

BSM 420 – BİLGİSAYAR MİMARİ

Deney Tasarımı
10.Hafta

Giriş (1 / 3)

Deney hiçbir zaman başarısızlıkla sonuçlanmaz. Kötü deney olumsuz bir örnek olur.

– Arthur Bloch - Yazar

Bilimin temel prensibi, tanımı şudur:

herhangi bir fikrin geçerliliğinin yegane testi deneydir.

– Richard P. Feynman - Fizikçi

- Amaç, minimum deney sayısı ile maksimum bilgi elde etmektir,
- Doğru analiz faktörleri ayıklamaya yardımcı olacaktır
- İstatistiksel teknikler, farklılıkların hatalardan kaynaklanıp kaynaklanmadığını belirlemeye yardımcı olacaktır

Giriş (2 / 3)

- Sıfır-maliyetli bir süreç değildir
 - Veri toplamak zaman ve çaba gerektirir
 - Sonuçları analiz etmek ve sunmak
 - Deney sayısı en aza indirilmelidir
- İyi deneysel tasarım aşağıdaki avantajları sağlar:
 - Her giriş değişkeninin etkilerini izole etme
 - Giriş değişkenlerinin etkileşimlerinin etkilerini belirleme
 - Deneysel hatanın büyüklüğünü belirleme
 - Minimum çabayla maksimum bilgi edinme

Giriş (3 / 3)

- Diğerlerini sabit tutarken sadece bir girişi değiştirin
 - Basit, ancak iki giriş değişkeni arasındaki olası etkileşimi yok sayar
- Giriş değişkenlerinin olası tüm kombinasyonlarını test edin
 - Etkileşim etkilerini belirleyebilir, ancak çok büyük olabilir
 - Örn: 4 seviyeli 5 faktör • $4^5 = 1024$ deney.
Ölçüm hatasını elde etmek için yineleme sayısı $1024 \times 3 = 3072$

İçindekiler

- Giriş
- Terminoloji
- Genel Hatalar
- Basit Tasarımlar
- Tam Faktöryel Tasarımları
 - 2^k Faktöriyel Tasarımları
- 2^{kr} Faktöriyel Tasarımları

Terminoloji (1 / 4)

- PC performansını ele alalım
 - Cpu seçimi: 6800, z80, 8086
 - Bellek boyutu: 512 KB, 2 MB, 8 MB
 - Disk sürücüler: 1-4
 - İş yükü: sekreterlik, yönetsel, bilimsel
 - Kullanıcılar: lise, kolej, mezun
- *Yanıt değişkeni* – sonuç veya ölçülen performans
 - Örn: görev/dk cinsinden çıkış (throughput) veya saniye cinsinden bir görev in yanıt süresi

Terminoloji (2 / 4)

- *Faktörler* – yanıtı etkileyen her değişken
 - Örn: CPU, bellek, diskler, iş yükü, kullanıcı
 - Ayrıca kestirimci (*predictor*) *değişkenleri* olarak ta adlandırılırlar
- *Düzeyler* – faktörler farklı değerler alabilir
 - ÖR: CPU 3, bellek 3, diskler 4, iş yükü 3, kullanıcılar
- *Birincil faktörler* – en önemli olanlar
 - Örn: CPU ve bellek

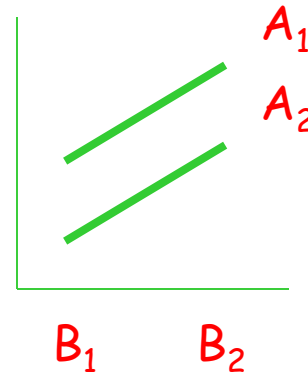
Terminoloji (3 / 4)

- *İkincil faktörler* – daha az öneme sahip
 - Örn: belki kullanıcı türü o kadar önemli değil
- *Yineleme* – Tüm veya bazı deneylerin tekrarı
 - Örn: üç kez çalıştırılırsa, üç yineleme
- *Tasarım* – yineleme özellikleri, faktörler, düzeyler
 - Örn: 5 yinelemeli yukarıdaki seviyelerde tüm faktörleri belirtin, böylece $3 \times 3 \times 4 \times 3 \times 3 = 324$ çarpı 5 yineleme 1215 toplam deney sayısı

Terminoloji (4 / 4)

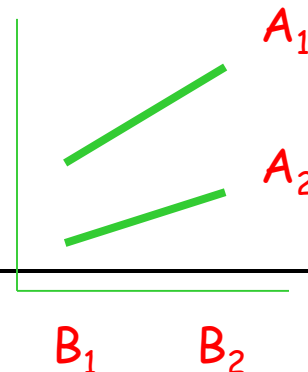
- *Etkileşim* – eğer biri diğerine bağımlılık gösterirse A ve B faktörleri etkileşir
 - Örn: A her zaman 2 arttığı için etkileşime giremeyen faktör

	A_1	A_2
B_1	3	5
B_2	6	8



- Örn: A değişimi B'ye bağlı olduğundan etkileşim faktörleri

	A_1	A_2
B_1	3	5
B_2	6	9



Anahat

- Giriş
- Terminoloji
- Genel Hatalar
- Basit Tasarımlar
- Tam Faktöryel Tasarımlar
 - 2^K Faktöriyel Tasarımları
- $2^K r$ Faktöriyel Tasarımları

Deneylerde Sık Yapılan Hatalar (1/2)

- *Deneyssel hata nedeniyle sapma yok sayılır.*
 - Ölçülen değerler ölçüm hatasına bağlı olarak rasgeleliğe sahiptir. Tüm sapmanın faktörlere bağlı olduğunu varsaymayın
- *Önemli parametrelerin kontrol edilmemesi*
 - Tüm parametreler (faktörler) listelenmeli ve tümü değişemese bile hesaba katılmalıdır.
- *Farklı faktörlerin etkileri izole değil.*
 - Aynı anda çeşitli faktörler değişebilir
 - Basit tasarımların kullanılması faydalı olabilir ama sorunları ortadan kaldırmaz

Deneylerde Sık Yapılan Hatalar (2/2)

- *Etkileşimler yok sayılması*
 - Genellikle bir faktörün etkisi diğerine bağlıdır. Örn: önbelleğin etkileri programın boyutuna bağlı olabilir.
- *Çok fazla deney yapılması*
 - Tüm faktörleri, tüm düzeyleri, tüm kombinasyonlarda, çalıştırmak yerine, adımlara bölmek
 - İlk adım, birkaç faktör ve birkaç düzey
 - Hangi faktörlerin önemli olduğunu belirleme
 - Faktör başına iki düzey
 - Daha sonraki tasarımda uygun olduğunda daha fazla düzey eklenir

İçindekiler

- Giriş
- Terminoloji
- Genel Hatalar
- Basit Tasarımlar
- Tam Faktöryel Tasarımları
 - 2^k Faktöriyel Tasarımları
- 2^{kr} Faktöriyel Tasarımları

Basit Tasarımlar

- Tipik yapılandırma ile başlayın
- Bir seferde bir faktörü değiştirin
- Örn: tipik bir PC z80 işlemcili, 2 MB RAM, 2 disk, üniversite öğrencisi tarafından üretilen iş yükü
 - diğer her şeyi sabit tutarak CPU'yu değiştir ve karşılaştır
 - diğer her şeyi sabit tutarak Disk sürücülerini değiştir ve karşılaştır
- Verilen i adet k faktörü her biri n_i düzeyine sahip
$$\text{Toplam} = 1 + \sum (n_i - 1) \text{ her } i = 1 \text{ den } k \text{ ya kadar}$$
- Örnek: iş istasyonu çalışmasında
$$1 + (3-1) + (3-1) + (4-1) + (3-1) + (3-1) + (3-1) = 14$$
- Ama etkileşimi göz ardı edebilir

Faktörlerin Etkileşimi Örneği

- Yanıt süresini vs. bellek boyutuna ve çoklu programlama derecesini ele alalım

Derece	32 MB	64 MB	128MB
1	0.25	0.21	0.15
2	0.52	0.45	0.36
3	0.81	0.66	0.50
4	1.50	1.45	0.70

- Eğer programlama derecesi 3, sabit olup 64 MB bellek değiştirilmez ise etkileşim farkedilmez
 - Örnek: derece 4, bellek ile doğrusal olmayan bir yanıt süresine sahiptir

Anahat

- Giriş
- Terminoloji
- Genel Hatalar
- Basit Tasarımlar
- Tam Faktöryel Tasarımlar
 - 2^K Faktöriyel Tasarımları
- 2^K r Faktöriyel Tasarımları

Tam Faktöryel Tasarımlar

- Tüm faktörlerin her düzeyinde mümkün olan her kombinasyon
- Verilen k faktörü, herbiri n_i düzeyine sahip
Toplam = $\prod n_i$ $i = 1$ den k ya
- Örnek: CPU tasarım çalışmasında
(3 CPU) (3 mem) (4 disk) (3 yük) (3 kullanıcı)
= 324 deney
- Avantaj her etkileşen bileşeni bulabilirsiniz
- Dezavantajı maliyetleri (zaman ve para), özellikle de birden fazla yineleme (daha sonra) gerekebilir
- Düzeyleri azaltmak, faktörleri azaltmak, tam faktöryelin bir kısmını çalıştırmak suretiyle maliyetleri azaltabilir:

2^k Faktöriyel Tasarımları

İşlerin %20'si kaynak tüketiminin %80'ini oluşturuyor.
– Pareto Yasası

- Her faktör de birçok düzeye sahip
 - Ör: ağ gecikmesinin kullanıcı yanıt süresi üzerindeki etkisi → test edilecek çok sayıda gecikme değeri vardır
- Genellikle, düzeylere göre performans sürekli artar veya azalır
 - Ör: yanıt süresi her zaman daha yüksek olur
 - Min ve max ile yönünü belirleyebilirsiniz
- Her faktör için, her düzeyde 2 alternatif seçin
 - 2^k faktöriyel tasarımları
- Daha sonra, hangi faktörlerin performansı en çok etkilediğini belirleyebilir ve ileri seviyede incelenebilir

2² Faktöriyel Tasarım (1/4)

- Sadece 2 faktörlü özel durum
 - Regresyon ile kolayca analiz edilebilir
- Örnek: Mem (4 veya 16 Mbyte) ve Önbellek (1 veya 2 Kbyte) için MIPS

	<u>Mem 4MB</u>	<u>Mem 16MB</u>
Önbellek 1 KB	15	45
Önbellek 2 KB	25	75

- x_A = eğer 4 Mbytes ise -1, 16 Mbyte ise +1
- X_B = eğer 1 Kbyte önbellek ise -1, 2 Kbyte ise +1
- Performans:

$$y = q_0 + q_a x_a + q_b x_b + q_{ab} x_a x_b$$

2² Faktöriyel Tasarım (2 / 4)

- Değiştirme:

$$15 = q_0 - q_A - q_B + q_{A\text{Çık}}$$

$$45 = q_0 + q_A - q_B - q_{A\text{Çık}}$$

$$25 = q_0 - q_A + q_B - q_{A\text{Çık}}$$

$$75 = q_0 + q_A + q_B + q_{A\text{Çık}}$$

(4 denklem
4 bilinmeyen)

- Çözüldüğünde :

$$y = 40 + 20x_A + 10x_B + 5x_Ax_B$$

- Yorum:

- Ortalama performans 40 MIPS, bellek etkisi 20 MIPS, önbellek etkisi 10 MIPS ve etkileşim etkisi 5 MIPS

2² Faktöriyel Tasarım (3 / 4)

Deny	a	b	y
1	-1	-1	y_1
2	1	-1	y_2
3	-1	1	y_3
4	1	1	y_4

$$y = q_0 + q_a x_a + q_b x_b + q_{ab} x_a x_b$$

- Dolayısıyla:

$$y_1 = q_0 - q_a - q_b + q_{ab}$$

$$y_2 = q_0 + q_a - q_b - q_{ab}$$

$$y_3 = q_0 - q_a + q_b - q_{ab}$$

$$y_4 = q_0 + q_a + q_b + q_{ab}$$

- Çözerek:

$$q_0 = \frac{1}{4}(y_1 + y_2 + y_3 + y_4)$$

$$q_a = \frac{1}{4}(-y_1 + y_2 - y_3 + y_4)$$

$$q_b = \frac{1}{4}(-y_1 - y_2 + y_3 + y_4)$$

$$q_{ab} = \frac{1}{4}(y_1 - y_2 - y_3 + y_4)$$

- q_a için a sütunu ile y sütunu çarpılarak elde edilir

- q_b ve q_{ab} de aynı şekilde hesaplanır

2² Faktöriyel Tasarım (4 / 4)

i	a	b	ab	y
1	-1	-1	1	15
1	1	-1	-1	45
1	-1	1	-1	25
1	1	1	1	75
160	80	40	20	Total

4020	10	5	Ttl/4
------	----	---	-------

- “i” sütunu 1 ile dolu
- “a” ve “b” kolonları 1, -1
- “ab” sütunu ise “a” ve “b” nin çarpımı

- Sütunları y ile çarpıp toplayarak
- Regresyon modelinde ağırlık vermek için 4’e böl
- Sonuç:

$$y = 40 + 20x_A + 10x_B + 5x_Ax_B$$

Sapmayı Tayin etme (1/3)

- Bir faktörün önemi: faktör cevabındaki toplam sapma oranı ile ölçülür
 - Bu nedenle, iki faktör cevabı sırasıyla %90 ve %5 ise, ikinci önemsizdir, ihmal edilebilir
 - Örn: TCP sürüm faktörü (Reno veya Sack) karşısında kapasite faktörü (768 Kbps veya 10 Mbps) karşı

- y örnek sapması

$$s_y^2 = \Sigma(y_i - \underline{y})^2 / (2^2 - 1)$$

- Toplam sapma olan pay veya Toplam Kareler Toplamı (SST)

$$SST = \Sigma(y_i - \underline{y})^2$$

Sapmayı Tayin Etme(2 / 3)

- 2^2 tasarımı için, sapma 3 bölümden oluşur:

- $SST = 2^2q_a^2 + 2^2q_b^2 + 2^2q_{ab}^2$

(Türev 17.1, s.287)

- Toplam sapmanın kısımları:

- a için $2^2q_a^2$
 - b için $2^2q_b^2$
 - ab için $2^2q_{ab}^2$

- Böylece, $SST = SSA + SSB + SSAB$

- Ve kesir *Varyasyon* tarafından açıklanabilir:

$$= SSA/SST$$

- Dikkat ediniz, *sapma* hatalara bağlı olduğundan aynı kesir değil

Sapmayı Tayin Etme(3 / 3)

- Bellek önbellek çalışmasında

$$\underline{y} = \frac{1}{4} (15 + 55 + 25 + 75) = 40$$

- Toplam sapma

$$\begin{aligned} &= \Sigma(y_i - \underline{y})^2 = (25^2 + 15^2 + 15^2 + 35^2) \\ &= 2100 = 4 \times 20^2 + 4 \times 10^2 + 4 \times 5^2 \end{aligned}$$

- Böylece, toplam sapma 2100 olur
 - 1600 (2100'ün % 76'sı) belleğe atfedilir
 - 400 (2100'ün % 19'u) önbelleğe atfedilir
 - Sadece 100 (2100, 5%) etkileşime atfedilir
- Bu veriler, belleği daha fazla araştırmayı ve önbellekte (veya etkileşime) daha fazla zaman harcamamayı gerektirir

Genel 2^k Faktöriyel Tasarımlar (1/4)

- Aynı metodoloji her biri 2 seviyeli k faktörlerine genişletebilir $\rightarrow 2^2$ deneye ihtiyaç duyulur
 - k ana etkileri
 - ($k \rightarrow 2$) iki faktör etkiler
 - ($k \rightarrow 3$) üç faktör etkiler
- İşaret tablosu yöntemini kullanabilirsiniz

Genel 2^k Faktöriyel Tasarımlar (2/4)

- Örnek: LISP makinesi tasarımı

- Önbellek, bellek ve işlemciler

<u>Faktör</u>	<u>-1 düzey</u>	<u>1 düzey</u>
Bellek (a)	4 Mbyte	16 Mbyte
Önbellek (b)	1 Kbyte	2 Kbyte
İşlemciler (c)	1	2

- 2.³ tasarımı ve MIPS perf sonuçları şunlardır:

<u>4 Mbytes Mem(a)</u>			<u>16 Mbytes Mem</u>		
<u>Önbellek (b)</u>	<u>Bir proc (c)</u>		<u>İki proc</u>	<u>Bir proc</u>	<u>İki procs</u>
1 KB		14	46	22	58
2 KB		10	50	34	86

Genel 2^k Faktöriyel Tasarımlar (3 / 4)

- İşaret tablosunu hazırla:

i	a	b	c	ab	ac	bc	abc	y
1	-1	-1	-1	1	1	1	-1	14
1	1	-1	-1	-1	-1	1	1	22
1	-1	1	-1	1	-1	-1	-1	10
1	1	1	-1	1	-1	-1	-1	34
1	-1	1	1	-1	-1	1	-1	46
1	1	-1	1	-1	1	-1	-1	58
1	-1	1	1	-1	-1	1	-1	50
1	1	1	1	1	1	1	1	86
320	80	40	160	40	16	24	9	Ttl
40	10	5	20	5	2	3	1	Ttl/8

$$q_a=10, q_b=5, q_c=20 \text{ and } q_{ab}=5, q_{ac}=2, q_{bc}=3 \text{ ve } q_{abc}=1$$

Genel 2^k Faktöriyel Tasarımlar (3 / 4)

- $q_a=10, q_b=5, q_c=20$ ve $q_{ab}=5, q_{ac}=2, q_{bc}=3$ ve $q_{abc}=1$
- $$\begin{aligned} SST &= 2^3 (q_a^2 + q_b^2 + q_c^2 + q_{ab}^2 + q_{ac}^2 + q_{bc}^2 + q_{abc}^2) \\ &= 8 (10^2 + 5^2 + 20^2 + 5^2 + 2^2 + 3^2 + 1^2) \\ &= 800 + 200 + 3200 + 200 + 32 + 72 + 8 \\ &= 4512 \end{aligned}$$
- 7 adet faktörün katkısı :

mem = $800/4512$ (18%)	cache = $200/4512$ (4%)
proc = $3200/4512$ (71%)	mem-cache = $200/4512$ (4%)
mem-proc = $32/4512$ (1%)	cache-proc = $72/4512$ (2%)
mem-proc-cache = $8/4512$ (0%)	

İçindekiler

- Giriş
- Terminoloji
- Genel Hatalar
- Basit Tasarımlar
- Tam Faktöryel Tasarımlar
 - 2^k Faktöriyel Tasarımlar
- 2^{kr} Faktöriyel Tasarımlar

2^k Faktöriyel Tasarımları

Hiçbir deney beni haklılığımı kanıtlayamaz; ama tek bir deney yanıldığımı kanıtlayabilir. **-Albert Einstein**

- 2^k faktöriyel tasarımları sadece bir kez tekrarlandığı için hata tahmin etmek mümkün değildir
- 2^k tasarımını r kez tekrarlanır
- Daha önce olduğu gibi, 2^{2r} adet model genişletilir
- İki düzeyde iki faktör ile deneysel hatalar izole edilmek istenir
 - 4 konfigürasyonu r kez tekrar et
- Hata terimini verir:
 - $y = q_0 + q_a x_a + q_b x_b + q_{ab} x_a x_b + e$
 - E nasıl ölçülür

2²r Faktöriyel Tasarım Hataları (1 / 2)

- Daha önceki önbellek örneği r=3

i	a	b	ab	y	ortalama y
1	-1	-1	1	(15, 18, 12)	15
1	1	-1	-1	(45, 48, 51)	48
1	-1	1	-1	(25, 28, 19)	24
1	1	1	1	(75, 75, 81)	77
164	86	38	20		Toplam
41	21.5	9.5	5		Ttl/4

- Herbir y nin hesabı

- $y_i = q_0 + q_a x_{ai} + q_b x_{bi} + q_{ab} x_{ai} x_{bi} + e_i$

- Her tekrarın farkı (hata)

- $e_{ij} = y_{ij} - y_i = y_{ij} - q_0 - q_a x_{ai} - q_b x_{bi} - q_{ab} x_{ai} x_{bi}$

2²r Faktöriyel Tasarım Hataları (2/2)

- Sapma ve güven aralığını hesaplamak için hataların karesinin toplamı (SSE) kullanılır

$$SSE = \sum \sum e_{ij}^2 \text{ for } i = 1 \text{ to } 4 \text{ and } j = 1 \text{ to } r$$

- Örnek

i	a	b	ab	y _i	y _{i1}	y _{i2}	y _{i3}	e _{i1}	e _{i2}	e _{i3}
1	-1	-1	1	15	15	18	12	0	3	-3
1	1	-1	-1	48	45	48	51	-3	0	3
1	-1	1	-1	24	25	28	19	1	4	-5
1	1	1	1	77	75	75	81	-2	-2	4

- Ex: $y_1 = q_0 - q_a - q_b + q_{ab} = 41 - 21.5 - 9.5 + 5 = 15$
- Ex: $e_{11} = y_{11} - y_1 = 15 - 15 = 0$
- $SSE = 0^2 + 3^2 + (-3)^2 + (-3)^2 + 0^2 + 3^2 + 1^2 + 4^2 + (-5)^2$
 $+ (-2)^2 + (-2)^2 + 4^2$
 $= 102$

2²r Faktöriyel Sapma Tayini

- Toplam sapma (SST)

$$SST = \sum (y_{ij} - \bar{y}_{..})^2$$

- 4 bölüme ayrılabilir:

$$\sum (y_{ij} - \bar{y}_{..})^2 = 2^2 r q_a^2 + 2^2 r q_b^2 + 2^2 r q_{ab}^2 + \sum e_{ij}^2$$
$$SST = SSA + SSB + SSAB + SSE$$

- Böylece

- SSA, SSB, SSAB a, b ve ab faktörlerindeki sapma
- SSE deneysel hatalar nedeniyle açıklanamayan bir sapmadır

- Ayrıca $SST = SSY - SS0$ yazılabilir

- Burada, SS0 ortalamanın toplam kareleridir

2²r Faktöriyel Sapma Tayini Örneği

- Bellek ön belleği çalışması için:
 - $SSY = 15^2 + 18^2 + 12^2 + \dots + 75^2 + 81^2 = 27,204$
 - $SS0 = 2^2 r q^2_0 = 12 \times 41^2 = 20,172$
 - $SSA = 2^2 r q^2_a = 12 \times (21.5)^2 = 5547$
 - $SSB = 2^2 r q^2_b = 12 \times (9.5)^2 = 1083$
 - $SSAB = 2^2 r q^2_{ab} = 12 \times 5^2 = 300$
 - $SSE = 27,204 - 2^2 \times 3(41^2 + 21.5^2 + 9.5^2 + 5^2) = 102$
 - $SST = 5547 + 1083 + 300 + 102 = 7032$
- Böylece, 7032 toplam sapma 4 bölüme ayrılmıştır:
 - Faktör a 5547/7032 (%78,88), b %15,40, ab %4,27
 - Kalan % 1.45 açıklanamayan

Etkiler için Güven Aralıkları

- Hataların normal dağıldığını varsayarsak, y_{ij} normal olarak aynı sapma ile dağıtılır
- q_o, q_a, q_b, q_{ab} y_{ij} lerin tüm doğrusal kombinasyonu olduğu için (2^2r ile bölünür) aynı sapma değerine sahiptir
- Sapma $s^2 = SSE / (2^2(r-1))$
- Etkileri için güven aralıkları :
 - $q_i \pm t_{[1-\alpha/2; 2^2(r-1)]} s_{qi}$
- Güven aralığı sıfır içermiyorsa, etki önemli

Etkiler için Güven Aralıkları (Örnek)

- Bellek önbellek çalışması, hataların standart sapması

$$s_e = \sqrt{SSE / (2^2(r-1))} = \sqrt{102/8} = 3.57$$

- Etkilerin standart sapması:

$$s_{q_i} = s_e / \sqrt{2^2 r} = 3.57/3.47 = 1.03$$

- 8 serbestlik derecesindeki ve %95 güvendeki t değeri 1,86
- Parametreler için güven aralıkları:

$$q_i \pm (1.86)(1.03) = q_i \pm 1.92$$

- $q_0 \rightarrow (39.08, 42.91)$, $q_a \rightarrow (19.58, 23.41)$,
 $q_b \rightarrow (7.58, 11.41)$, $q_{ab} \rightarrow (3.08, 6.91)$

- Hiçbiri sıfır içermediğinden, tüm bunlar istatistiksel olarak anlamlıdır

Tahmini Yanıtlar için Güven Aralıkları (1/2)

- Ortalama tahmini yanıt
 - $y = q_0 + q_a x_a + q_b x_b + q_{ab} x_a x_b$
- m ilave deneyden gelen tahmini ortalama aynı olursa ancak tahmini yanıt üzerindeki güven aralığı azalır
- Bu şunu gösterir: tahmini y değerinin standart sapması daha fazla deneyden hesaplanmıştır
 - $s_{ym} = s_e \sqrt{1/n_{\text{eff}} + 1/m}$
 - Burada $n_{\text{eff}} = \text{runs}/(1+df)$
 - 2 seviye durumunda, her parametre 1 df, bu nedenle $n_{\text{Elver}} = 2^2 r/5$

Tahmini Yanıtlar için Güven Aralıkları (2/2)

- $100(1-\alpha)\%$ yanıt güven aralığı:
 - $y_p \pm t_{[1-\alpha/2; 2(r-1)]} s_{ym}$
- İki durum ilgi çekicidir
 - Bir adet çalışmanın standart sapması ($m=1$)
 - $s_{y1} = s_e \sqrt{5/2^2 r + 1}$
 - Birçok çalışmanın standart sapması ($m=\infty$)
 - $s_{y1} = s_e \sqrt{5/2^2 r}$

Tahmini Yanıtlar İçin Güven Aralıkları Örneği (1 /2)

- Mem-cache çalışması, için $x_a=-1$, $x_b=-1$
- Gelecekteki deney için tahmini ortalama yanıt
 - $y_1 = q_0 - q_a - q_b + q_{ab} = 41 - 21.5 + 1 = 15$
 - Std dev = $3.57 \times \sqrt{5/12 + 1} = 4.25$
- $t[0.95;8] = 1.86$, 90% conf interval kullanarak
$$15 \pm 1.86 \times 4.25 = (8.09, 22.91)$$
- Gelecekteki 5 deney için tahmini ortalama yanıt
 - Std dev = $3.57(\sqrt{5/12 + 1/5}) = 2.80$
$$15 \pm 1.86 \times 2.80 = (9.79, 20.29)$$

- Çok Sayıda Deney için Öngörülen Ortalama Yanıt

- Std dev = $3.57 \times \sqrt{5/12} = 2,30$

- Güven aralığı:

$$15 \pm 1.86 \times 2.30 = (10.72, 19.28)$$

BİLGİSAYAR BİLİMİNDE DENEYLER

Giriş

- Bazıları bilgisayar biliminin deneysel bir bilim olmadığını iddia eder
 - Bilgisayarlar insan yapımı, öngörülebilir
 - Teorik bir bilimdir (Matematik gibi)
- Bazı iddialar
 - sistem geliştirme = bilgisayar bilimi
 - Bir işletim sistemi veya birleşik veritabanı oluşturma
 - Bilgisayar mühendisliği
 - «bilim» ifadesi daha sonra gelir

Teori ve Mühendislik

- Şu ana kadar aldığınız bilgisayar teorisi:
"Yukarıdaki koddaki hatalara dikkat edin; sadece doğru olduğunu kanıtladım, denemedim."
 - - Donald E. Knuth- Amerikan bilişimci, bilim tarihçisi
- Şu ana kadarki mühendislik seviyemiz
 - Bir aparat geliştirmek faydalı olsa da, yeni bilgi vermedikçe boşa bir çabadır
 - Bilgiyi artırmak için bilime ihtiyaç var
- Teori veya aparatı değerlendirmek için deneyler kullanılır!

Bilgisayar Biliminde Deneyler

- "*Bilimin temel prensibi, neredeyse tanımı şudur: herhangi bir fikrin geçerliliğinin yegane yolu deneydir*"
 - Richard P. Feynman
- Fizik, Biyoloji, Kimya'dan denenmiş ve gerçek deneysel bilimsel metodoloji ...
 - Bilgisayar Bilimlerinde takip edilmez
- Daha iyi Bilgisayar *Uzmanları* olalım !

Bilimsel Metodoloji

- Gözlem
 - (Çözüm geliştirin)
- Hipotez / Varsayımda bulunmak
- Tasarım
- Deney
- Analiz
- Rapor

Metodoloji: Gözlemleyin ve Anlayın

- Sorun Bul
 - Test: *Netscape Ses*
 - Derle: *Ses konferansı*
 - Oku: *Kevin Jeffay diyor ki...*
- İlişkileri Anlayın
 - *UDP paketleri kaybediyor*
 - *TCP gecikmeyi artırır*
 - *P-çerçeveleri I-çerçevelerine bağlıdır*

Metodoloji: Tasarla ve hipotez

- Çözüm Tasarlayın (deneysel değilse)
 - *Claypool Güvenilir Ses Protokolü (CRAP)*
 - *Claypool tamponlama algoritması*
- Hipotez Yap
 - İlişkiler hakkında genelleme
 - *İşlemci yükü ısınmayı artırır*
 - *Java sanal makinesi ısınmayı artırır*
 - Test edilmesi gerekenler (kanıtlanmamış)

Metodoloji: Deney

- Tasarım Deneyi
 - Değişken: *işlemci iş yükü*
 - Kontrol: *temel iş yükü*
- Deney yap
 - “Eyvah! Beklediğim bu değil! ”
 - Koddaki hata (*soketi paylaşan iki işlem*)
 - “Çalıştır” a geri dön
 - Kontrolsüz olay (*sistem yedeklemesi*)
 - Tasarım'a geri dön
 - Yetersiz anlayış (*Unix sıralaması*)
 - «Anlayışa” ya geri dön

Metodoloji: Analiz et

- Yorumlama ve Değerlendirme
 - İstatistiksel anlamlılık
 - ortalama, güven aralıkları, korelasyon,
 - Uygunluğun güzelliği
 - Veriler hipotezi *destekliyor veya reddediyor mu?*
 - Diğer fenomenlerin açıklaması
 - *İşlemci yükü telefon aramamı zorlaştırıyor*
 - *Etkileşimli multimedya için Java yetersiz*

Çukurlar

- Mini deneyler (hayır, “Pilot Testleri”)
- Aslında hipotezler
 - Kodun çalışması anlamayı sağlar
- Kontrollü sistem hala gerçek dünya hakkında anlamlı şeyler söylüyor
- Bir sistemi gözlemlemek sistemi değiştirmez