EGE UNIVERSITY

TERM PROJECT

Distributed Least Squares Identification

BIL 556: Distributed Estimation and Control

MURAT MULAYIM
91170000035

Lecturer: Assistant Professor K. Sinan YILDIRIM

June 20th, 2019

1. Giriş

Merkezi tahminleme (centralised estimation) yöntemi ile düğümlerden elde edilen verilerin merkezi bir bilgisayar ve/veya düğümde toplanarak, benimsenen tahminleme algoritmasının bu merkezde işlenmesi ve istenilen tahminlemenin yapılmasıdır.

Dağıtık tahminleme (distributed estimation) yöntemi ise; merkezi tahminleme yönteminin aksine düğümlerdeki veri toplanmaz, veri düğümlerde işlenir. Düğümler komşuları ile veri aktarımı yaparak daha güvenilir sonuçlar elde edilmeye çalışılır.

Merkezi tahminleme algoritmaları, her ne kadar daha doğru, daha güvenli sonuçlar verse de dağıtık tahminlemeye göre dezavantajları bulunmaktadır. Bunlar;

- Düğümlerce toplanan veriler merkezi bir ortama iletilmelidir. Bu iletim doğrudan veya dolaylı olabilir.
- Merkeze iletilmek istenen verinin merkeze nasıl iletileceği; öncesinde berlirlenen yönlendirme algoritmaları ile yapılmalıdır.
- Merkezde toplanan veriler ile elde edilen sonucun gerekli düğümler ile paylaşılması gerekebilir.

Yukarıdaki nedenler dolayısı ile her düğüm; mesajlaşma için ekstra enerji harcamalıdır. Bu nedenle merkezi olarak çalışan en küçük kareler metodunun dağıtık çalışan sistemler üzerinde çalışabilecek duruma evrilmesi ele alınacak olup, örnek bir graf üzerinde değişkenlerin evrimi üzerinde durulacaktır.

2. Least Squares Algoritması

Tahminleme işleminin merkezde yapılmasının aksine dağıtık hesaplanması amacıyla en küçük kareler(least squares) yöntemi geliştirilmiştir. Bu yöntem, birbirine bağlı olarak değişen iki fiziksel büyüklük arasındaki ilişkiyi mümkün olduğunca gerçeğe uygun bir denklem olarak yazmak için kullanılır. Bu sayede elde edilen denklem kullanılarak, doğruluğundan emin olunan değişken ile diğer değişken hesaplanabilir.

$$A.\theta = b \qquad (1)$$

$$\theta = (A^T A)^{-1} A^T b \qquad (2)$$

En küçük kareler algoritmasının merkezi tahminleme yöntemine uygulanması için gerekli formül yukarıda belirtilmiştir. 1 numaralı eşitlik tanımda da belirtildiği üzere birbirine bağlı iki değişken arasındaki ilişkiyi belirtmektedir. θ değeri iki değişken arasındaki güven katsayısını belirtmektedir. A ve b matrisleri her a_i ve b_i değerlerinden oluşan matrislerdir.

$$A = \begin{bmatrix} a_1 \\ a_2 \\ . \\ . \\ a_n \end{bmatrix} \quad , \quad b = \begin{bmatrix} b_1 \\ b_2 \\ . \\ . \\ b_n \end{bmatrix}$$

En küçük kareler algoritmasının dağıtık çalışan sistemler ile uyumlu çalışabilmesi amacıyla formüller üzerinde düzenlemelere ihtiyaç duymaktayız. Çünkü, bu sistemlerde verileri toplayan merkezi bir ortam bulunmadığı için, her düğüm her düğümün verisine sahip değildir. Her düğüm yalnızca kendi ve komşularının verilerine sahiptir.

Her düğüm ne kadar süre algoritmayı çalıştıracağını bilmediği için algoritma sonsuza kadar çalışacakmış gibi düşünülür ve formüller bu yolda evrilir.

$$\theta = \left(\frac{1}{N}\sum_{i=1}^{N} a_i a_i\right)^{-1} \left(\frac{1}{N}\sum_{i=1}^{N} a_i b_i\right)$$

$$b = a_i\,\theta(0) + \theta(1)$$

Formülde kullanılan a_i değerleri her düğümün kendi ve kendine iletilen (komşularından gelen) a değerlerini temsil ederken b_i değerleri de b değerlerini temsil eder. N sayısı komşularından gelen (a,b) veri çifti sayısıyla ile orantılıdır. Bu nedenle N sayısı her döngüde değişkenlik gösterebilir.

3. Nasıl Çalışır ?

En küçük kareler algoritmasının dağıtık sistemler üzerinde uygulanması amacıyla 3 aşamalı çalışacak şekilde gerçeklendirilir. Her düğüm θ ve b değerlerini hesaplapladığı **hesaplama**, hesaplanan değerlerin komşularla paylaşıldığı mesaj **gönderme** ve komşulardan gelen mesajların toplandığı **mesaj alma** aşamalarına sahip en küçük kareler algortimasını yürütür.

3.1. Hesapla

Her düğüm başlangıçta ilk olarak, yalnızca kendi (a, b) very ikilisini kullanarak aşağıdaki formül aracılığıyla θ değerini hesaplar ve güncel b değeri bulunur.

$$\theta = \left(\frac{1}{N}\sum_{i=1}^{N} a_i a_i\right)^{-1} \left(\frac{1}{N}\sum_{i=1}^{N} a_i b_i\right) = \frac{\sum_{i=1}^{N} a_i b_i}{\sum_{i=1}^{N} a_i a_i}$$

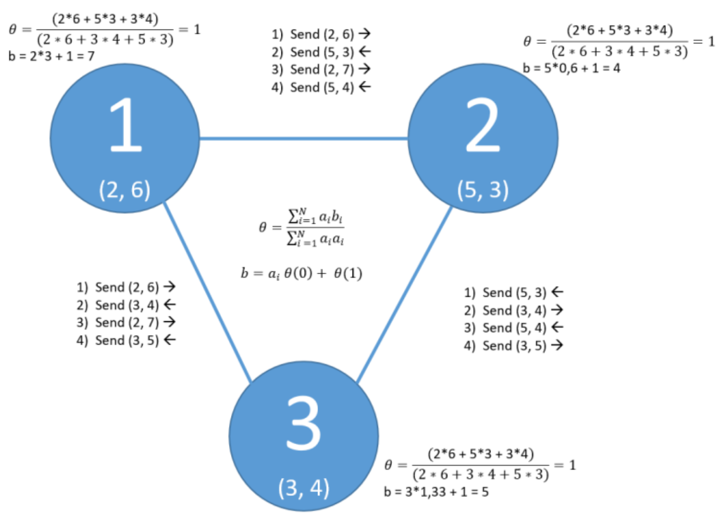
Komşulardan alınan mesajlar aracılığı ile düğümlerde toplanan (a_i, b_i) ikili verileri artacağından toplanan veri sayısı kadar N değeri de artacaktır.

3.2. Mesaj Gönder

Hesaplanan θ değeri kullanılarak elde edilen, o düğüme ait yeni b değerinin komşular ile paylaşıldığı aşamadır. Komşulara b değeri ile birlite a değeri de (a, b) veri ikilisi olarak paylaşılır.

3.3. Mesaj Al

Her bir komşudan hesaplanan son (a_i, b_i) verisi alınır. Fakat daha önce o komşuya ilişkin (a_i, b_i) değerleri alındı ise, eski değer saklanmamalı, yenisi ile güncellenmelidir.



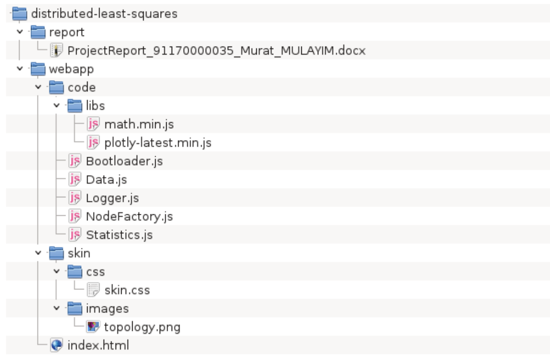
Şekil-1

4. Gerçeklendirilmesi

En küçük kareler algoritması Javascript programlama dili ile gerçeklendirilmiş olup, HTML ve CSS ile arayüz düzenlemesi yapılmıştır. Algoritmayı çalıştırmak için proje dizini içerisinde yer alan **index.html** herhangi bir tarayıcı ile açmak gerekmektedir.

Proje dosya yapısı Şekil-2'de görülmektedir. Dosya yapısının içeriği olarak:

- /distributed-least-squares/webapp/code/* : Projeye ilişkin bütün Javascript dosyaları(.js uzantılı) dizin altında bulunmaktadır.
- /distributed-least-squares/webapp/code/libs/* : Projeye dahil olmayan fakat matematiksel ve istatistiksel işlemler için kullanılan kütüphaneleri barındırır.
- /distributed-least-squares/webapp/**index.html** : Uygulamayı başlatmak için açılması gereken dosyadır.
- /distributed-least-squares/webapp/skin/* : Arayüz düzenlemesi için kullanılan CSS ve resimlerin bulunduğu dizindir.



Şekil-2

Algoritmanın gerçeklendirilmesi kısmı NodeFactory.js dosyası içerisinde bulunmaktadır. Dosya içerisinde bulunan **Node** sınıfı graftaki her bir düğümü temsil eder ve 3 temel fonksiyonu bulunmaktadır.

- **iterate()** : Toplanan (a_i, b_i) değerleri kullanılarak en küçük kareler algoritması çalışıtırılır.
- **sendMessage()** : Elde edilen değerler (a, b) komşulara gönderilir.
- **receiveMessage()** : Herhangi bir komşudan bir mesaj(a_i, b_i) alındığında çalışır ve iterate() fonksiyonu tetiklenir.

```
Node.prototype.iterate = function () {
	setTimeout( function (node) {
		if(!node.converged){
			Logger.log(module.id, "Node-" + (node.id+1) + ": iterate()");
			node.teta_zero = node.teta_last;

			// Formula is: ((1/N*(all a_i*a_i))^(-1)*)(1/N*(all a_i*b_i))
			//                                        : (all a_i*b_i)/(all a_i*a_i)
			var upperResult = node.a*node.b; // a*b
			var lowerResult = node.a*node.a; // a*a
			for(var nodeId in node.received_A_s) {
				upperResult += node.received_A_s[nodeId]*node.received_b_s[nodeId];
				lowerResult += node.received_A_s[nodeId]*node.received_A_s[nodeId];
			}

			node.teta_last = (upperResult / lowerResult);
			node.b = node.a*node.teta_zero + node.teta_last;


			var msg = {
				a: node.a,
				b: node.b
			};
			if(_isConverge(node.teta_last)){
				node.converged = true;
			}
			node.sendMessage(msg);
		}
	}, 100, this);
}
```

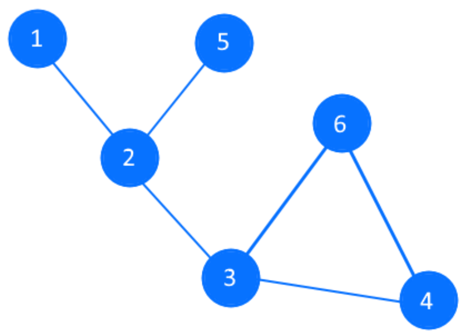
```
Node.prototype.receiveMessage = function (fromNodeId, msg) {
	if(this.received_A_s.fromNodeId == undefined) {
		this.N++;
	}

	this.received_A_s[fromNodeId] = msg.a;
	this.received_b_s[fromNodeId] = msg.b;

	this.iterate();
}
```

```
Node.prototype.sendMessage = function (msg) {
	for(var i = 0; i < this.neighbors.length; i++) {
		module.nodeMap[this.neighbors[i]].receiveMessage(this.id, msg);
	}
}
```

Proje içeriğinde kullanılan topoloji 6 adet düğümden oluşup grafiksel olarak aşağıdaki şekilde gibidir.



Şekil-3

Düğümler üzerinde ölçülen (a, b) ikili verileri ise

$$\begin{pmatrix} 2 & 3 \\ 8 & 8 \\ 4 & 3 \\ 1 & 5 \\ 8 & 5 \\ 6 & 3 \end{pmatrix}$$

şeklinde ifade edilebilir. İlk satır 1 nolu düğüme ait (a, b) verisini, son satır ise 6 numaralı düğüme ait (a, b) verisini temsil etmektedir.

5. Yakınsama Adımları

Kullanılan grafta yer alan her düğümdeki ölçümler ve bu düğümlerin üzerinde çalışan en küçük kareler metodu yardımıyla, bu farklı değerlerin birbirine yakınsaması aşağıdaki şekildedir. Ölçülen a_i değerlerinin doğruluğundan emin olunduğu farzedilmiş, bu değişkenler aracılığı ile b_i değerleri hesaplanmak istenmektedir. Bu amaçla hesaplanan θ katsayısı her düğüm üzerinde bir değere yakınsamaktadır.

En küçük kareler metodunun merkezi yöntem ile çalıştırılması sonucu;

$$\theta = (A^T * A)^{-1} * A^T * b$$

$$\theta = \left(\begin{bmatrix} 2 & 8 & 4 & 1 & 8 & 6 \end{bmatrix} * \begin{bmatrix} 2 \\ 8 \\ 4 \\ 1 \\ 8 \\ 6 \end{bmatrix} \right)^{-1} * \begin{bmatrix} 2 & 8 & 4 & 1 & 8 & 6 \end{bmatrix} * \begin{bmatrix} 3 \\ 8 \\ 3 \\ 5 \\ 5 \\ 3 \end{bmatrix}$$

$$\theta = 0.7837837837837839$$

sonucu elde edilmektedir. Elde edilen θ değeri kullanılarak;

(a_1, b_1): (2, 1.5675675675675678)
(a_2, b_2): (8, 6.270270270270271)
(a_3, b_3): (4, 3.1351351351351355)
(a_4, b_4): (1, 0.7837837837837839)
(a_5, b_5): (8, 6.270270270270271)
(a_6, b_6): (6, 4.7027027027027035)

yakınsanan (a_i, b_i) değerleri hesaplanır. Dağıtık çalışan en küçük kareler metodunu aynı değişkenler üzerine uyguladığımızda aşağıdaki tablo ile karşılaşıyoruz. Adımlara ait θ_i ve $(a_i\ , b_i)$ sütunlarındaki değerler 6 adettir ve bu değerler sıralı olarak düğümlerdeki değerleri temsil eder.

Adım	θ_i	$(a_i\ , b_i)$
1	1.5 0.9852941176470589 0.2485294117647059 0.3525951557093426 0.3740808823529412 0.3650323170333616	(2, 1.5) (8, 0.9852941176470589) (4, 0.2485294117647059) (1, 0.3525951557093426) (8, 0.3740808823529412) (6, 0.3650323170333616)
2	0.9832729355653232 0.10046701112877583 0.5760638147300027 0.1664814133031097З 0.5223063021661368	(2, 3.9832729355653234) (8, 7.982819952305247) (4, 1.5701814617888261) (1, 0.5190765690124524) (8, 3.5149533609896664)

	0.16962257204467784	(6, 2.3598164742448473)
3	0.8392882782850216 0.7177666096259816 0.283168571375097 0.3084061652706569 0.6033178432825637 0.4722020255965974	(2, 2.805834149415668) (8, 1.5215026986561884) (4, 2.5874238302951076) (1, 0.4748875785737666) (8, 4.781768260611658) (6, 1.489955634827727)
4	0.7165900988696324 0.3959992545234994 3 0.5890048098411117 0.3976380076841452 0.6357078477281011 0.4019994528611458	(2, 2.3951666554396756) (8, 6.138132131531353) (4, 1.7216790953414995) (1, 0.7060441729548022) (8, 5.46225059398861) (6, 3.2353206682191042)
5	0.8240032690525689 0.7069666517276688 0.559029015565444 0.5977909057219746 0.7417179198172205 0.6210407602054219	(2, 2.257183466791834) (8, 3.874960687915664) (4, 2.915048254929891) (1, 0.9954289134061198) (8, 5.827380701642029) (6, 3.0330374773722966)
6	0.8221851024196085 0.5846330601808749 0.6401249540708774 0.6032509951644522 0.7657286378444215 0.6071304854942694	(2, 2.4701916405247464) (8, 6.240366274002225) (4, 2.8762410163326537) (1, 1.2010419008864268) (8, 6.699471996382185) (6, 4.3333750467268)
7	0.8284500487058706 0.7124928215909593 0.638798210870972 0.765122596170902 0.8202343437656812 0.7713935488720711	(2, 2.4728202535450876) (8, 5.389557303037958) (4, 3.199298027154482) (1, 1.368373591335354) (8, 6.946063446521053) (6, 4.4141764618376875)
8	0.8285273608535277 0.7052935503592431 0.6394693980351297 0.7682797978774856 0.8356463093993605 0.7805408788846243	(2, 2.485427458265269) (8, 6.405236123086917) (4, 3.194662241519018) (1, 1.5334023940483874) (8, 7.397521059524809) (6, 5.408902172117051)
SON DURUM	0.8289136146521843 0.7819291009947689 0.7796737618121251 0.8981101652225605 0.8733318993297308 0.7805408788846243	(2, 2.4867099366368977) (8, 6.424277503868714) (4, 3.337551353952644) (1, 1.7962203288281997) (8, 7.784231181026493) (6, 5.408902172117051)

Her adım sonucunda θ değeri, merkezi çalışan en küçük kareler metodunun elde ettiği θ değerine yaklaşmaktadır. Çünkü, bir düğüm herhangi bir komşusundan mesaj aldığında; tek değil birden fazla (a, b) değeri ile işlem yapmaktadır ve bu verilerin ortalamasına yaklaşmaktadır. Buradaki örnekte ise bu değer merkezi çözümde elde edilen $\boldsymbol{\theta = 0.783}$ ... değerine yakınsamaktadır. Metodun sonuna yaklaşıldığında, her düğüm her komşusunun verisine sahip olacak duruma gelir. Bu sayede grafta bulunan bütün düğümlere ilişkin verileri kullanan merkezi çalışan en küçük kareler metodunun sonucuna yaklaşmaktadır.

6. Neden Yakınsar

En küçük kareler metodu ile her düğümde ölçülen değer arasındaki katsayı θ değeri bir değere yakınsamaktadır. Çünkü iki (a, b) değişkenleri arasında aşağıdaki eşitliği kullanarak lineer bağlantı kurmaya çalışıyoruz.

$$b = a * \theta_0 + \theta_1$$

Burada kullanılan θ değerini elde etmek için ise döngüsel olarak aşağıdaki formülü uygulamaktayız.

$$\theta = \left(\frac{1}{N}\sum_{i=1}^{N} a_i a_i\right)^{-1} \left(\frac{1}{N}\sum_{i=1}^{N} a_i b_i\right) = \frac{\sum_{i=1}^{N} a_i b_i}{\sum_{i=1}^{N} a_i a_i}$$

Her düğüm öncelikle kendi (a, b) veri ikilisine sahip olacağından ilk aşamada;

$$\theta = \frac{ab}{aa}$$

Değerini elde edecektir ve bu θ değeri a ve b değerleri arasındaki lineer oranı verecektir. Herhangi bir komşusundan mesaj alan düğüm; ikinci aşamada ise bu iki düğümün değerleri arasındaki lineer oranların ortalamasını hesaplayacaktır.

$$\theta = \frac{a_1 b_1 + a_2 b_2}{a_1 a_1 + a_2 a_2}$$

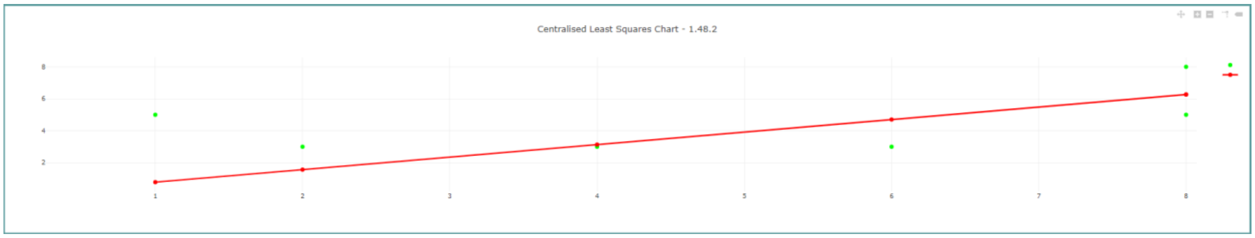
3.adımda ise bir diğer komşudan alınan mesajın da dahil olduğu 3 adet (a, b) ikili verisi kullanılarak θ değerlerinin ortalaması alınır.

$$\theta = \frac{a_1 b_1 + a_2 b_2 + a_3 b_3}{a_1 a_1 + a_2 a_2 + a_3 a_3}$$

İterasyonu sonsuza götürdüğümüzde ise; her düğüm dolaylı olarak graftaki her düğümün değerlerine sahip olacaktır. Fakat sahip olduğu değerler, gerçek (a, b) değerleri değil, komşuları aracılığıla hesaplanan (a, b) değerlerine sahip olacaktır.

$$\theta = \frac{a_1 b_1 + \ a_2 b_2 + \cdots + \ a_N b_N}{a_1 a_1 + \ a_2 a_2 + \cdots + \ a_N a_N}$$

Unutulmamalıdır ki en küçük kareler metodu ile (a, b) değerleri belli bir değere yakınsamamaktadır. θ değeri yani (a, b) arasındaki lineer oran bir değere yakınsamaktadır. Bu sayede Şekil-4'te de görüldüğü üzere, systemin genelini kapsayan bir eşitlik aracılığıyla bilinen bir değer ile bilinmeyen bir değeri hesaplayabiliriz.



Şekil-4

7. Düğümlerin Hareketli Olduğu Durumda

En küçük kareler metodu merkezi ve dağıtık çalışacak şekilde entegre edilmiş olup, uygulaması ve çalışma şekli önceki kısımlarda anlatılmıştır. Fakat uygulama için kullanılan graftaki her düğümün sabit bir pozisyonda durduğu kabul edilmiştir. Her bir düğümün veya kısmi hareketli olduğu bir grafta en küçük kareler yöntemini uygulamak ayrı bir zorluk oluşturacaktır. Bu zorluklar,

- Düğümler hareketli olduğu için bir düğüme ait komşular her an değişebilir.
- Hareketli olan düğümler graftan kopabilir, hiçbir düğüm ile iletişim kuramayabilir.

yukarıdaki değişen koşullar nedeniyle, en küçük kareler metodunu uygulamak zor olabilir hatta imkansız durumlar ile de karşılaşılabilir.

Düğümlerin hareketli olduğu bir sistemde en küçük kareler metodunun sağlıklı çalışabilmesi, yani sistemin (θ değerinin) yakınsaması durumunun sağlaması aşağıdaki şartlarda gerçekleşir.

- Sistem bağlı graf şeklinde kalmalıdır.
- Her düğüm en az bir komşu ile iletişime geçebiliyor olmalıdır.

Bu koşullar sağlandığında her düğüm yine dolaylı olarak diğer düğümlerin verilerine sahip olacaklardır. Fakat buradaki yakınsama statik duran sisteme göre daha yavaş veya dalgalanmalı şeklinde olabilir.