. ISO/IEC 11172-3:1993. Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 3: Audio, 1993.

3. ISO/IEC 13818-3:1998. Information technology – Generic coding of moving pictures and associated audio information – Part 3: Audio, 1998.

4. Pickles J.O. An Introduction to the Physiology of Hearing. Academic Press, London, UK, 1988.

5. Spanias A., Painter T., and Atti V. Audio Signal Processing and Coding. Wiley, NJ, 2007.

6. Wold E., Blum T., and Wheaton J. Content-based classification, search and retrieval of audio. IEEEMultimed., 3(3):27–36, 1996.

7. Yang X., Wang K., and Shamma S.A.Auditory representations of acoustic signals. IEEE Trans. Inform. Theory, 38:824–839, 1992.

Answer the following questions:

1) Describe what the article refers to as *audible frequency range.*
2) Describe the mechanism of sound perception.
3) What is *sampling*?
4) How does *quantization* work?
5) What is *WAV format* and what does it consist of? Why is it referred to as “raw”?
6) What is the purpose of *audio content analysis?*
7) Describe what the article refers to as *audio retrieval.* What is its purpose?
8) The text mentions *query-by-example.* What is it?

Match the following terms with their definitions:

1) PCM
2) WMA
3) metadata
4) audio representation
5) audio segmentation
6) audio classification

a) a process of revealing semantically meaningful elements in an audio signal
b) information about data
c) a process of assigning a piece of audio signal to a pre-defined semantic class and assigning a label to it
d) a process of extracting audio signal properties
e) a standard digital audio format in computers and compact discs
f) an audio compression format

Mark the following statements as *true* or *false:*

1) During sampling, the sampling period T determines the sampling frequency that can be defined as $f = T$.
2) The *WAV format* preserves all the information contained in the original analog signal.
3) The *query-by-example* consists in searching for an audio element by a keyword assigned to the element.
4) *Audio fingerprinting* is based on searching for similarities between a query and an audio element.
5) Audio fingerprinting represents one of the slowest search methods.

Vocabulary

algorithm ['æl.gə.rɪ.ðᵊm] 🔊 – algoritmus

analog ['æn.ə.lɒg] US [–lɑːg] 🔊 – analogový

analysis (pl. analyses) [ə'næl.ə.sɪs] 🔊, pl. [ə'næl.ə.siːz] 🔊 – analýza

analyst ['æn.ə.lɪst] 🔊 – analytik

audible ['ɔː.dɪ.bl] US ['ɑː–] 🔊 – slyšitelný

audio ['ɔː.di.əʊ] US ['ɑː.di.oʊ] 🔊 – audio

auditory ['ɔː.dɪ.tri] US ['ɑː–] 🔊 – sluchový

band [bænd] 🔊 – pásmo

cochlea ['kɒk.li.ə] US ['kɑːk–] 🔊 🔊 – kochlea, hlemýžďovitá část ušního labyrintu

concept ['kɒn.sept] US ['kɑːn–] 🔊 – koncept, pojetí

consumer [kən'sjuː.məʳ] US [–'suː.mɚ] 🔊 – spotřebitel

continuous [kən'tɪn.ju.əs] 🔊 – neustálý

corpus ['kɔː.pəs] US ['kɔːr–] 🔊 – korpus

decrease ['diː.kriːs] 🔊 🔊 – snížení

dimension [ˌda ɪ'men.*t*ʃᵊn] 🔊 – rozměr, dimenze

discrete [dɪ'skriːt] 🔊 – nespojitý, rozpojený

displace [dɪ'spleɪs] 🔊 – přemístit, přesunout, pohnout

domain [dəʊ'meɪn] US [doʊ–] 🔊 – doména

eardrum ['ɪə.drʌm] US ['ɪr–] 🔊 – ušní bubínek

efficiency [ɪ'fɪʃ.ᵊn*t*.si] 🔊 – efektivita, účinnost

electrical [ɪ'lek.trɪ.kᵊl] – elektrický

fiber ['faɪ.bəʳ] US [–bɚ] 🔊 – vlákno

function ['fʌŋk.ʃᵊn] 🔊 – funkce

fundamental [ˌfʌn.də'men.tᵊl] US [–t̬ ᵊl] 🔊 – základní

generate st ['dʒen.ᵊr.eɪt] US [–ɚ–] 🔊 – generovat něco, vytvářet

header ['hed.əʳ] US [–ɚ] 🔊 – hlavička, záhlaví

inner ear ['ɪn.əʳ] US[–ɚ] 🔊 [ɪəʳ] US [ɪr] 🔊 – střední ucho

interdisciplinary [ˌɪn.tə'dɪs.ɪ.plɪ.nᵊr.i] US [–t̬ ɚ'dɪs.ə.plɪ.ner–] 🔊 – mezioborový

mechanical [mə'kæn.ɪ.kᵊl] 🔊 – mechanický

numeric [njuː'mer.ɪ.k] US [nuː–] 🔊 – numerický

one-dimensional [ˌwʌn.d*a*ɪ'men*t*.ʃᵊn.ᵊl] – jednorozměrný

outer ear ['aʊ.təʳ] US [–t̬ ɚ] 🔊 [ɪəʳ] US [ɪr] 🔊 – vnější ucho

overall [ˌəʊ.və'rɔːl] US [ˌoʊ.vɚ'ɑːl] 🔊 – celkový

perceive [pə'siːv] US [pɚ–] 🔊 – vnímat

perceivable [pə'siːvə.bl] US [pɚ–] 🔊 – vnímatelný

perception [pə'sep.ʃᵊn] US [pɚ–] 🔊 – vnímání

pre-defined [priː.dɪ'faɪnd] – předem definovaný

pressure ['preʃ.əʳ] US [–ɚ] 🔊 – tlak

properties ['prɒp.ə.tɪːs] US ['prɑː.pɚ.t̬ ɪːs] – vlastnosti, rysy

psychology [saɪ'kɒl.ə.dʒi] US [–'kɑː.lə–] 🔊 – psychologie

quantization [kwɒn.tai.'zei.ʃᵊn] US [kwɑːn.tai.'zei.ʃᵊn] 🔊 – kvantování

query ['kwɪə.ri] US ['kwɪr.i] 🔊 – dotaz

resolution [ˌrez.ə'luː.ʃᵊn] 🔊 – rozlišení

retrieval [rɪ'triː.vᵊl] 🔊 – vyzvednutí, získávání, hledání

sample ['sɑːm.pl] US ['sæm–] 🔊 – vzorek; vzorkovat

sampling frequency ['sɑːm.plɪŋ] US ['sæm–] 🔊 ['friː.kwən*t*.si] 🔊 – vzorkovací frekvence

sampling rate [reɪt] 🔊 – vzorkovací frekvence

scenario [sɪ'nɑː.ri.əʊ] US [sə'ner.i.oʊ] 🔊 – scénář

signal ['sɪg.nəl] 🔊 – signál

speech recognition [spiːtʃ] 🔊 [ˌrek.əg'nɪʃ.ᵊn] 🔊 – rozpoznávání řeči

stereo ['ster.i.əʊ] US [–oʊ] 🔊 – stereo

storage ['stɔː.rɪdʒ] US ['stɔːr.ɪdʒ] 🔊 – uchovávání, ukládání

synthesis (pl. syntheses) ['sɪn.θə.sɪs] 🔊 – syntéza

theory ['θɪə.ri] US ['θɪr.i] 🔊 – teorie

transmission [trænz'mɪʃ.ᵊn] US [træns–] 🔊 🔊 – přenos

variable ['veə.ri.ə.bl] US ['ver.i–] 🔊 – proměnná

WAV format [weiv] Ⓤⓢ [weiv] ['fɔː.mæt] Ⓤⓢ ['fɔːr-] 🔊 – formát WAV

to aim at st [eɪm] 🔊 – zaměřovat se na něco, mít za cíl něco

to analyze ['æn.əl.aɪz] 🔊 – analyzovat

to conduct research [kən'dʌkt] 🔊 [rɪ'sɜːtʃ] Ⓤⓢ ['riː.sɝːtʃ] 🔊 🔊 – dělat, podnikat výzkum

to decrease st [dɪ'kriːs] Ⓤⓢ ['diː.kriːs] 🔊 🔊 – snížit něco

to digitize st ['dɪdʒ.ɪ.taɪz] 🔊 – digitalizovat

to facilitate st [fə'sɪl.ɪ.teɪt] 🔊 – zjednodušit něco

to fall into st [fɔːl] Ⓤⓢ [fɑːl] 🔊 – spadat do něčeho

to gain in strength [geɪn] 🔊 [streŋθ] 🔊 🔊 – získávat na síle

to hum [hʌm] 🔊 – broukat, pobrukovat

to improve st [ɪm'pruːv] 🔊 – zlepšit něco

to initiate st [ɪ'nɪʃ.i.eɪt] 🔊 – iniciovat něco, začít

to preserve st [prɪ'zɜːv] Ⓤⓢ [-'zɝːv] 🔊 – zachovat něco

to process st ['prəʊ.ses] Ⓤⓢ ['prɑː-] 🔊 🔊 – zpracovat něco

to quantize st ['kwɒn.taiz] Ⓤⓢ ['kwɑːn.taiz] 🔊 – kvantovat

to relate to st [rɪ'leɪt] 🔊 – vztahovat se na něco

to retrieve [rɪ'triːv] 🔊 – vyzvednout, získat

to reveal st [rɪ'viːl] 🔊 – odhalit něco

to transduce st [trænz'djuː.s] Ⓤⓢ [trɑːn*t*s'duː.s] 🔊 – přetvořit něco, přetransformovat (energii)

to transmit st [trænz'mɪt] Ⓤⓢ [træns-] 🔊 🔊 – přenášet něco, vysílat

to vibrate [vaɪ'breɪt] 🔊 – vibrovat

vibration [vaɪ'breɪ.ʃən] 🔊 – vibrace

Data Encryption

NINGHUI LI
Purdue University, West Lafayette, IN, USA

Definition

Data encryption is the process of transforming data (referred to as plaintext) to make it unreadable except to those possessing some secret knowledge, usually referred to as a key. The result of the process is encrypted data (referred to as ciphertext). Data encryption aims at preserving confidentiality of messages. The reverse process of deriving the plaintext from the ciphertext (using the key) is known as decryption. A cipher is a pair of algorithms which perform encryption and decryption. The study of data encryption is part of cryptography. The study of how to break ciphers, i.e., to obtain the meaning of encrypted information without access to the key, is called cryptanalysis.

Historical Background

Encryption has been used to protect communications since ancient times by militaries and governments to facilitate secret communication. The earliest known usages of cryptography include a tool called Scytale, which was used by the Greeks as early as the seventh century BC, and the Caesar cipher, which was used by Julius Caesar in the first century B.C.

The main classical cipher types are transposition ciphers, which rearrange the order of letters in a message, and substitution ciphers, which systematically replace letters or groups of letters with other letters or groups of letters. Ciphertexts produced by classical ciphers always reveal statistical information about the plaintext. Frequent analysis can be used to break classical ciphers.

Early in the twentieth century, several mechanical encryption/decryption devices were invented, including rotor machines – most famously the Enigma machine used by Germany in World War II. Mechanical encryption devices, and successful attacks on them, played a vital role in World War II.

Cryptography entered modern age in the 1970s, marked by two important events: the introduction of the U.S. Data Encryption Standard and the invention of public key cryptography. The development of digital computers made possible much more complex ciphers. At the same time, computers have also assisted cryptanalysis. Nonetheless, good modern ciphers have stayed ahead of cryptanalysis; it is usually the case that use of a quality cipher is very efficient (i.e., fast and requiring few resources), while breaking it requires an effort many orders of magnitude larger, making cryptanalysis so inefficient and impractical as to be effectively impossible.

Today, strong encryption is no longer limited to secretive government agencies. Encryption is now widely used by the financial industry to protect money transfers, by merchants to protect credit-card information in electronic commerce, by corporations to secure sensitive communications of proprietary information, and by citizens to protect their private data and communications.

Foundations

Data encryption can be either secret-key based or public-key based. In secret-key encryption (also known as symmetric encryption), a single key is used for both encryption and decryption. In public-key encryption (also known as asymmetric encryption), the encryption key (also called the public key) and the corresponding decryption key (also called the private key) are different. Modern symmetric encryption algorithms are often classified into stream ciphers and block ciphers.

Stream Ciphers

In a stream cipher, the key is used to generate a pseudo-random key stream, and the ciphertext is computed by using a simple operation (e.g., bit-by-bit XOR or byte-by-byte modular addition) to combine the plaintext bits and the key stream bits. Mathematically, a stream cipher is a function $f :\{0,1\}^l !\{0,1\}^m$, where l is the key size, and m determines the length of the longest message that can be encrypted under one key.

Many stream ciphers implemented in hardware are constructed using linear feedback shift registers (LFSRs). The use of LFSRs on their own, however, is insufficient to provide good security. Additional variations and enhancements are needed to increase the security of LFSRs.

The most widely-used software stream cipher is RC4. It was designed by Ron Rivest of RSA Security in 1987. It is used in popular protocols such as Secure Socket Layer (SSL) (to protect Internet traffic) and WEP (to secure wireless networks).

Stream ciphers typically execute at a higher speed than block ciphers and have lower hardware complexity. However, stream ciphers can be susceptible to serious security problems if used incorrectly; in particular, the same starting state (i.e., the same generated key stream) must never be used twice.

Block Ciphers

A block cipher operates on large blocks of bits, often 64 or 128 bits. The two most widely used block ciphers are the Data Encryption Standard (DES) and the Advanced Encryption Standard (AES).

DES is a block cipher selected as Federal Information Processing Standard for the United States in 1976. It has subsequently enjoyed widespread use internationally. The block size of DES is 64 bits, and the key size 56 bits. The main weakness of DES is its short key size, which makes it vulnerable to brute force attacks that try all possible keys.

One way to overcome the short key size of DES is to use Triple DES (3DES), which encrypts a 64-bit block by running DES three times using three DES keys.

AES was announced as an U.S. Federal Information Processing Standard on November 26, 2001. The algorithm has been invented by Joan Daemen and Vincent Rijmen and is formerly known as Rijndael. AES uses a block size of 128 bits, and supports key sizes of 128 bits, 192 bits, and 256 bits.

As messages to be encrypted may be of arbitrary length, and as encrypting the same plaintext under the same key always produces the same output, several modes of operation have been invented which allow block ciphers to provide confidentiality for messages of arbitrary length. For example, in the electronic codebook (ECB) mode, the message is divided into blocks and each block is encrypted separately. The disadvantage of this method is that identical plaintext blocks are encrypted into identical ciphertext blocks. It is not recommended for use in cryptographic protocols. In the cipher-block chaining (CBC) mode, each block of plaintext is XORed with the previous ciphertext block before being encrypted. This way, each ciphertext block is dependent on all plaintext blocks processed up to that point. Also, to make each message unique, an initialization vector must be used in the first block and should be chosen randomly.

Public Key Encryption Algorithms

When using symmetric encryption for secure communication, the sender and the receiver must agree upon a key and the key must be kept secret so that no other party knows the key. This means that the key must be distributed using a secure, but non-cryptographic, method; for example, a face-to-face meeting or a trusted courier. This is expensive and even impossible in some situations. Public key encryption was invented to solve the key distribution problem. When public key encryption is used, users can distribute public keys over insecure channels.

One of the most widely used public-key encryption algorithms is RSA. RSA was publicly described in 1977 by Ron Rivest, Adi Shamir and Leonard Adleman at MIT; the letters RSA are the initials of their surnames.

A central problem for public-key cryptography is proving that a public key is authentic and has not been tampered with or replaced by a malicious third party. The usual approach to this problem is to use a public key infrastructure (PKI), in which one or more third parties, known as certificate authorities, certify ownership of key pairs.

Asymmetric encryption algorithms are much more computationally intensive than symmetric algorithms. In practice, public key cryptography is used in combination with secret-key methods for efficiency reasons. For encryption, the sender encrypts the message with a secret-key algorithm using a randomly generated key, and that random key is then encrypted with the recipient's public key.

Attack Models

Attack models or attack types for ciphers specify how much information a cryptanalyst has access to when cracking an encrypted message. Some common attack models are:

- *Ciphertext-only attack:* the attacker has access only to a set of ciphertexts.
- *Known-plaintext attack:* the attacker has samples of both the plaintext and its encrypted version (ciphertext).
- *Chosen-plaintext attack:* the attacker has the capability to choose arbitrary plaintexts to be encrypted and obtain the corresponding ciphertexts.
- *Chosen-ciphertext attack:* the attacker has the capability to choose a number of ciphertexts and obtain the plaintexts.

(*Abridged*)

Recommended Reading

1. Federal information processing standards publication 46-3: data encryption standard (DES), 1999.

2. Federal information processing standards publication 197: advanced encryption standard, Nov. 2001.

3. Diffie W. and Hellman M.E. New directions in cryptography. IEEE Trans. Inform. Theory, 22:644–654, 1976.

4. Kahn D. The codebreakers: the comprehensive history of secret communication from ancient times to the internet. 1996.

5. Menezes A.J., Oorschot P.C.V., and Vanstone S.A. Handbook of applied cryptography (revised reprint with updates). CRC, West Palm Beach, FL, USA, 1997.

6. Rivest R.L., Shamir A., and Adleman L.M. A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM, 21:120–126, 1978.

Answer the following questions:

1) What is data encryption?
2) What is the difference between plaintext and ciphertext?
3) Is decryption synonymous with cryptanalysis?
4) What are the main classical cipher types?
5) Name some of the ways in which data encryption is currently used.
6) What is symmetric encryption?
7) Explain the difference between symmetric and asymmetric encryption.
8) Into what two groups are modern symmetric algorithms classified?
9) What are the advantages and disadvantages of stream ciphers?
10) What are the two most widely used block ciphers?
11) „The key must be distributed using a secure method." Does this apply to symmetric or public-key encryption?
12) What are some of the most common attack models?

Match the following terms with their definitions:

1) plaintext
2) key
3) encryption algorithm
4) block cipher
5) ciphertext
6) cryptanalysis
7) stream cipher
8) public key cryptography
9) encryption

a) an encrypted text message
b) a variable that is combined in some way with the unencrypted text
c) a formula used to turn the ordinary data or "plaintext" into a secret code or "ciphertext"
d) breaks a message up into chunks and combines a key with each chunk
e) applies a key to each bit, one at a time
f) an ordinary readable text before being encrypted or after being decrypted
g) algorithmic schemes that encode plain text into non-readable form providing privacy
h) retrieval of the plaintext from the ciphertext, without necessarily knowing the key or the algorithm
i) a system that uses two keys that work together

Mark the following statements as *true* or *false*:

1) The process of deriving the plaintext from the ciphertext using a key is decryption.
2) Cryptography is part of data encryption.
3) Cryptanalysis is a synonym of *breaking* the cipher, ciphertext, or cryptosystem.
4) Transposition is systematic replacement of letters or groups of letters with other letters or groups of letters.
5) Secret-key encryption is also known as asymmetric encryption.
6) The Secure Socket Layer (SSL) is a common encryption protocol used to protect Internet traffic.
7) Triple-DES is no more secure than DES even though it encrypts the data three times.
8) One of the most widely used public-key encryption algorithms is RSA.

Vocabulary

ancient ['eɪn.*t*ʃᵊnt] 🔊 – starý, starodávný

approach to st [ə'prəʊtʃ] US [–'proʊtʃ] 🔊 – přístup

brute force [bruːt] 🔊 [fɔːs] US [fɔːrs] 🔊 – hrubá síla

ciphertext ['saɪ.fəʳ.tekst] US [–fɚ] 🔊 – zašifrovaný text

confidentiality [ˌkɒn.fɪ.den.*t*ʃi'æl.ɪ.ti] US [ˌkɑːn.fɪ.den.*t*ʃi'æl.ə.t̬i] 🔊 – důvěrnost

cryptanalyst [krɪpt'æn.ə.lɪst] 🔊 – kdo se zabývá kryptanalýzou

cryptography [krɪp'tɒgrəfi] US [krɪp'tɑːgrəfi] 🔊 – kryptografie

decryption [dɪ'krɪpʃᵊn] 🔊 – rozkódování, dešifrace

dependent on st [dɪ'pen.dᵊnt] 🔊 – závislý na něčem

efficient [ɪ'fɪʃ.ᵊnt] 🔊 – rychlý

encryption [ɪn'krɪpʃᵊn] 🔊 – kódování, kryptování, šifrování

enhancement [ɪn'hɑːn*t*s.mənt] US [–'hæn*t*s–] 🔊 – vylepšení, posílení

formerly ['fɔː.mə.li] US ['fɔːr.mɚ–] 🔊 – dříve

in particular [pə'tɪk.jʊ.ləʳ] US [pɚ'tɪk.jə.lɚ] 🔊 – obzvlášť, zejména

key [kiː] 🔊 – klíč

merchant ['mɜː.tʃᵊnt] US ['mɝː–] 🔊 – obchodník

nonetheless [ˌnʌn.ðə'les] 🔊 – nicméně, přesto

order of magnitude ['ɔː.dəʳ] US ['ɔːr.dɚ] 🔊 ['mæg.nɪ.tjuːd] US [–tuːd] 🔊 – řádová hodnota

publicly ['pʌb.lɪ.kli] 🔊 – veřejně

randomly ['ræn.dᵊm.li] – náhodně

reverse [rɪ'vɜːs] US [–'vɝːs] 🔊 – opačný

secretive ['siː.krə.tɪv] US [–t̬ɪv] 🔊 – uzavřený

subsequently ['sʌb.sɪ.kwənt.li] 🔊 – později

substitution [ˌsʌb.stɪ'tjuː.ʃᵊn] US [–'tuː–] 🔊 – nahrazení, substituce

susceptible [sə'sep.tɪ.bl] 🔊 – náchylný

to agree upon [ə'griː] 🔊 [ə'pɒn] US [–'pɑːn] 🔊 – ujednat, dohodnout se na něčem

to aim at st [eɪm] 🔊 – zaměřovat se na něco, zamířit

to be ahead of [ə'hed] 🔊 – mít náskok

to break a cipher [breɪk] 🔊 ['saɪ.fəʳ] US [–fɚ] 🔊 – prolomit šifru

to crack [kræk] 🔊 – rozluštit

to derive st from st [dɪ'raɪv] 🔊 – odvodit něco z něčeho

to encrypt st [ɪn'krɪpt] 🔊 – zašifrovat něco, zakódovat

to execute ['ek.sɪ.kjuːt] 🔊 – provést, vykonat

to facilitate st [fə'sɪl.ɪ.teɪt] 🔊 – umožnit, zjednodušit něco

to overcome [ˌəʊ.və'kʌm] US [ˌoʊ.vɚ–] 🔊 – překonat, přemoci

to reveal [rɪ'viːl] 🔊 – odhalit, vyzradit

to reverse st [rɪ'vɜːs] US [–'vɝːs] 🔊 – obrátit něco, zvrátit, vrátit zpět

to secure [sɪ'kjʊəʳ] US [–'kjʊr] 🔊 – zabezpečit

to substitute 1 for 2 ['sʌb.stɪ.tjuːt] US [–tuːt] 🔊 – nahradit 2 1

to tamper with st ['tæm.pə] US ['tæm.pɚ] 🔊 – hrát si s něčím (nedovoleně)

to transform st [træns'fɔːm] US [–'fɔːrm] 🔊 – (pře)transformovat něco, (pře)měnit

transposition [ˌtræn.spə'zɪʃ.ᵊn] 🔊 – transpozice, přemístění

vulnerable to st ['vʌl.nᵊr.ə.bl] US ['vʌl.nɚ.ə–] 🔊 – náchylný k

Phrases

one way to overcome – způsob, jak překonat

Decision Tree Classification

ALIN DOBRA
University of Florida, Gainesville, FL, USA

Definition

Decision tree classifiers are decision trees used for classification. As any other classifier, the decision tree classifiers use values of attributes/features of the data to make a class label (discrete) prediction. Structurally, decision tree classifiers are organized like a *decision tree* in which simple conditions on (usually single) attributes label the edge between an intermediate node and its children. Leaves are labeled by class label predictions.

Foundations

Decision tree classifiers are especially attractive in a data mining environment for several reasons. First, due to their intuitive representation, the resulting model is easy to assimilate by humans [2]. Second, decision tree classifiers are non-parametric and thus especially suited for exploratory knowledge discovery. Third, decision tree classifiers can be constructed relatively fast compared to other methods [8]. And last, the accuracy of decision tree classifiers is comparable or superior to other classification models [8].

As it is the case for most classification tasks, the kind of data that can be represented by decision tree classifiers is of tabular form, as depicted in Table 1. Each data point occupies a row in the table. The names of columns are characteristics of the data and are called *attributes.* Attributes whose domain is numerical are called *numerical attributes*, whereas attributes whose domain is not numerical are called *categorical attributes.* One of the categorical attributes is designated as the *predictive attribute.* The predictive attribute needs to be predicted from values of the other attributes. For the example in Table 1, ''Car Type'' is a categorical attribute, ''Age'' is a numerical attribute and ''Lives in Suburb?'' is the predictor attribute.

Figure 1 (see the exercise section) depicts a classification tree, which was built based on data in Table 1. It predicts if a person lives in a suburb based on other information about the person. The predicates, that label the edges (e.g., Age <_30), are called *split predicates* and the attributes involved in such predicates, *split attributes.* In traditional classification and regression trees only deterministic split predicates are used (i.e., given the split predicate and the value of the attributes, it can be determined if the attribute is true or false). Prediction with classification trees is done by navigating the tree on true predicates until a leaf is reached, when the prediction in the leaf (YES or NO in the example) is returned.

Formal Definition

A classification tree is a directed, acyclic graph τ with tree shape. The root of the tree – denoted by Root (τ) – does not have any incoming edges. Every other node has exactly one incoming edge and may have 0, 2 or more outgoing edges. A node T without outgoing edges is called *leaf node,* otherwise T is called an *internal node.* Each leaf node is labeled with one class label; each internal node T is labeled with one attribute variable XT, called the *split attribute.* The class label associated with a leaf node T is denoted by Label(T).

Decision Tree Classification
Table 1: Example Training Database

Car Type	Driver Age	Children	Lives in Suburb?
sedan	23	0	yes
sports	31	1	no
sedan	36	1	no
truck	25	2	no
sports	30	0	no
sedan	36	0	no
sedan	25	0	yes
truck	36	1	no
sedan	30	2	yes
sedan	31	1	yes
sports	25	0	no
sedan	45	1	yes
sports	23	2	no
truck	45	0	yes

(Abridged)

Recommended Reading

1. Agresti A. Categorical data analysis. John Wiley and Sons. (1990).
2. Breiman L., Friedman J.H., Olshen R.A., and Stone C.J. (1984). Classification and regression trees. Belmont: Wadsworth.
3. Buntine W. Learning classification trees. Artificial Intelligence frontiers in statistics Chapman & Hall, London. (pp. 182–201).
4. Cox L.A., Qiu Y., and Kuehner W. Heuristic least-cost computation of discrete classification functions

with uncertain argument values. Annals of Operations Research, 21, 1–30. (1989).

5. Frank E. Pruning decision trees and lists. Doctoral dissertation, Department of Computer Science, University of Waikato, Hamilton, New Zealand. (2000).

6. Hyafil L., and Rivest R.L. Constructing optimal binary decision trees is np-complete. Information Processing Letters, 5, 15–17. (1976).

7. James M. Classification algorithms.Wiley. (1985).

8. Lim T.-S., Loh W.-Y., and Shih Y.-S. An empirical comparison of decision trees and other classification methods (Technical Report 979). Department of Statistics, University of Wisconsin, Madison. (1997).

9. Loh W.-Y. and Shih Y.-S. Split selection methods for classification trees. Statistica Sinica, 7. (1997).

10. Murthy S.K. Automatic construction of decision trees from data: A multi-disciplinary survey. Data Mining and Knowledge Discovery. (1997).

11. Quinlan J.R. Induction of decision trees. Machine Learning, 1, 81–106. (1986).

12. Quinlan J.R. Learning with Continuous Classes. In: Proc. 5th Australian Joint Conference on Artificial Intelligence (pp. 343–348). (1992).

13. Quinlan J.R. C4.5: Programs for machine learning. Morgan Kaufman. (1993b).

14. Murphy O.J. and Mccraw R.L. Designing storage efficient decision trees. IEEE Transactions on Computers, 40, 315–319. (1991)

Answer the following questions:

1) What are *decision trees?*
2) How are decision tree classifiers organized (structurally)?
3) What makes decision tree classifiers useful in data mining environment?
4) The text says that *"… the kind of data that can be represented by decision tree classifiers is of tabular form."* What is meant by this?
5) Describe the difference between *numerical* and *categorical* attributes.
6) What are *predictive attributes?*
7) What is a *leaf node?*

Match the following terms with their definitions:

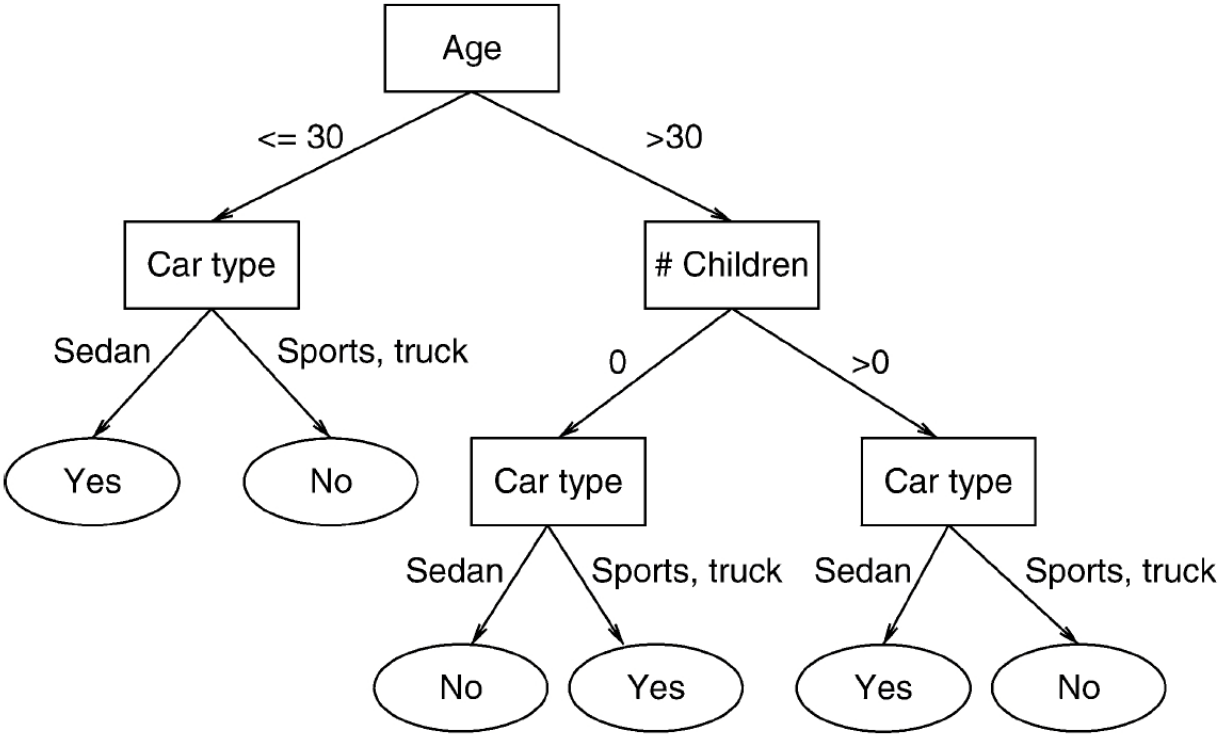
1) classification tree
2) predictive attribute
3) split attribute
4) data mining

a) a way of searching for certain information in a huge amount of data
b) an attribute determined by values of other attributes
c) an acyclic graph of a tree shape
d) an element of a split predicate

Mark the following statements *true* or *false:*

1) Deterministic split predicates are those on the grounds of which it can be determined whether an attribute is true or false.
2) A leaf represents an intermediate node.
3) Each leaf node is labeled with two or more class labels.
4) Each internal node is labeled with one predictive attribute.

Name the individual elements of the following tree (Figure 1):



Vocabulary

accuracy ['æk.jʊ.rə.si] 🔊 – přesnost

accurate ['æk.jʊ.rət] 🔊 – přesný

acyclic [ə'saɪ.klɪ.k, ə'sɪk.lɪ–, eɪ'saɪ.klɪ.k, eɪ'sɪk.lɪ–] 🔊 – necyklický

attribute ['æt.rɪ.bjuːt] 🔊 – atribut, rys

categorical [ˌkæt.ə'gɒr.ɪ.kᵊl] US [ˌkæt̬.ə'gɑːr–] 🔊 – kategorický

categorical attributes ['æt.rɪ.bjuːts] – kategorické atributy

classification [ˌklæs.ɪ.fɪ'keɪ.ʃᵊn] 🔊 – klasifikace

classifier ['klæ.sɪ.fai.ə] 🔊 – klasifikátor

column ['kɒl.əm] US ['kɑː.ləm] 🔊 – sloupec (tabulky)

comparable to st ['kɒm.p ᵊ r.ə.bl] US ['kɑːm.pɚ–] 🔊 🔊 – srovnatelný s něčím

compared to st [kəm'peəʳd] US [–per] 🔊 – ve srovnání s něčím

data mining ['deɪ.tə] US [–t̬ ə] 🔊 🔊 ['maɪ.nɪŋ] 🔊 – dolování dat

decision tree [dɪ'sɪʒ.ᵊn] 🔊 [triː] 🔊 – rozhodovací strom

deterministic [di.tɜː.mɪ'nɪs.tɪk] US [–'tɝː–] 🔊 – deterministický

discrete [dɪ'skriːt] 🔊 – nespojitý, rozpojený

edge [edʒ] 🔊 – hrana, okraj

exploratory [ek'splɒr.ə.tri] US [–'splɑː.rə.tɔːr.i] 🔊 – průzkumný

feature ['fiː.tʃəʳ] US [–tʃɚ] 🔊 – rys, znak

incoming ['ɪnˌkʌm.ɪŋ] 🔊 – příchozí

intermediate node [ˌɪn.tə'miː.di.ət] US [–t̬ ɚ–] 🔊 [nəʊd] US [noʊd] 🔊 – mezilehlý uzel

internal node [ɪn'tɜː.nəl] US [–'tɝː–] 🔊 – vnitřní uzel

intuitive [ɪn'tjuː.ɪ.tɪv] US [–'tuː.ɪ.t̬ ɪv] 🔊 – intuitivní

label ['leɪ.bᵊl] 🔊 – popisek, značka

leaf (pl. leaves) [liːf] 🔊, pl. [liːvz] – list

model ['mɒd.ᵊl] US ['mɑː.dᵊl] 🔊 – model

node [nəʊd] US [noʊd] 🔊 – uzel

numerical [njuː'mer.ɪ.kl] US [nuː–] 🔊 – numerický

numerical attributes ['æt.rɪ.bjuːts] – numerické atributy

outgoing [ˌaʊt'gəʊ.ɪŋ] US ['aʊt.goʊ–] 🔊 – odchozí

predicate ['pred.ɪ.kət] US [–kɪt] 🔊 – predikát

prediction [prɪ'dɪk.ʃᵊn] 🔊 – predikce, předpověď

row [rəʊ] US [roʊ] 🔊 – řádek (tabulky)

regression tree [rɪ'greʃ.ᵊn] 🔊 [triː] 🔊 – regresní strom

split st [splɪt] 🔊 – rozdělit něco, rozštěpit

superior to st [suː'pɪə.ri.əʳ] US [–'pɪr.i.ɚ] 🔊 – nadřazený něčemu

to assimilate [ə'sɪm.ɪ.leɪt] 🔊 – přizpůsobit

to construct st [kən'strʌkt] 🔊 – vytvořit něco , zkonstruovat

to label st ['leɪ.bᵊl] 🔊 – označit něco (popiskem, značkou atd.)

to navigate st ['næv.ɪ.geɪt] 🔊 – navigovat něco; pohybovat se po něčem

value ['væl.juː] 🔊 – hodnota

whereas [weə'ræz] US [wer'æz] 🔊 – kdežto

Steganography

RADU SION
Stony Brook University, Stony Brook, NY, USA

Definition

Steganography (from the Greek ‘‘steganos’’ covered) is a term denoting mechanisms for hiding information within a ‘‘cover’’ such that, generally, only an intended recipient will (i) have knowledge of its existence, and (ii) will be able to recover it from within its cover. In modern digital steganography applications, the cover is often a multimedia object such as an image that is minorly altered in the steganographic process.

Steganographic techniques have been deployed for millennia and several primitive war-time instances are described in the Histories of Herodotus of Halicarnassus, including a case of a message tattooed on the shaven head of a slave, which, when covered with grown hair acted as an effective ‘‘cover’’ when traversing enemy lines.

Steganography versus Watermarking

A common trend of term misuse is associated with steganography. Specifically, many sources consider the term ‘‘watermarking’’ as equivalent. This is incorrect. There are fundamental differences, from both application perspectives and associated challenges. Steganography usually aims at enabling Alice and Bob to exchange messages in a manner as stealthy as possible, through a hostile medium where Malory could lurk.

On the other hand, *Digital Watermarking* is deployed by a rights holder (Alice) as a court proof of rights over a Work, usually in the case when an adversary (Mallory) would benefit from using or selling that very same Work or maliciously modified versions of it. In Digital Watermarking, the actual value to be protected lies in the Works themselves, whereas pure steganography usually makes use of them as simple value ‘‘transporters.’’ In Watermarking, Rights Assessment is achieved by demonstrating (with the aid of a ‘‘secret’’ known only to Alice – ‘‘watermarking key’’) that a particular Work exhibits a rare property (‘‘hidden message’’ or ‘‘watermark’’). For purposes of convincing the court, this property needs to be so rare that if one considers any other random Work ‘‘similar enough’’ to the one in question, this property is ‘‘very improbable’’ to apply (i.e., bound false-positives rate). It also has to be relevant, in that it somehow ties to Alice (e.g., by featuring the bit string ‘‘(c) by Alice’’). There is a threshold determining the ability to convince the court, related to the ‘‘very improbable’’ assessment. This defines a main difference from steganography: from the court’s perspective, specifics of the property (e.g., watermark message) are not important as long as they link to Alice (e.g., by saying ‘‘(c) by Alice’’) and, she can prove ‘‘convincingly’’ it is she who induced it to the (non-watermarked) original. In watermarking, the emphasis is on ‘‘detection’’ rather than ‘‘extraction.’’ Extraction of a watermark, or bits of it, is usually a part of the detection process but just complements the process up to the extent of increasing the ability to convince in court.

Fingerprinting

In this application of steganography, license violators are ‘‘tracked’’ by hiding uniquely identifying ‘‘*fingerprints*.’’ If the Work would then be found in the public domain, the fingerprints can then be used to assess the source of the leak.

(Abridged)

Recommended Reading

1. WatermarkingWorld. Online at http://www.watermarkingworld.org/

Answer the following questions:

1) What is steganography?
2) What is the meaning of the word *steganography*?
3) How is the information hidden?
4) What can be the “cover” in modern digital steganography applications?
5) What is the difference between steganography and watermarking?
6) In digital watermarking is the work itself the actual value or is it just a value “transporter”?
7) Why do people watermark their works?
8) What is digital fingerprinting?
9) Does the watermark affect the use of the work?
10) What is the watermark key?

Match the following terms with their definitions:

1) steganography
2) watermarking
3) digital fingerprinting
4) cover
5) leak

a) adding new information, embedding it within a video and/or an audio signal
b) a multimedia object such as an image that is slightly changed in the steganographic process
c) hiding of a secret message within an ordinary message and extraction of it at its destination
d) the act of making secret information generally known
e) analyzing the media, identifying a unique set of inherent properties

Mark the following statements as *true* or *false*:

1) Digital watermarks and digital fingerprinting are now in use to track the copyright and ownership of electronic media.
2) Steganography hides the message so that it cannot be seen.
3) Steganography is always used for illegitimate reasons.
4) Steganography is the only way to protect the confidentiality of data.
5) The watermark key is the same as the encryption key.
6) Watermarks can just be applied to written forms of communication.
7) Extracting invisible watermarks requires a password.
8) Fingerprinting can be used to detect copyright violators.

Vocabulary

adversary ['æd.və.sər.i] US ['æd.vɚ.ser–] 🔊 – protivník, soupeř

as long as [lɒŋ] US [lɑːŋ] 🔊 – pokud

assessment [ə'ses.mənt] 🔊 – ohodnocení, posudek

convincingly [kən'vɪn*t*.sɪŋ.li] 🔊 – přesvědčivě

court [kɔːt] US [kɔːrt] 🔊 – soud

detection [dɪ'tek.ʃən] 🔊 – zjištění

emphasis on st ['em*p*.fə.sɪs] 🔊 – důraz na něco

evidence ['ev.ɪ.dən*t*s] 🔊 – důkaz

extent [ɪk'stent] 🔊 – rozsah

holder ['həʊl.dər] US ['hoʊl.dɚ] 🔊 – držitel

hostile ['hɒs.taɪl] US ['hɑː.stəl] 🔊 🔊 – nepřátelský

characteristic [ˌkær.ɪk.tə'rɪs.tɪk] US [ˌker–] 🔊 – typický znak

intended [ɪn'ten.dɪd] – zamýšlený

leak [liːk] 🔊 – únik informací, prozrazení

maliciously [mə'lɪʃ.ə.sli] – zlomyslně, záludně

minorly ['maɪ.nərli] US [–nɚ] – nepatrně

misuse [ˌmɪs'juːz] 🔊 – nesprávné použití

particular [pə'tɪk.jʊ.lər] US [pɚ'tɪk.jə.lɚ] 🔊 – konkrétní, jednotlivý

proof [pruːf] 🔊 – důkaz

property ['prɒp.ə.ti] US ['prɑː.pɚ.t̬i] 🔊 – vlastnictví, majetek, vlastnost

pure [pjʊər] US [pjʊr] 🔊 – čistý

purpose ['pɜː.pəs] US ['pɝː–] 🔊 – účel

random ['ræn.dəm] 🔊 – náhodný

rare [reər] US [rer] 🔊 – vzácný

recipient [rɪ'sɪp.i.ənt] 🔊 – příjemce

rights [raɪts] 🔊 – práva

shaven ['ʃeɪ.vən] 🔊 – oholený

slave [sleɪv] 🔊 – otrok

source [sɔːs] US [sɔːrs] 🔊 – zdroj

specifically [spə'sɪf.ɪ.kli] 🔊 – konkrétně

specifics [spə'sɪf.ɪks] 🔊 – specifika

stealthy ['stel.θi] 🔊 – tajný, kradmý

steganography [ste.gə'nɒ.grə.fi] US [ste.gə'nɑː.grə.fi] 🔊 – steganografie

string [strɪŋ] 🔊 – řetězec

technique [tek'niːk] 🔊 – technika, postup

threshold ['θreʃ.*h*əʊld] US [–*h*oʊld] 🔊 – práh

to achieve [ə'tʃiːv] 🔊 – dosáhnout

to aim at st [eɪm] 🔊 – zaměřit se na co

to alter ['ɒl.tər] US ['ɑːl.t̬ɚ] 🔊 – změnit

to apply [ə'plaɪ] 🔊 – uplatnit

to assess [ə'ses] 🔊 – ohodnotit, posoudit

to associate st with st [ə'səʊ.si.eɪt] US [–'soʊ–] 🔊 – spojovat (si) co s čím

to benefit from st ['ben.ɪ.fɪt] 🔊 – mít užitek, prospěch

to complement ['kɒm.plɪ.ment] US ['kɑːm–] 🔊 – doplnit

to consider st [kən'sɪd.ər] US [–ɚ] 🔊 – považovat, pokládat

to convince [kən'vɪn*t*s] 🔊 – přesvědčit, ubezpečit

to denote [dɪ'nəʊt] US [–'noʊt] 🔊 – označit

to deploy [dɪ'plɔɪ] 🔊 – rozvinout

to determine [dɪ'tɜː.mɪn] US [–'tɝː–] 🔊 – určit, stanovit

to enable [ɪ'neɪ.bl] 🔊 – umožnit

to exhibit [ɪg'zɪb.ɪt] 🔊 – projevit

to feature ['fiː.tʃər] US [–tʃɚ] 🔊 – obsahovat

to fingerprint ['fɪŋ.gə.prɪnt] US [–gɚ–] 🔊 – snímat otisky prstů

to hide [haɪd] 🔊 – skrýt

to increase [ɪn'kriːs] 🔊 🔊 – zvětšit, zvýšit

to induce [ɪn'djuːs] US [–'duːs] 🔊 – zavést

to lie [laɪ] 🔊 – ležet

to link [lɪŋk] 🔊 – spojovat, souviset

to lurk [lɜːk] US [lɝːk] 🔊 – číhat

to make use of st [meɪk] 🔊 [juːz] 🔊 – využít, zužitkovat co

to recover from st [rɪ'kʌv.ər] US [–ɚ] 🔊 – získat z

to tie [taɪ] 🔊 – svázat, propojit

to track [træk] 🔊 – stopovat, sledovat

to traverse [trə'vɜːs] US [–'vɝːs] 🔊 – překročit

violator ['vaɪə.leɪt] 🔊 – kdo porušuje zákon, dohodu

whereas [weə'ræz] US [wer'æz] 🔊 – kdežto, zatímco

work (a work) [wɜːk] US [wɝːk] 🔊 – dílo

Phrases

in a manner ['mæn.əʳ] US [-ɚ] 🔊 – způsobem

in question – dotyčný, zmíněný, příslušný

on the one hand – na jedné straně

on the other hand – na druhé straně

with the aid of st – za pomoci, s pomocí

Web Spam Detection

MARC NAJORK
Microsoft Research, Mountain View, CA, USA

Definition

Web spam refers to a host of techniques to subvert the ranking algorithms of web search engines and cause them to rank search results higher than they would otherwise. Examples of such techniques include content spam (populating web pages with popular and often highly monetizable search terms), link spam (creating links to a page in order to increase its link based score), and cloaking (serving different versions of a page to search engine crawlers than to human users). Web spam is annoying to search engine users and disruptive to search engines; therefore, most commercial search engines try to combat web spam. Combating web spam consists of identifying spam content with high probability and – depending on policy – downgrading it during ranking, eliminating it from the index, no longer crawling it, and tainting affiliated content. Commercial search engines treat their precise set of spam-prediction features as extremely proprietary, and features (as well as spamming techniques) evolve continuously as search engines and web spammers are engaged in a continuing ‘‘arms race.’’

Historical Background

Web spam is almost as old as commercial search engines. The first commercial search engine, Lycos, was incorporated in 1995; and the first known reference to ‘‘spamdexing’’ (a combination of ‘‘spam’’ and ‘‘indexing’’) dates back to 1996. Commercial search engines began to combat spam shortly thereafter, increasing their efforts as it became more prevalent.

Foundations

Given that the objective of web spam is to improve the ranking of select search results, web spamming techniques are tightly coupled to the ranking algorithms employed (or believed to be employed) by the major search engines. As ranking algorithms evolve, so will spamming techniques.

Given that web spamming techniques are constantly evolving, any taxonomy of these techniques must necessarily be ephemeral, as will be any enumeration of spam detection heuristics. However, there are a few constants:

Any successful web spamming technique targets one or more of the features used by the search engine’s ranking algorithms.

Web spam detection is a classification problem, and search engines use machine learning algorithms to decide whether or not a page is spam.

In general, spam detection heuristics look for statistical anomalies in some of the features visible to the search engines.

Web Spam Detection as a Classification Problem

Web spam detection can be viewed as a binary classification problem, where a classifier is used to predict whether a given web page or entire web site is spam or not. The machine learning community has produced a large number of classification algorithms, e.g. decision-tree based classifiers, SVM-based classifiers, Bayesian classifiers, and logistic regression classifiers. Some classifiers perform better than others and the spam detection community seems to favor decision tree- based ones.

Taxonomy of Web Spam Techniques

Content spam refers to any web spam technique that tries to improve the likelihood that a page is returned as a search result and to improve its ranking by populating the page with salient keywords. Populating a page with words that are popular query terms will cause that page to be part of the result set for those queries.

Naive spammers might perform content spam by stringing together a wide array of popular query terms. Search engines can counter this by employing language modeling techniques, since web pages that contain many topically unrelated keywords or that are grammatically ill-formed will exhibit statistical differences from normal web pages.

More sophisticated spammers might generate not a few, but rather millions of target web pages, each page augmented with just one or a few popular query terms. The remainder of the page may be entirely machine-generated (which might exhibit statistical anomalies that can be detected by the search engine), entirely copied from a human-authored web site such as Wikipedia (which can be detected by using near-duplicate detection algorithms), or stitched together from fragments of several human authored web sites (which is much harder, but not impossible to detect).

Link spam refers to any web spam technique that tries to increase the link-based score of a target web page by creating lots of hyperlinks pointing to it. The hyperlinks may originate from web pages owned and controlled by the spammer (generically called a link farm), they may originate from partner web sites (a technique known as link exchange), or they may originate from unaffiliated (and sometimes unknowing) third parties, for example, web-based discussion forums or in blogs that allow comments to be posted (a phenomenon called blog spam). Search engines can respond to link spam by mining the web graph for anomalous components, by propagating distrust from spam pages backwards through the web graph, and by using content-based features to identify spam postings to a blog. Many link spam techniques specifically target Google's PageRank algorithm, which not only counts the number of hyperlinks referring to a web page, but also takes the PageRank of the referring page into account. In order to increase the PageRank of a target page, spammers should create links on sites that have high PageRanks, and for this reason, there is a marketplace for expired domains with high PageRank, and numerous brokerages reselling them. Search engines can respond by temporarily dampening the endorsement power of domains that underwent a change in ownership.

Click spam refers to the technique of submitting queries to search engines that retrieve target result pages and then ‘‘clicking’’ on these pages in order to simulate user interest in the result. The result pages returned by the leading search engines contain client-side scripts that report clicks on result URLs to the engine, which can then use this implicit relevance feedback in subsequent rankings. Click spam is similar in method to click fraud, but different in objective. The goal of click spam is to boost the ranking of a page, while the goal of click fraud (generating a large number of clicks on search engine advertisements) is to spend the budget associated with a particular advertisement (to hurt the competitor who has placed the ad or simply to lower the auction price of said ad, which will drop once the budget of the winning bidder has been exhausted). In a variant of click fraud the spammer targets ads delivered to his own web by an ad network such as Google AdSense and obtains a revenue share from the ad-network. Both click fraud and click spam are trivial to detect if launched from a single machine, and hard to detect if launched from a bot-net consisting of tens of thousands of machines. Search engines tackle the problem by mining their click logs for statistical anomalies, but very little is known about their algorithms.

Cloaking refers to a host of techniques aimed at delivering (apparently) different content to search engines than to human users. Cloaking is typically used in conjunction with content spam, by serving a page containing popular query terms to the search engine (thereby increasing the likelihood that the page will be returned as the result of a search), and presenting the human user with a different page.

Cloaking can be achieved using many different techniques: by literally serving different content to search engines than to ordinary users (based for example on the well-known IP addresses of the major search engine crawlers), by rendering certain parts of the page invisible (say by setting the font to the same color as the background), by using client-side scripting to rewrite the page after it has been delivered (relying on the observation that search engine crawlers typically do not execute scripts), and finally by serving a page that immediately redirects the user's browser to a different page (either via client-side scripting or the HTML ‘‘meta-redirect’’ tag). Each variant of cloaking calls for a different defense. Search engines can guard against different versions of the same page by probing the page from unaffiliated IP addresses; they can detect invisible content by rendering the page; and they can detect page modifications and script-driven redirections by executing client-side scripts.

Key Applications

Web spam detection is used primarily by advertisement financed general-purpose consumer search engines. Web spam is not an issue for enterprise search engines, where the content providers, the search engine operator and the users are all part of the same organization and have shared goals. However, web spam is bound to become a problem in any setting where these three parties – content providers, searchers, and search engines – have different objectives. Examples of such settings include vertical search services, such as product search engines, company search engines, people search engines, or even scholarly literature search engines. Many of the basic concepts described above are applicable to these domains as well; the precise set of features useful for spam detection will depend on the ranking algorithms used by these vertical search engines.

(Abridged)

Recommended Reading

1. Becchetti L., Castillo C., Donato D., Leonardi S., and Baeza-Yates R. Using rank propagation and probabilistic counting for link-based spam detection. In Proc. KDD Workshop on Web Mining and Web Usage Analysis, 2006.

2. Castillo C., Donato D., Becchetti L., Boldi P., Leonardi S.,Santini M., and Vigna S. A reference collection for Web spam.ACM SIGIR Forum, 40(2):11–24, 2006.

3. Daswani N. and Michael Stoppelman and the Google Click Quality and Security Teams. The anatomy of clickbot.A. In Proc. 1stWorkshop on Hot Topics in Understanding Botnets, 2007.

4. Davison B.D. Recognizing nepotistic links on the web. In Proc. AAAI Workshop on Artificial Intelligence for Web Search, 2000.

5. Fetterly D., Manasse M., and Najork M. Spam, damn spam and statistics. In Proc. 7th Int. Workshop on the World Wide Web and Databases, 2004, pp. 1–6.

6. Gyöngyi Z., and Garcia-Molina H. Spam: its not just for Inboxes anymore. IEEE Comput. Mag., 38(10):28–34, 2005.

7. Gyöngyi Z. and Garcia-Molina H. Web Spam Taxonomy. In Proc. 1st Int.Workshop on Adversarial Information Retrieval on the Web, 2005, pp. 39–47.

8. Gyöngyi Z., Garcia-Molina H., and Pedersen J. Combating Web spam with trust rank. In Proc. 30th Int. Conf. on Very Large Data Bases, 2004, pp. 576–587.

9. Henzinger M., Motwani R., and Silverstein C. Challenges in web search engines. ACM SIGIR Forum 36(2):11–22, 2002.

10. Mishne G., Carmel D., and Lempel R. Blocking blog spam with language model disagreement. In Proc. 1st Int. Workshop on Adversarial Information Retrieval on theWeb, 2005, pp. 1–6.

11. Ntoulas A., Najork M., Manasse M., and Fetterly D. Detecting spam web pages through content analysis. In Proc. 15th Int. World Wide Web Conference, 2006, pp. 83–92.

12. Wang Y.M., Ma M., Niu Y., and Chen H. Spam double-funnel: connecting Web spammers with advertisers. In Proc. 16th Int. World Wide Web Conference, 2007, pp. 291–300.

13. Wu B. and Davison B. Detecting semantic cloaking on the web. In Proc. 15th Int. World Wide Web Conference, 2006, pp. 819–828

Answer the following questions:

1) What is web spam and why does it exist?
2) What are some of web spam techniques?
3) What does *web spam combating* consist of?
4) What is a search engine and what search engines do you know?
5) What is the difference between web spam detection and spamdexing?
6) "As ranking algorithms evolve, so will spamming techniques." Explain.
7) How is web spam detected?
8) What is the difference between *content spam* and *link spam*?
9) Is a *link farm* the same as *link exchange*? If not, explain the difference.
10) What does *cloaking* refer to?

Match the following terms with their definitions:

1) web spamming
2) spammers
3) search engine
4) content spam
5) link spam
6) cloaking

a) techniques delivering different content to search engines than to human users
b) behaviour that attempts to deceive search engine ranking algorithms
c) a technique populating the page with popular keywords
d) people who perform spamming
e) a technique creating lots of hyperlinks pointing to a web page
f) any program that searches a database and produces a list of results

Mark the following statements as *true* or *false*:

1) Web spam had existed long before the first commercial search engine was developed.
2) The goal of web spam is to mislead search engines and rank some pages higher than they deserve.
3) The primary consequence of web spamming is that the quality of search results decreases.
4) Redirection was invented to facilitate spamming.
5) Click spam and click fraud are different in method but their objective is the same.
6) Some spamming techniques can be combined.
7) The goal of click spam is to boost the ranking of a page.
8) Cloaking can be achieved e.g., by rendering certain parts of the page invisible.

Vocabulary

affiliated [ə'fɪl.i.eɪtɪd] 🔊 – přidružený, přičleněný

apparently [ə'pær.ᵊnt.li] US [–'per–] 🔊 – zřejmě

applicable to st [ə'plɪk.ə.bl] 🔊 🔊 – platný, uplatnitelný

array [ə'reɪ] 🔊 – množství, řada

brokerage [brəʊ.kə.rɪdʒ] US [broʊ–] 🔊 – makléřská firma

competitor [kəm'pet.ɪ.təʳ] US [–'pet̬.ɪ.t̬ ɚ] 🔊 – konkurent

crawler ['krɔː.ləʳ] US ['krɑː.lɚ] 🔊 – vyhledávací robot

disruptive [dɪs'rʌp.tɪv] 🔊 – rušivý

distrust [dɪ'strʌst] 🔊 – nedůvěra

endorsement [ɪn'dɔː.smənt] US [–'dɔːr–] 🔊 – podpora, doporučení

ephemeral [ɪ'fem.ᵊr.ᵊl] US [–ɚ–] 🔊 🔊 – prchavý, pomíjivý

host of st [həʊst] US [hoʊst] 🔊 – spousta, velké množství

likelihood ['laɪ.kli.hʊd] 🔊 – pravděpodobnost

monetizable ['mɒ.nɪ.tai.zə.bl] US ['mɑː.nɪ.tai.zə.bl] – zpeněžitelný

objective [əb'dʒek.tɪv] 🔊 – cíl

prevalent ['prev.ᵊl.ənt] 🔊 – obvyklý, běžný

proprietary [prə'praɪə.tri] US [–ter.i] 🔊 – vlastnický, soukromý

query ['kwɪə.ri] US ['kwɪr.i] 🔊 🔊 – dotaz

remainder [rɪ'meɪn.dəʳ] US [–dɚ] 🔊 – zbytek, zbývající část

revenue ['rev.ᵊn.juː] US [–ə.nuː] 🔊 – příjem, tržba

salient ['seɪ.li.ənt] 🔊 – význačný

thereby [ˌðeə'baɪ] US [ˌðer–] 🔊 – tímto, čímž

tightly ['taɪt.li] – těsně

to augment [ɔːg'ment] US [ɑːg–] 🔊 – zvětšit, zvýšit

to be bound to [baʊnd] 🔊 – muset

to boost [buːst] 🔊 – zvýšit, pozvednout

to cloak [kləʊk] US [kloʊk] 🔊 – zahalit, schovat

to combat st ['kɒm.bæt] US ['kɑːm–] 🔊 – bojovat proti něčemu

to counter st ['kaʊn.təʳ] US [–t̬ ɚ] 🔊 – odporovat, čelit

to couple ['kʌp.l] 🔊 – propojit, spojit

to dampen ['dæm.pən] 🔊 – ztlumit, zmenšit

to downgrade [ˌdaʊn'greɪd] 🔊 – degradovat

to engage in st [ɪn'geɪdʒ] 🔊 – věnovat se čemu

to lower ['ləʊ.əʳ] US ['loʊ.ɚ] 🔊 🔊 – snížit

to originate from [ə'rɪdʒ.ɪ.neɪt] 🔊 – pocházet z něčeho

to perform [pə'fɔːm] US [pɚ'fɔːrm] 🔊 – provést, uskutečnit

to populate ['pɒp.jʊ.leɪt] US ['pɑː.pjə–] 🔊 – zaplnit

to probe [prəʊb] US [proʊb] 🔊 – zkoumat, zjišť ovat

to propagate ['prɒp.ə.geɪt] US ['prɑː.pə–] 🔊 – šířit

to rank [ræŋk] 🔊 – zaujmout místo, být hodnocen

to redirect [ˌriː.dɑɪ'rekt] 🔊 – přesměrovat

to rely on st [rɪ'laɪ] 🔊 – spoléhat na co

to retrieve [rɪ'triːv] 🔊 – vyhledat

to string [strɪŋ] 🔊 – svázat

to subvert [səb'vɜːt] US [–'vɝːt] 🔊 – podkopat, rozvrátit

to tackle st ['tæk.l] 🔊 🔊 – vypořádat se s něčím

to taint [teɪnt] 🔊 – zabarvit, poznamenat

to treat [triːt] 🔊 – považovat

to undergo a change [ˌʌn.də'gəʊ] US [–dɚ'goʊ] 🔊 [tʃeɪndʒ] 🔊 – projít změnou

Phrases

for this reason ['riː.zᵊn] 🔊 – z tohoto důvodu

in conjunction with [kən'dʒʌŋk.ʃᵊn] 🔊 – společně, dohromady s

Peer to Peer Overlay Networks: Structure, Routing and Maintenance

WOJCIECH GALUBA,
SARUNAS GIRDZIJAUSKAS
EPFL, Lausanne, Switzerland

Definition

A peer-to-peer overlay network is a computer network built on top of an existing network, usually the Internet. Peer-to-peer overlay networks enable participating peers to find the other peers not by the IP addresses but by the specific logical identifiers known to all peers. Usually, peer-to-peer overlays have the advantage over the traditional client-server systems because of their scalability and lack of single-point-of-failure. Peer-to-peer overlays are commonly used for file sharing and real time data streaming.

Historical Background

The rise of the Internet brought the first instances of peer-to-peer overlays like the Domain Name System (DNS), the Simple Mail Transfer Protocol (SMTP), USENET and more recently IPv6, which were needed to facilitate the operation of the Internet itself. These peer-to-peer overlays were intrinsically decentralized and represented symmetric nature of the Internet, where every node in the overlay had equal status and assumed cooperative behavior of the participating peers. The beginning of the file-sharing era and the rise and fall of the first file-sharing peer-to-peer system Napster [9] (2000–2001) paved the way for the second generation of peer-to-peer overlays like Gnutella [5] (2000) and Freenet [4] (2001). The simple protocols and unstructured nature made these networks robust and lacking Napster's drawbacks like single-point-of-failure. Since 2001, these peer-to-peer overlays became extensively popular and accounted for the majority of the Internet traffic. Soon after it was evident that the unstructured nature of Gnutella-like systems is embarrassingly wasteful in bandwidth, more efficient structured overlays appeared, like the Distributed Hash Tables (DHTs), which used the existing resources more effectively (e.g., Chord [13]). Currently, unstructured peer-to-peer overlays are sparsely used, as the most popular peer-to-peer applications for file-sharing and data-streaming (e.g., Skype [12], Kademlia [8], KaZaA [6]) are implemented using structured or hybrid overlay concepts.

Foundations

Taxonomy

There are many features of peer-to-peer overlays, by which they can be characterized and classified [2,11]. However, strict classification is not easy since many features have mutual dependencies on each other, making it difficult to identify the distinct overlay characteristics (e.g., overlay topologies versus routing in overlays). Although every peer-to-peer overlay can differ by many parameters, each of them will have to have certain network structure with distinctive routing and maintenance algorithms allowing the peer-to-peer application to achieve its purpose. Thus, most commonly, peer-to-peer overlays can be classified by:

1. Purpose of use;
2. Overlay structure;
3. Employed routing mechanisms;
4. Maintenance strategies.

Purpose of Use

Peer-to-peer overlays are used for an efficient and scalable sharing of individual peers' resources among the participating peers. Depending on the type of the resources which are shared, the peer-to-peer overlays can be identified as oriented for:

1. Data-sharing (data storage and retrieval);
2. Bandwidth-sharing (streaming);
3. CPU-sharing (distributed computing).

Data-sharing peer-to-peer overlays can be further categorized by their purpose to perform one or more specific tasks like file-sharing (by-far the most common use of the peer-to-peer overlays), information retrieval (peer-to-peer web search), publish/subscribe services and semantic web applications. The examples of such networks are BitTorrent [3] (file-sharing), YaCy-Peer [14] (web search), etc.

Bandwidth-sharing peer-to-peer overlays to some extent are similar to the data-sharing ones, however, are mainly aimed at the efficient streaming of real-time data over the network. Overlay's ability to find several disjoint paths from source to destination can significantly boost the performance of the data streaming applications. Bandwidth-sharing peer-to-peer overlays are mostly found in peer-to-peer telephony, peer-to-peer video/TV, sensor networks and peer-to-peer publish/subscribe services. Currently Skype [12] is arguably the most prominent peer-to-peer streaming overlay application.

For the computationally intensive tasks, when the CPU resources of a single peer cannot fulfill its needs, *CPU-sharing* peer-to-peer overlays can provide plenty of CPU resources from the participating idle overlay peers. Currently, only major scientific experiments employ such strategy for the tasks like simulation of protein folding or analysis of an astronomic radio signals. Although not being a pure peer-to-peer overlay, Berkeley Open Infrastructure for Network Computing (BOINC) is very popular among such networks, supporting such distributed computing projects as SETI@home, folding@home, AFRICA@home, etc.

Overlay Structure

Peer-to-peer overlays significantly differ by the topology of the networks which they form. There exist a wide scope of possible overlay instances, ranging from centralized to purely decentralized ones, however, most commonly three classes of network topology are identified:

1. Centralized overlays;
2. Decentralized overlays;
3. Hybrid overlays.

Depending on the routing techniques and whether the overlay network was created by some specific rules (deterministically) or in an ad hoc fashion (non-deterministically), overlay networks can be also classified into structured and unstructured peer-to-peer overlays.

Centralized Overlays

Peer-to-peer overlays based on *centralized* topologies are pretty efficient since the interaction between peers is facilitated by a central server which stores the global index, deals with the updates in the system, distributes tasks among the peers or quickly responds to the queries and gives complete answers to them (Fig.1(a)). However, not all the purposes of use fit the centralized network overlay model. Centralized overlays usually fail to scale with the increase of the number of participating peers. The centralized component rapidly becomes the performance bottleneck. The existence of a single-point-of-failure (e.g., Napster [9]) also prevents from using centralized overlays for many potential data-sharing applications.

Decentralized Overlays

Because of the aforementioned drawbacks, *decentralized* structured and unstructured overlays emerged, which use purely decentralized network model, and do not differ peers as servers or clients, but treat all of them equally – as if they were both servers and clients at the same time (Fig.1(b)). Thus, such peer-to-peer overlays successfully deal with the scalability and can exist without any governing authority.

Hybrid Overlays

There also exist many *hybrid* peer-to-peer overlays (super-peer systems) which are a trade-off between different degree of topology centralization and structure flexibility. Hybrid overlays usually use hierarchical network topology consisting of regular peers and super-peers, which act as local servers for the subsets of regular peers (Fig.1(c)). For example, a hybrid overlay might consist of the super-peers forming a structured network which serves as a backbone for the whole overlay, enabling an efficient communication among the super-peers themselves. Hybrid overlays have an advantage over simple centralized networks since the super-peers can be dynamically replaced by regular peers, hence do not constitute single points of failure, but have the benefits of centralized overlays.

Routing

Peer-to-peer overlay networks enable the peers to communicate with one another even if the communicating peers do not know their addresses in the underlying network. For example, in an overlay deployed on the Internet, a peer can communicate with another peer without knowing its IP address. The way it is achieved in the overlays is by routing overlay messages. Each overlay message originates at a source and is forwarded by the peers in the overlay until the message reaches one or more destinations. A number of routing schemes have been proposed.

Maintenance

Peer-to-peer systems are commonly deployed in environments characterized by high dynamicity, peers can depart or join the system at any time. These continuous joins and departures are commonly referred to as *churn*. Instead of gracefully departing from the network peers can also abruptly fail or the network connection with some of its neighbors may be closed. In all of these cases the changes in the routing tables may adversely affect the performance of the system. The overlay topology needs to be *maintained* to guarantee message delivery and routing efficiency.

There are two main approaches to overlay maintenance: proactive and reactive. In *proactive maintenance* peers periodically update their routing tables so that they satisfy the overlay topology invariants. For example, Chord periodically runs a “stabilization” protocol to ensure that every peer is linked to other peers at exponentially increasing distance. This ensures routing efficiency. To ensure message delivery each Chord peer maintains connections to its immediate predecessor and successor on the Chord ring.

In contrast to proactive maintenance, *reactive maintenance* is triggered immediately after the detection of a peer failure or peer departure. The missing entry in the routing table is replaced with a new one by sending a connect request to an appropriate peer.

Failures and departures of peers are detected in two ways: by probing or through usage. In probe-based failure detection each peer continuously runs a ping response protocol with each of its neighbors. When ping timeouts occur repeatedly the neighbor is considered to be down and is removed from the routing table. In usage-based failure detection when a message is sent to a neighbor but not acknowledged within a timeout, the neighbor is considered to have failed.

The more neighbors a peer must maintain the higher the bandwidth overhead incurred by the maintenance protocol. In modern structured overlays maintenance bandwidth typically scales as O(log(N)) in terms of the network size.

(Abridged)

Recommended Reading

1. Aberer K. P-Grid: A self-organizing access structure for P2P information systems. In Proc. Int.Conf. on Cooperative Inf. Syst., 2001.

2. Androutsellis-Theotokis S. and Spinellis D. A survey of peer-to-peer content distribution technologies. ACM Comput. Surv., 36 (4):335–371, December 2004.

3. Bittorrent. http://www.bittorrent.com/.

4. Clarke I., Sandberg O., Wiley B., and Hong T.W. Freenet: A distributed anonymous information storage and retrieval system. In Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability, 2001.

5. Gnutella Homepage. http://www.gnutella.wego.com/.

6. Kazaa Homepage. http://www.kazaa.com/.

7. Manku G.S., Bawa M., and Raghavan P. Symphony: Distributed hashing in a small world. In Proc. 4th USENIX Symp. on Internet Tech. and Syst., 2003.

8. Maymounkov P. and Mazie`res D. Kademlia: A peer-to-peer information system based on the XOR metric. In 2429 of Lecture Notes in Computer Science, 2002, pp. 53–65.

9. Napster. http://www.napster.com/.

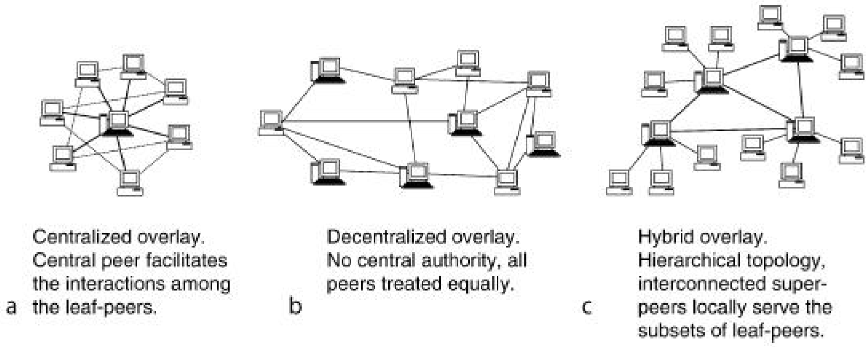
10. Ripeanu M., Foster I., and Iamnitchi A. Mapping the Gnutella network: Properties of large-scale peer-to-peer systems and implications for system design. IEEE Internet Comput. J., 6(1), August 2002.

11. Risson J. and Moors T. Survey of research towards robust peer-to-peer networks: Search methods. Comput. Netw., 50(17):3485–3521, 2006.

12. Skype Homepage. http://www.skype.com/.

13. Stoica I., Morris R., Karger D.R., Kaashoek M.F., and Balakrishnan H. Chord: A scalable peer-to-peer lookup service for internet applications. In Proc. ACM Int. Conf. of the on Data Communication, 2001, pp. 149–160.

14. YaCyPeer. http://www.yacyweb.de/. Peer



Centralized overlay. Central peer facilitates the interactions among the leaf-peers.

Decentralized overlay. No central authority, all peers treated equally.

Hybrid overlay. Hierarchical topology, interconnected super-peers locally serve the subsets of leaf-peers.

Peer to Peer Overlay Networks: Structure, Routing and Maintenance. Figure 1. Examples of peer-to-peer overlays.

Answer the following questions:

1) What does *overlay* network mean?
2) What is the basic difference between *data-sharing/bandwidth-sharing/CPU-sharing*?
3) What is the disadvantage of *centralized* overlays?
4) Which type of the mentioned overlays can exist without any governing authority and why?
5) What is the advantage of *hybrid* overlays?
6) What does *churn* refer to?
7) Why does the overlay topology need to be maintained?
8) When is a peer considered to be down in *probe-based* failure detection?
9) When is a peer considered to be down in *usage-based* failure detection?
10) Does the bandwidth overhead increase or decrease with a higher number of peers and why?

Match the following terms with their definitions:

1) central peer
2) leaf-peer
3) proactive maintenance
4) super-peer
5) reactive maintenance
6) topology

a) an approach to overlay maintenance in which a missing entry in the routing table is immediately replaced with a new one upon a peer failure or departure
b) component that supervises all communication in a subset of a given P2P network
c) an arrangement and interconnections of the elements of a network
d) component that itself does not have any knowledge of the other peers
e) component that supervises all communication in a given P2P network
f) an approach to overlay maintenance in which peers repeatedly update their routing tables

Mark the following statements as *true* or *false:*

1) Peer-to-peer overlay networks enable participating peers to find the other peers by their IP addresses.
2) Nowadays, unstructured peer-to-peer overlays are very frequently used.
3) BitTorrent is an example of a bandwidth-sharing network.
4) Hybrid overlays have advantage over simple centralized networks.
5) Overlay networks do not need to be maintained.
6) The only way to detect failures and departures of peers is through probing.

Vocabulary

abruptly [ə'brʌp*t*.li] 🔊 – náhle, neočekávaně

ad hoc [ˌæd'hɒk] US [–'hɑːk] 🔊 🔊 – pro tento případ, za tímto účelem

adversely ['æd.vɜː.sli] US [æd'vɝː–] – nepříznivě

aforementioned [ə'fɔːˌmen.*t*ʃᵊnd] US [–'fɔːr–] 🔊 – výše zmíněný

arguably ['ɑːg.ju.ə.bli] US ['ɑːrg–] 🔊 – pravděpodobně

backbone ['bæk.bəʊn] US [–boʊn] 🔊 – páteř

bandwidth ['bænd.wɪtθ] 🔊 – rozsah

behaviour [bɪ'heɪ.vjəʳ] US [–vjɚ] 🔊 – chování

bottleneck ['bɒt.l̩.nek] US ['bɑː.t̬l̩–] 🔊 – zúžení, nesnáz

currently (X actually) ['kʌr.ᵊnt.li] US ['kɝː–] ['æk.tʃu.ə.li] 🔊 – aktuálně, v současné době (X skutečně, doopravdy)

dependency [dɪ'pen.dᵊn*t*.si] 🔊 – závislost

disjoint [dɪs'dʒɔɪn.t] 🔊 – rozpojený, disjunkční

distinct/distinctive [dɪ'stɪŋkt] 🔊 [dɪ'stɪŋk.tɪv] 🔊 – zřetelný, odlišný

drawback ['drɔː.bæk] US ['drɑː–] 🔊 – nevýhoda, nedostatek

efficient [ɪ'fɪʃ.ᵊnt] 🔊 – výkonný, rychlý

hence [hen*t*s] 🔊 – proto, tudíž

idle ['aɪ.dl] 🔊 – nečinný

intrinsically [ɪn'trɪn.zɪ.kli] 🔊 – skutečně, vnitřně

invariant [in'veə.ri.ənt] US [in'ver.i.–] 🔊 – neměnný, stálý

maintenance ['meɪn.tɪ.nən*t*s] 🔊 – údržba, správa

mutual ['mjuː.tʃu.əl] 🔊 – vzájemný, společný

node [nəʊd] US [noʊd] 🔊 – uzel, průsečík

overhead ['əʊ.və.hed] US ['oʊ.vɚ–] 🔊 – zátěž, náklady

peer [pɪəʳ] US [pɪr] 🔊 – vrstevník, druh (stejného postavení)

predecessor ['priː.dɪˌses.əʳ] US ['pred.ə.ses.ɚ] 🔊 🔊 – předchůdce

purpose ['pɜː.pəs] US ['pɝː–] 🔊 – účel

retrieval [rɪ'triː.vᵊl] 🔊 – získávání, vyhledávání

robust [rə*ʊ*'bʌst] US [roʊ–] 🔊 – silný, odolný

routing [ruːt.ɪŋ] US [raʊt.ɪŋ] – směrování

scalability [skeɪ.lə'bɪ.lə.tɪ] US [–ə.t̬ɪ] 🔊 – škálovatelnost

scalable [skeɪ.lə.bl] 🔊 – škálovatelný

scheme [skiːm] 🔊 – návrh, plán, schéma

scope of [skəʊp] US [skoʊp] 🔊 – rozsah, oblast

significantly [sɪg'nɪf.ɪ.kənt.li] 🔊 – významně, podstatně

since (conjunction) [sɪn*t*s] 🔊 – protože, poněvadž

sparsely ['spɑː.sli] US ['spɑːr–] – řídce

successor [sək'ses.əʳ] US [–ɚ] 🔊 – následovník, nástupce

thus [ðʌs] 🔊 – tudíž, a tak

to account for [ə'kaʊnt] 🔊 – (z)odpovídat, vysvětlit

to achieve [ə'tʃiːv] 🔊 – dosáhnout,

to acknowledge [ək'nɒl.ɪdʒ] US [–'nɑː.lɪdʒ] 🔊 – potvrdit, brát na vědomí, uznat

to aim at [eɪm] 🔊 – mířit na, směřovat k

to assume [ə'sjuːm] US [–'suːm] 🔊 – předpokládat, domnívat se

to boost [buːst] 🔊 – podpořit, posílit

to deploy [dɪ'plɔɪ] 🔊 – použít, využít

to emerge [ɪ'mɜːdʒ] US [–'mɝːdʒ] 🔊 – objevit se, vzniknout

to enable [ɪ'neɪ.bl] 🔊 – umožnit, dovolit

to facilitate [fə'sɪl.ɪ.teɪt] 🔊 – usnadnit, napomáhat

to forward ['fɔː.wəd] US ['fɔːr.wɚd] 🔊 – přeposlat, posunout dopředu

to incur [ɪn'kɜːʳ] US [–'kɝː] 🔊 – způsobit

to occur [ə'kɜːʳ] US [–'kɝː] 🔊 – vyskytovat se, objevovat se

to participate [pɑː'tɪs.ɪ.peɪt] US [pɑːr'tɪs.ə–] 🔊 – (z)účastnit se

to probe [prəʊb] US [proʊb] 🔊 – prozkoumat, zjišťovat

to scale [skeɪl] 🔊 – měnit velikost, odstupňovat

to subscribe [səb'skraɪb] 🔊 – předplatit, odebírat

to trade off ['treɪd.ɒf] US [–ɑːf] 🔊 – přijmout/udělat kompromis

to trigger ['trɪg.əʳ] US [–ɚ] 🔊 – spustit

topology [tə'pɒ.lə.dʒɪ] US [tə'pɑː.lə.dʒɪ] 🔊 – topologie

underlying [ˌʌn.də'laɪ.ɪŋ] US [–dɚ–] 🔊 – základový, zásadní

wasteful ['weɪs*t*.fᵊl] 🔊 – nešetrný, nadměrný

Phrases

to pave the way [peɪv] 🔊 – vydláždit cestu, připravit půdu

XML

MICHAEL RYS
Microsoft Corporation, Sammamish, WA, USA

Definition

The Extensible Markup Language or XML for short is a markup definition language defined by a World Wide Web Consortium Recommendation that allows annotating textual data with tags to convey additional semantic information. It is extensible by allowing users to define the tags themselves.

Historical Background

XML was developed as a simplification of the ISO Standard General Markup Language (SGML) in the mid 1990s under the auspices of the World Wide Web Consortium (W3C). Some of the primary contributors were Jon Bosak of Sun Microsystems (the working group chair), Tim Bray (then working at Textuality and Netscape), Jean Paoli of Microsoft, and C. Michael Sperberg-McQueen of then the University of Chicago. Initially released as a version 1.0 W3C recommendation on 10 Feb. 1998, XML has undergone several revisions since then. The latest XML 1.0 recommendation edition is the fourth edition as of this writing. A fifth edition is currently undergoing review. The fifth edition is adding some functionality into XML 1.0 that was part of the XML 1.1 recommendation, which has achieved very little adoption. Based on the XML recommendation, a whole set of related technologies has been developed, both at the W3C and other places. Some technologies are augmenting the core syntactic XML recommendation such as the XML Namespaces and XML Information Set recommendations, others are building additional infrastructure on it such as the XML Schema recommendation or the XSLT, XPath and XQuery family of recommendations. Since XML itself is being used to define markup vocabularies, it also forms the basis for vertical industry standards in manufacturing, finance and other areas, including standard document formats such as XHTML, DocBook, Open Document Format (ODF) and Office Open XML (OOXML) and forms the foundation of the web services infrastructure.

Foundations

XML’s markup format is based on the notion of defining well-formed documents and is geared towards human-readability and international usability (by basing it on Unicode). Among its design goals were ease of implementation by means of a simple specification, especially compared to its predecessor, the SGML specification, and to make it available on a royalty-free basis to both implementers and users. Well-formed documents basically contain markup elements that have a begin tag and an end tag as in the example below:

```
< tag > character data < /tag >
```

Element tags can have attributes associated with it that provide information about the tag, without interfering with the flow of the textual character data that is being marked up:

```
<tag attribute1=”value” attribute2=”42”>
        character data </tag>
```

Processing instructions can be added to convey processing information to XML processors and comments can be added. And comments can be added for commenting and documentation purposes. = They follow different syntactic forms than element tags and can appear anywhere in a document, except within the begin tag and end tag tokens themselves:

```
<?xml-stylesheet type=”application/xslt + xml”
        href=”#style1”? >
<!- This is a comment –>
```

A well-formed document must also have exactly one top-level XML element, and can contain several processing instructions and comments on the top-level next to the element.
The order among these elements is information-bearing, since they are meant to mark up an existing document flow. Thus, the following two well -formed XML documents are not the same:

```
<doc><element> value1 </element>
<element> value2 </element></doc>
<doc><element> value2 </element>
<element> value1 </element></doc>
```

The XML information set recommendation defines an abstract data model for these syntactic components, introducing the notion of document information items for a document, element information items for element tags, attribute information items for their attributes, character information items for the marked up character data etc.
The XML namespace recommendation adds the ability to scope an element tag name to a namespace URI, to provide the ability to scope markup vocabularies to a domain identifier.
Besides defining an extensible markup format, the XML recommendation also provides a mechanism to

constrain the XML document markup to follow a certain grammar by restricting the allowed tag names and composition of element tags and attributes with document type declarations (DTDs). Documents that follow the structure given by DTDs are not only well-formed but also valid. Note that the XML Schema recommendation provides another mechanism to constrain XML documents.

Finally, XML also provides mechanisms to reuse parts of a document (down to the character level) using so called entities.

For more information about XML, please refer to the recommended reading section.

Key Applications

While XML was originally designed as an extensible document markup format, it has quickly taken over tasks in other areas due to the wide-availability of free XML parsers, its readability and flexibility. Besides the use for document markup, two of the key application scenarios for XML are the use for interoperable data interchange in loosely-coupled systems and for ad hoc modeling of semi-structured data.

XML's first major commercial applications actually have been to describe the data and, with DTDs or XML schema formats, structures of messages that are being exchanged between different computer systems in application to application data exchange scenarios and web services. XML is not only being used to describe the message infrastructure format and information such as the SOAP protocol, RSS or Atom formats, but also to describe the structure and data of the message payloads. Often, XML is also used in more ad hoc micro-formats for more REST-ful web services.

At the same time that XML was being developed, several researcher groups were looking into data models that were less strict than relational, entity-relationship and object-oriented models, by allowing instance based properties and heterogeneous structures. XML's tree model provides a good fit to represent such semi-structured, hierarchical properties, and its flexible format is well–suited to model the sparse properties and rapidly changing structures that are often occurring in semi- structured data. Therefore, XML has often been used to model semi-structured data in data modeling.

XML support has been added to databases on form of either pure XML databases or by extending existing database platforms such as relational database systems to enable databases to manage XML documents serving all these three application scenarios.

(Abridged)

Recommended Reading

1. Namespaces in XML 1.0, latest edition. Available at: http://www.w3.org/TR/xml-names
2. Wikipedia entry for XML. Available at: http://en.wikipedia.org/wiki/XML
3. XML 1.0 information Set, latest edition. Available at: http://www.w3.org/TR/xml-infoset
4. XML 1.0 recommendation, latest edition. Available at: http://www.w3.org/TR/xml
5. XML 1.1 recommendation, latest edition. Available at: http://www.w3.org/TR/xml11

Answer the following questions:

1) What does XML stand for?
2) What does *extensible* mean in this context?
3) What was the predecessor of XML?
4) What were some of the design goals of XML?
5) What are some of the XML-related technologies?
6) Why is basing XML on Unicode beneficial?
7) What are the key application scenarios of XML that the text mentions?
8) What does it mean that the order of elements (line 92) is information-bearing?
9) What is a *valid* XML document?

Match the following terms with their definitions:

1) element
2) attribute
3) processing instructions
4) comment

a) a part of the document ignored by the processor
b) a part of the document influencing the way the data is parsed
c) a part of the document marked by the begin and end tags
d) way of providing additional information about a tag

Mark the following statements as *true* or *false:*

1) A well-formed document can contain only one processing instruction.
2) Comments follow the same syntactic forms as element tags.
3) A well-formed document must have exactly one top-level element.
4) Every XML document must be provided with DTDs.
5) Free XML parsers are widely available.
6) XML is rarely used in data modeling.

Vocabulary

ad hoc [ˌæd'hɒk] US [–'hɑːk] 🔊 🔊 – pro tento případ, za tímto účelem

attribute ['æt.rɪ.bjuːt] 🔊 – atribut, rys, vlastnost

contributor [kən'trɪb.jʊ.təʳ] US [–jə.t̬ɚ] 🔊 – přispěvatel

extensible [ɪks'ten.sə.bl] 🔊 – rozšiřitelný

foundation [faʊn'deɪ.ʃᵊn] 🔊 – základ

goal [gəʊl] US [goʊl] 🔊 – cíl, záměr

heterogeneous [ˌhet.ᵊr.ə'dʒiː.ni.əs] US [ˌhet̬.ə.roʊ–] 🔊 – různorodý, heterogenní

initially [ɪ'nɪʃ.ᵊl.i] 🔊 – původně, zpočátku

interoperable [ɪn.tə'ɒ.pə.rə.bl] US [–t̬ɚ'ɑː–] 🔊 – schopný spolupráce

item ['aɪ.təm] US [–t̬əm] 🔊 – položka, člen

flow [fləʊ] US [floʊ] 🔊 – tok

loose [luːs] 🔊 – volný, neurčitý

payload ['peɪ.ləʊd] US [–loʊd] 🔊 – zatížení, vytížení

recommendation [ˌrek.ə.men'deɪ.ʃᵊn] 🔊 – doporučení

royalty–free ['rɔɪ.əl.ti friː] 🔊 🔊 – bez poplatku za autorská práva

set (adjective) [set] 🔊 – stanovený, určený

sparse [spɑːs] US [spɑːrs] 🔊 – vzácný, rozptýlený, řídký

tag [tæg] 🔊 – značka, visačka

therefore ['ðeə.fɔːʳ] US ['ðer.fɔːr] 🔊 – tudíž, tedy

token ['təʊ.kᵊn] US ['toʊ–] 🔊 – znak, příznak

to achieve [ə'tʃiːv] 🔊 – dosáhnout

to augment [ɔːg'ment] US [ɑːg–] 🔊 – rozšířit, rozmnožit

to constrain [kən'streɪn] 🔊 – omezovat, přinutit

to convey [kən'veɪ] 🔊 – zprostředkovat, vyjádřit

to gear (towards) [gɪəʳ] US [gɪr] 🔊 – směřovat, vést k

to interfere [ˌɪn.tə'fɪəʳ] US [–t̬ɚ'fɪr] 🔊 – překážet, rušit

to occur [ə'kɜːʳ] US [–'kɝː] 🔊 – vyskytovat se, objevovat se

to release [rɪ'liːs] 🔊 – vydat

to restrict [rɪ'strɪkt] 🔊 – omezit

to scope [skəʊp] US [skoʊp] 🔊 – přidružit

to take over [teɪk] 🔊 ['əʊ.vəʳ] US ['oʊ.vɚ] 🔊 – převzít

to undergo [ˌʌn.də'gəʊ] US [–dɚ'goʊ] 🔊 – projít

Phrases

by means of – prostřednictvím

under the auspices of ['ɒː.spi.sis] US [ɑː.spi.sis] 🔊 – pod záštitou

Web Advertising

VANJA JOSIFOVSKI[1], ANDREI BRODER[2]
[1]Uppsala University, Uppsala, Sweden
[2]Yahoo! Research, Santa Clara, CA, USA

Definition

Web advertising aims to place relevant ads on web pages. As in traditional advertising, most of the advertising on the web can be divided into brand advertising and direct advertising. In the majority of cases, brand advertising is achieved by banners – images or multimedia ads placed on the web page. As opposed to traditional brand advertising, on the web the user can interact with the ad and follow the link to a website of advertisers choice. Direct advertising is mostly in the form of short textual ads on the side of the search engine result pages and other web pages. Finally, the web allows for other types of advertising that are hybrids and cross into other media, such as video advertising, virtual worlds advertising, etc. Web advertising systems are built by implementing information retrieval, machine learning, and statistical techniques in a scalable, low-latency computing platform capable of serving billions of requests a day and selecting from hundreds of millions of individual advertisements.

Historical Background

The Web emerged as a new publishing media in the late 1990. Since then, the growth of web advertising has paralleled the growth in the number of web users and the increased time people spend on the Web. Banner advertising started with simple placement of banners on the top of the pages at targeted sites, and has since evolved into an elaborate placement scheme that targets a particular web user population and takes into account the content of the web pages and sites. Search advertising has its beginnings in the specialized search engines for ad search. Combining the web search with ad search was not something web users accepted from the start, but has become mainstream today. Furthermore, today's search advertising platforms have moved from simply asking the advertiser to provide a list of queries for which the ad is to be shown to employing variety of mechanisms to automatically learn what ad is appropriate for which query. Today's ad platforms are large, scalable and reliable systems running over clusters of machines that employ state-of-the-art information retrieval and machine learning techniques to serve ads at rates of tens of thousands times a second. Overall, the technical complexity of the advertising platforms rivals those of the web search engines.

Foundations

Web advertising spans Web technology, sociology, law, and economics. It has already surpassed some traditional mass media like broadcast radio and it is the economic engine that drives web development. It has become a fundamental part of the web eco-system and touches the way content is created, shared, and disseminated – from static html pages to more dynamic content such as blogs and podcasts, to social media such as discussion boards and tags on shared photographs. This revolution promises to fundamentally change both the media and the advertising businesses over the next few years, altering a $300 billion economic landscape.

As in classic advertising, in terms of goals, web advertising can be split into brand advertising whose goal is to create a distinct favorable image for the advertiser's product, and direct-marketing advertising that involves a ''direct response'': buy, subscribe, vote, donate, etc, now or soon.

In terms of delivery, there are two major types:

1. *Search advertising* refers to the ads displayed alongside the ''organic'' results on the pages of the Internet search engines. This type of advertising is mostly direct marketing and supports a variety of retailers from large to small, including micro-retailers that cover specialized niche markets.
2. *Contextual advertising* refers to ads displayed alongside some publisher-produced content, akin to traditional ads displayed in newspapers. It includes both brand advertising and direct marketing. Today, almost all non-transactional web sites rely on revenue from content advertising. This type of advertising supports sites that range from individual bloggers and small community pages, to the web sites of major newspapers. There would have been a lot less to read on the web without this model!

From an ad-platform standpoint, both search and content advertising can be viewed as a matching problem: a stream of queries or pages is matched in real time to a supply of ads. A common way of measuring the performance of an ad-platform is based on the clicks on the placed ads. To increase the number of clicks, the ads placed must be relevant to the user's query or the page and their general interests.

There are several data engineering challenges in the design and implementation of such systems.

The first challenge is the volume of data and transactions: Modern search engines deal with tens of billions of pages from hundreds of millions of publishers, and billions of ads from tens of millions of advertisers. Second, the number of transactions is huge: billions of searches and billions of page views per day. Third, to deal with that, there is only a very short processing time available: when a user requests a page or types her query, the expectation is that the page, including the ads, will be shown in real time, allowing for at most a few tens of milliseconds to select the best ads.

To achieve such performance, ad-platforms usually have two components: a *batch processing component* that does the data collection, processing, and analysis, and a *serving component* that serves the ads in real time. Although both of these are related to the problems solved by today's data management systems, in both cases existing systems have been found inadequate for solving the problem and today's ad-platforms require breaking new grounds.

The batch processing component of an ad-system processes collections of multiple terabytes of data. Usually the data are not shared and a typical processing lasts from a few minutes to a few hours over a large cluster of hundreds, even thousands of commodity machines.

The serving component of an advertising platform must have high throughput and low latency. To achieve this, in most cases the serving component operates over read-only copy of the data replaced occasionally by the batch component. The ads are usually pre-processed and matched to an incoming query or page. The serving component has to implement reasoning that, based on a variety of features of the query/page and the ads, estimates the top few ads that have the maximum expected revenue within the constraints of the marketplace design and business rules associated to that particular advertising opportunity.

Several different techniques can be used to select ads. When the number of ads is small, linear programming can be used to optimize for multiple ads grouped in *slates*. Banner ads are sometimes placed using linear programming as optimization technique [1,2]. In textual advertising the number of ads is too large to use linear programming. One alternative in such case is to use unsupervised methods based on information retrieval ranking [4,11]. Scoring formulas can be adapted to take into account the specificity of ads as documents: short text snippets with distinct multiple sections. Information retrieval scoring can be augmented with supervised ranking mechanisms that use the click logs or editorial judgements [9] to learn what ads work for a particular page/user/query. Here, the system needs to explore the space of possible matches. Some explore-exploit methods have been adapted to minimize the cost of exploration based on one-arm-bandit algorithms [3].

In summary, today's search and content advertising platforms are massive data processing systems that apply complicated data analysis and machine learning techniques to select the best advertising for a given query or page. The sheer scale of the data and the real-time requirements make this problem a very challenging task. Today's implementations have grown quickly and often in an ad-hoc manner to deal with a $15 billion fast growing market, but there is a need for improvement in almost every aspect of these systems as they adapt to even larger amounts of data, traffic, and new business models in web advertising.

(Abridged)

Recommended Reading

1. Abe N. Improvements to the linear programming based scheduling of Web advertisements. J. Electron. Commerce Res., 5(1):75–98, 2005.

2. Abrams Z., Mendelevitch O., and Tomlin J.A. Optimal delivery of sponsored search advertisements subject to budget constraints. In Proceedings of the ACM EC '07, 2007, pp. 272–278.

3. Agarwal D., Chakrabarti D., Josifovski V., and Pandey S. Bandits for taxonomies: a model based approach. In Proceedings of the SIAM SDM 2007, 2007.

4. Broder A., Fontoura M., Josifovski V., and Riedel L. A semantic approach to contextual advertising. In Proc. 33rd Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, 2007, pp. 559–566.

5. Fain D. and Pedersen J. Sponsored search: a brief history. In Proc. 2ndWorkshop on Sponsored Search Auctions.Web publication, 2006.

6. Group C.S. Community systems research at yahoo! SIGMOD Rec., 36(3):47–54, 2005.

7. Jones R. and Fain D.C. Query word deletion prediction. In Proc. 26th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, 2003, pp. 435–436.

8. Jones R., Rey B., Madani O., and Greiner W. Generating query substitutions. In Proc. 15th Int. World Wide Web Conference, 2006, pp. 387–396.

9. Lacerda A., Cristo M., Andre M.G., Fan W., Ziviani N., and Ribeiro-Neto B. Learning to advertise. In Proc. 32nd Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, 2006, pp. 549–556.

10. Pike R., Dorward S., Griesemer R., and Quinlan S. Interpreting the data: parallel analysis with Sawzall. Sci. Program. J., 13(4): 277–298, 2005.

11. Ribeiro-Neto B., Cristo M., Golgher P.B., and de Moura E.S. Impedance coupling in content-targeted advertising. In Proc. 31st Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, 2005, pp. 496–503.

Answer the following questions:

1) What are *banners*?
2) What are some of the types of advertising mentioned in the text?
3) What is the principle of *brand advertising*?
4) What techniques and technologies does web advertising implement?
5) When is it beneficial to use linear programming?
6) What are *niche markets*?
7) What is the difference between *supervised* and *unsupervised* methods used in textual advertising?
8) What is a common way of measuring an ad-platform performance?
9) Why is search and content advertising such a challenge?

Match the following terms with their definitions:

1) batch processing component
2) search advertising
3) contextual advertising
4) serving component

a) ads displayed alongside some published content
b) ad-platform component that deals with large data collection, time-consuming processing and analysis
c) ads displayed alongside the results provided by a search engine
d) ad-platform component that provides relevant ads in real time

Mark the following statements as *true* or *false*:

1) The technical complexity of the advertising platforms is similar to that of web search engines.
2) The serving component of an advertising platform must have high throughput and high latency.
3) Brand advertising is usually achieved by banners.
4) Only search advertising can be viewed as a matching problem.
5) Today’s ad platforms are large, non-scalable and reliable systems.
6) Advertising is the economic engine driving web development.
7) Contextual advertising is similar in principle to traditional newspapers advertising.

Vocabulary

ad hoc [ˌæd'hɒk] US [–'hɑːk] 🔊 🔊 – pro tento případ, za tímto účelem

akin to [ə'kɪn] 🔊 – podobný

alongside [əˌlɒŋ'saɪd] US [ə'lɑːŋ.saɪd] 🔊 – vedle, podél

appropriate [ə'prəʊ.pri.ət] US [–'proʊ–] 🔊 – vhodný, odpovídající

banner ['bæn.əʳ] US [–ɚ] 🔊 – transparent, prapor

brand [brænd] 🔊 – značka

capable of ['keɪ.pə.bl] 🔊 – schopný, umožňující

cluster ['klʌs.təʳ] US [–tɚ] 🔊 – seskupení

delivery [dɪ'lɪv.ᵊr.i] US [–ɚ–] 🔊 – doručení, dodání

distinct [dɪ'stɪŋkt] 🔊 – zřetelný, odlišný

economic (X economical) [iː.kə'nɒm.ɪk] US [–'nɑː.mɪk] 🔊 🔊 [ˌiː.kə'nɒm.ɪ.kᵊl] US [–'nɑː.mɪ–] 🔊 – ekonomický (X úsporný)

elaborate [ɪ'læb.ᵊr.ət] US [–ɚ–] 🔊 – propracovaný

furthermore [ˌfɜː.ðə'mɔːʳ] US ['fɝː.ðɚ.mɔːr] 🔊 – dále, navíc

inadequate [ɪ'næd.ɪ.kwət] 🔊 – nepřiměřený, nedostatečný

judgement ['dʒʌdʒ.mənt] 🔊 – úsudek

latency ['leɪ.tᵊn*t*.si] 🔊 – zpoždění

log [lɒg] US [lɑːg] 🔊 – záznam

multiple ['mʌl.tɪ.pl] US [–t̬ɪ–] 🔊 – vícenásobný, mnohočetný

one-arm(ed)-bandit [ˌwʌn.ɑːmd'bæn.dɪt] US [–ɑːrmd–] 🔊 – hrací automat

overall [ˌəʊ.və'rɔːl] US [ˌoʊ.vɚ'ɑːl] 🔊 – celkově

performance [pə'fɔː.mən*t*s] US [pɚ'fɔːr–] 🔊 – výkon

query ['kwɪə.ri] US ['kwɪr.i] 🔊 🔊 – dotaz

reasoning ['riː.zᵊn.ɪŋ] 🔊 – usuzování, logické myšlení

reliable [rɪ'laɪə.bl] 🔊 – spolehlivý

retailer ['riː.teɪ.ləʳ] US [–lɚ] – maloobchodník

retrieval [rɪ'triː.vᵊl] 🔊 – získávání, vyhledávání

revenue ['rev.ᵊn.juː] US [–ə.nuː] 🔊 – výnos

scalable [skeɪ.lə.bl] 🔊 – škálovatelný

scale [skeɪl] 🔊 – rozsah, škála

scoring formula [skɔːr.ɪŋ] US [skɔːr.ɪŋ] ['fɔː.mjʊ.lə] US ['fɔːr–] 🔊 – vyhodnocovací vzorec

search engine [sɜːtʃ] US [sɝːtʃ] 🔊 ['en.dʒɪn] 🔊 – vyhledávač

slate [sleɪt] 🔊 – tabulka (nejvhodnějších kandidátů)

snippet ['snɪp.ɪt] 🔊 – kousek, výstřižek

standpoint ['stænd.pɔɪnt] 🔊 – hledisko, stanovisko

state-of-the-art [steɪt əv ðiː ɑːt] – vyspělý, v současné době nejlepší

stream [striːm] 🔊 – proud, sled

supply [sə'plaɪ] 🔊 – dodávka, nabídka

targeted ['tɑː.gɪt.ɪd] US ['tɑːr–] – cílený

throughput ['θruː.pʊt] 🔊 – propustnost

to achieve [ə'tʃiːv] 🔊 – dosáhnout

to allow for [ə'laʊ] 🔊 – počítat s, brát v úvahu

to alter ['ɒl.təʳ] US ['ɑːl.t̬ɚ] 🔊 – měnit, upravit

to augment [ɔːg'ment] US [ɑːg–] 🔊 – rozšířit, rozmnožit

to disseminate [dɪ'sem.ɪ.neɪt] 🔊 – šířit

to divide into [dɪ'vaɪd] 🔊 – rozdělit

to donate [də*ʊ*'neɪt] US ['doʊ.neɪt] 🔊 – věnovat, darovat

to evolve into [ɪ'vɒlv] US [–'vɑːlv] 🔊 – vyvinout se

to exploit [ɪk'splɔɪt] 🔊 🔊 – využít, zužitkovat

to interact with [ˌɪn.tə'rækt] US [–t̬ɚ'ækt] 🔊 – vzájemně reagovat, působit

to match [mætʃ] 🔊 – párovat, hodit se k

to optimize ['ɒp.tɪ.maɪz] US ['ɑːp.tə–] 🔊 – přizpůsobit, optimalizovat

to parallel ['pær.ə.lel] US ['per–] 🔊 – odpovídat, podobat se

to provide [prə'vaɪd] 🔊 – poskytnout

to range from...to [reɪndʒ] 🔊 – v rozsahu od do

to rank [ræŋk] 🔊 – seřadit (dle kritérií)

to rely on [rɪ'laɪ] 🔊 – spoléhat na

to require [rɪ'kwaɪəʳ] US [–'kwaɪr] 🔊 – vyžadovat, nutně potřebovat

to rival ['raɪ.vᵊl] 🔊 – konkurovat, vyrovnat se

to span [spæn] 🔊 – obsáhnout

to split into [splɪt] 🔊 – rozdělit

to subscribe [səb'skraɪb] 🔊 – předplatit, odebírat

to surpass [sə'pɑːs] US [sɚ'pæs] 🔊 – překonat, předčít

volume (of data) ['vɒl.juːm] US ['vɑːl–] 🔊 – objem

Phrases

as in – jako v

as opposed to [ə'pəʊzd] US [–'poʊzd] 🔊 – na rozdíl od

at most – v nejlepším případě

at rates of [reɪts] – rychlostí

to break new grounds – prozkoumat nové obzory

in terms of – co se týká, ve vztahu k

per day [pɜːʳ] US [pɝː] 🔊 – denně

to take into account [ə'kaʊnt] 🔊 – brát v úvahu

within the constraints of [kən'streɪnts] 🔊 – v mezích