# lineare Regression

## Autor und Herausgeber

Alexander Jung*, Department of Computer Science, Aalto University, Espoo, Finland (ORCID 0000-0001-7538-0990)
Editors: Konstantina Olioumtsevits (ORCID 0000-0003-2697-8659), Ekkehard Schnoor (ORCID 0000-0001-8252-1499), Aalto University, Espoo, Finland
*Corresponding author: alex.jung@aalto.fi

## Zusammenfassung

Die lineare Regression ist eine regression-Methode, die eine lineare Hypothese-Abbildung lernt, um das numerische Label eines Datenpunkt aus dessen Merkmale vorherzusagen. Ihre least squares-Variante wählt eine Abbildung, die den durchschnittlichen squared error loss auf einem Trainingsdatensatz minimiert. Die lineare Regression ist eine Instanz von empirische Risikominimierung (ERM), die durch die Verwendung des lineares Modell und des squared error loss erhalten wird. Die optimalen Modellparameter werden durch die Normalgleichungen bestimmt. Dieses System kann in geschlossener Form oder iterativ gelöst werden, zum Beispiel durch Gradientenabstieg. Das Hinzufügen eines Strafterm zum durchschnittlichen squared error loss führt zu regularisierten Varianten wie der Ridge-Regression und dem least absolute shrinkage and selection operator (Lasso).

## Definition

Die lineare Regression kann verwendet werden, um die morgige Temperatur aus Wetterdaten der letzten fünf Tage vorherzusagen, wobei diese als Merkmale und die morgige Temperatur als Label dienen. Methoden der linearen Regression verwenden ein lineares Modell: Sie lernen eine lineare Hypothese-Abbildung, die eine Vorhersage des numerischen Label eines Datenpunkt liefert. Die Vorhersage basiert ausschließlich auf den Merkmale des Datenpunkt, die als Eingabe in die gelernte Hypothese-Abbildung eingespeist werden.

Formal lernt die lineare Regression eine lineare Hypothese-Abbildung $h^{(\mathbf{w})}(\mathbf{x}) = \mathbf{w}^{\top}\mathbf{x}$, um das numerische Label $y \in \mathbb{R}$ eines Datenpunkt aus dessen Merkmalsvektor $\mathbf{x} = \big(x_1, \ldots, x_d\big)^{\top} \in \mathbb{R}^d$ vorherzusagen. Hier bezeichnet $d$ die Anzahl der Merkmale eines Datenpunkt. Die Modellparameter $\mathbf{w}$ werden aus einem Trainingsdatensatz $\mathcal{D}^{(\text{train})} = \big\{ \big(\mathbf{x}^{(r)}, y^{(r)}\big) \big\}_{r=1}^{m}$ von Datenpunkte gelernt. Das Anhängen eines konstanten Merkmal an $\mathbf{x}$ lässt einen Eintrag von $\mathbf{w}$ als Achsenabschnitt wirken, sodass $\mathbf{w}^{\top}\mathbf{x}$ eine affine Funktion der ursprünglichen Messungen darstellt. Im Folgenden bezeichnet $\mathbf{x}$ den jeweiligen Merkmalsvektor, der der linearen Hypothese-Abbildung zugeführt wird, ob ursprünglich oder erweitert, sofern nicht anders angegeben.

Die least squares-Variante der linearen Regression misst die Qualität einer linearen Hypothese-Abbildung anhand ihres durchschnittlichen Verlust, des squa-

red error loss, auf dem Trainingsdatensatz. Als Instanz von empirische Risikominimierung (ERM) lernt sie die Modellparameter $\mathbf{w}$ durch Lösen des Optimierungsproblem

$$\min_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{m} \sum_{r=1}^{m} \left( y^{(r)} - \mathbf{w}^\top \mathbf{x}^{(r)} \right)^2.$$

Abb. 1 veranschaulicht dieses lineare Regressionsproblem für ein Trainingsdatensatz mit einem konstanten skalaren Merkmal $x^{(r)} = 1$ für $r = 1, \ldots, m$.

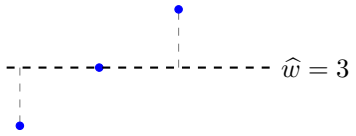


Abbildung 1: Für ein lineares Modell mit $d = 1$ und dem konstanten Merkmal $x = 1$ für jeden Datenpunkt (blaue gefüllte Kreise) reduziert sich die lineare Regression auf die Berechnung des Durchschnitts $\widehat{w} = (1/m) \sum_{r=1}^{m} y^{(r)}$ der Labels: dieser Durchschnitt minimiert den durchschnittlichen squared error loss. Die horizontale gestrichelte Linie liegt auf der Höhe $\widehat{w}$, und die vertikalen grauen Segmente sind die Fehler $y^{(r)} - \widehat{w}$, die in diesen Verlust eingehen.

Das Optimierungsproblem kann kompakter unter Verwendung der Merkmalsmatrix $\mathbf{X}$ und des Label-Vektor $\mathbf{y}$ geschrieben werden,

$$\mathbf{X} = \left( \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)} \right)^\top \in \mathbb{R}^{m \times d}, \qquad \mathbf{y} = \left( y^{(1)}, \ldots, y^{(m)} \right)^\top \in \mathbb{R}^m.$$

In Bezug auf $\mathbf{X}$ und $\mathbf{y}$ lautet das Optimierungsproblem

$$\min_{\mathbf{w} \in \mathbb{R}^d} \underbrace{\frac{1}{m} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2}_{f(\mathbf{w})}. \tag{1}$$

Der durchschnittliche squared error loss $f(\mathbf{w})$ ist eine konvex und differentiable Funktion von $\mathbf{w}$. Gemäß der zero-gradient condition (Boyd and Vandenberghe, 2004, Sect. 4.2.3) löst ein Vektor $\widehat{\mathbf{w}}$ das Optimierungsproblem (1) genau dann, wenn er das System der lineare Gleichungen (Strang, 2016) erfüllt

$$\mathbf{X}^\top \mathbf{X} \widehat{\mathbf{w}} = \mathbf{X}^\top \mathbf{y}. \tag{2}$$

Die Normalgleichungen (2) haben immer eine Lösung, da $\mathbf{X}^\top \mathbf{y}$ im column space von $\mathbf{X}^\top$ liegt, welcher mit dem column space von $\mathbf{X}^\top \mathbf{X}$ übereinstimmt (Meyer, 2000, Sect. 4.5).

Die Lösung der Normalgleichungen (2) ist jedoch nur dann eindeutig, wenn die Merkmalsmatrix $\mathbf{X}$ vollen Spalten-rank hat. Voller Spalten-rank erfordert mindestens so viele Datenpunkte wie Merkmale, $m \geq d$. Wenn dies zutrifft und die Spalten von $\mathbf{X}$ linearly independent sind, ist die Matrix $\mathbf{X}^\top \mathbf{X}$ invertible und

die Normalgleichungen (2) haben die eindeutige geschlossene Lösung (Golub and Van Loan, 2013, Sect. 5.3.1)

$$\widehat{\mathbf{w}} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{y}. \tag{3}$$

Wenn stattdessen $m < d$ (mehr Merkmale als Datenpunkte) gilt, kann $\mathbf{X}$ keinen vollen Spalten-rank haben, $\mathbf{X}^\top \mathbf{X}$ ist singulär, die geschlossene Form (3) ist nicht anwendbar, und das Optimierungsproblem hat unendlich viele Lösungen (Golub and Van Loan, 2013, Sect. 5.5). Alle diese Lösungen führen zum gleichen minimalen durchschnittlichen squared error loss auf dem Trainingsdatensatz, aber ihre Vorhersagen für Datenpunkte außerhalb des Trainingsdatensatz können beliebig voneinander abweichen. Die Wahl zwischen ihnen ist daher eine Frage der Generalisierung (siehe Überanpassung). Eine eindeutige Lösung kann ausgewählt werden, zum Beispiel durch die Wahl der Minimum-Norm-Lösung $\mathbf{X}^+ \mathbf{y}$, die durch die pseudoinverse $\mathbf{X}^+$ gegeben ist (Golub and Van Loan, 2013, Sect. 5.5.2).

Ein übliches Gegenmittel ist die Regularisierung: das Hinzufügen eines Strafterm zum durchschnittlichen squared error loss. Der Strafterm kann als Schätzung interpretiert werden, wie viel höher der Verlust auf Datenpunkte außerhalb des Trainingsdatensatz ist als auf diesem (Hastie et al., 2009, Sect. 7.4). Verschiedene Strafterme ergeben regularisierte Varianten der linearen Regression. Die Ridge-Regression verwendet den Strafterm $\alpha \|\mathbf{w}\|_2^2$ und das least absolute shrinkage and selection operator (Lasso) verwendet $\alpha \|\mathbf{w}\|_1$. Der Hyperparameter $\alpha > 0$ legt das Gewicht des Strafterm fest.

Die lineare Regression hat auch eine statistische Interpretation. Betrachten Sie ein Wahrscheinlichkeitsmodell mit einer gemeinsamen Wahrscheinlichkeitsverteilung $\mathbb{P}^{(\mathbf{x},y)}$ über den Merkmale und dem Label. Unter dem squared error loss ist der Bayes estimator (d.h. die Hypothese mit minimalem Risiko) der conditional expectation $\mathbb{E}\{y \mid \mathbf{x}\}$ (Lehmann and Casella, 1998, Ch. 4; Papoulis and Pillai, 2002). In dieser statistischen Interpretation bezeichnet $\mathbf{x}$ den ursprünglichen Merkmalsvektor. Seien die Merkmale und das Label gemeinsam Gaußsche Zufallsvariablen mit Null-Mittelwert und invertible Kovarianzmatrix $\mathbf{C}^{(\mathbf{x})} = \mathbb{E}\{\mathbf{x}\mathbf{x}^\top\}$. Dieser Bayes estimator ist dann (Bishop, 2006, Sect. 2.3.1, Gl. (2.81))

$$h^*(\mathbf{x}) = \left(\mathbf{w}^\star\right)^\top \mathbf{x}, \text{ mit } \mathbf{w}^\star = \left(\mathbf{C}^{(\mathbf{x})}\right)^{-1} \mathbf{c}^{(\mathbf{x},y)}.$$

Hier ist $\mathbf{c}^{(\mathbf{x},y)} = \mathbb{E}\{\mathbf{x}y\}$ die Kovarianz zwischen den Merkmale $\mathbf{x}$ und dem Label $y$. Dieser Bayes estimator ist linear in $\mathbf{x}$. Ohne die Annahme des Null-Mittelwerts ist es eine affine Funktion von $\mathbf{x}$, die ein konstantes Merkmal wie zuvor absorbiert.

Die Kleinste-Quadrate-lineare Regression ist das ERM-Gegenstück dieses predictor. Sie ersetzt $\mathbf{C}^{(\mathbf{x})}$ und $\mathbf{c}^{(\mathbf{x},y)}$ durch die stichprobenbasierten Schätzungen $(1/m)\mathbf{X}^\top \mathbf{X}$ und $(1/m)\mathbf{X}^\top \mathbf{y}$. Für invertible $\mathbf{X}^\top \mathbf{X}$ stellt dies die eindeutige Lösung $\widehat{\mathbf{w}} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{y}$ der Normalgleichungen (2) wieder her. Anstatt die Normalgleichungen (2) direkt zu lösen (mittels der Inverse oder der pseudoinverse), lösen machine learning (ML)-Methoden sie oft durch eine iterative Optimierungsverfahren, wie sie in weit verbreiteten ML-Softwarebibliotheken

wie scikit-learn (Pedregosa et al., 2011) und PyTorch (Paszke et al., 2019) implementiert ist. Ausgehend von initialen Modellparameter $\mathbf{w}^{(0)}$ wendet eine solche Methode wiederholt einen Update-Operator $\mathcal{F}$ an,

$$\mathbf{w}^{(t+1)} = \mathcal{F}\big(\mathbf{w}^{(t)}\big) \text{ für } t = 0, 1, \ldots.$$

Der Operator $\mathcal{F}$ ist so konzipiert, dass die Folge der Modellparameter $\mathbf{w}^{(t)}$ zu einer Lösung der Normalgleichungen (2) konvergiert.

Eine solche iterative Optimierungsverfahren ist die Gradientenabstieg, die den Gradientenabstieg-Schritt-Operator verwendet

$$\mathcal{F}^{(\eta)}(\mathbf{w}) = \mathbf{w} - \eta \nabla f(\mathbf{w}).$$

Hier ist $\eta > 0$ eine Schrittweite, die klein genug gewählt werden muss, um die convergence zu gewährleisten (Bertsekas, 2016, Prop. 1.2.2). Der hochgestellte Index in $\mathcal{F}^{(\eta)}$ macht die Abhängigkeit des Operators von $\eta$ explizit. Das Einsetzen des durchschnittlichen Trainingsfehlers in die allgemeine Form des Gradientenabstieg-Schritts ergibt das explizite Update

$$\begin{aligned}\mathcal{F}^{(\eta)}(\mathbf{w}) &= \mathbf{w} + \frac{2\eta}{m} \sum_{r=1}^{m} \big(y^{(r)} - \mathbf{w}^\top \mathbf{x}^{(r)}\big)\mathbf{x}^{(r)} \\ &= \Big(\mathbf{I} - \frac{2\eta}{m}\mathbf{X}^\top\mathbf{X}\Big)\mathbf{w} + \frac{2\eta}{m}\mathbf{X}^\top\mathbf{y}.\end{aligned}$$

Die Fixpunkte von $\mathcal{F}^{(\eta)}$ sind genau die Lösungen der Normalgleichungen (2), da $\mathcal{F}^{(\eta)}(\widehat{\mathbf{w}}) = \widehat{\mathbf{w}}$ genau dann gilt, wenn $\mathbf{X}^\top\mathbf{X}\widehat{\mathbf{w}} = \mathbf{X}^\top\mathbf{y}$. Der Update-Operator ist affin,

$$\mathcal{F}^{(\eta)}(\mathbf{w}) = \mathbf{M}\mathbf{w} + \mathbf{b},$$

mit linearem Anteil $\mathbf{M} = \mathbf{I} - (2\eta/m)\mathbf{X}^\top\mathbf{X}$ und Offset $\mathbf{b} = (2\eta/m)\,\mathbf{X}^\top\mathbf{y}$.

Wenn $\mathbf{X}$ vollen Spalten-rank hat, sodass $\mathbf{X}^\top\mathbf{X}$ positive definite (pd) ist, ist $\mathcal{F}^{(\eta)}$ ein contractive operator bezüglich der Euclidean norm für jede Schrittweite $0 < \eta < m/\lambda_{\max}$. Hier ist $\lambda_{\max}$ der größte Eigenwert von $\mathbf{X}^\top\mathbf{X}$. Die Eigenwerte des linearen Anteils $\mathbf{M}$ sind $1 - (2\eta/m)\,\lambda_j$, mit den positiven Eigenwerte $\lambda_j$ von $\mathbf{X}^\top\mathbf{X}$. Sie liegen genau für solche Schrittweiten streng zwischen $-1$ und 1. Der allgemeinere Fall einer beliebigen smooth, strongly convex Zielfunktion und Schrittweiten $\eta \leq m/(\lambda_{\min} + \lambda_{\max})$ wird von Nesterov (2004), Thm. 2.1.15 behandelt, wobei $\lambda_{\min}$ der kleinste Eigenwert von $\mathbf{X}^\top\mathbf{X}$ ist. Die fixed-point iteration konvergiert zur eindeutigen Lösung $\widehat{\mathbf{w}}$. Darüber hinaus wird die convergencesgeschwindigkeit der Gradientenabstieg durch das Verhältnis des größten zum kleinsten Eigenwert von $\mathbf{X}^\top\mathbf{X}$ bestimmt, d.h. durch ihre condition number $\kappa\big(\mathbf{X}^\top\mathbf{X}\big) = \lambda_{\max}/\lambda_{\min}$ (Boyd and Vandenberghe, 2004, Sect. 9.3).

Der obige Gradientenabstieg-Schritt verwendet den gesamten Trainingsdatensatz, um den Gradient zu berechnen. Wenn die Datenpunkte stattdessen sequenziell zu Zeitpunkten $t = 1, 2, \ldots$ eintreffen oder der Trainingsdatensatz zu groß ist, um in den Speicher zu passen, können die Modellparameter durch

online gradient descent (online GD) gelernt werden. Bei jedem Zeitschritt $t$ wendet sie einen einzelnen Gradientenabstieg-Schritt an. Dieser Schritt verwendet nur den eintreffenden Datenpunkt $\left(\mathbf{x}^{(t)}, y^{(t)}\right)$,

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + 2\eta\big(y^{(t)} - \left(\mathbf{w}^{(t)}\right)^\top \mathbf{x}^{(t)}\big)\mathbf{x}^{(t)}.$$

Dies ist der Gradientenabstieg-Schritt auf der Verlustfunktion $\left(y^{(t)} - \mathbf{w}^\top \mathbf{x}^{(t)}\right)^2$ des allein eintreffenden Datenpunkts, anstatt auf dem durchschnittlichen squared error loss $f(\mathbf{w})$. Es ist das least mean squares (LMS)-Update (Bishop, 2006, Sect. 3.1.3). Es vermeidet das Speichern des gesamten Trainingsdatensatzs, und seine Komplexität pro Iteration ist unabhängig von der Trainingsdatensatz-Größe $m$.

Die Optimalitätsbedingung (2) ermöglicht es, die stability der linearen Regression zu untersuchen. Idealerweise sind die gelernten Modellparameter unempfindlich gegenüber begrenzten Störungen des Trainingsdatensatzs. Ein gestörter Trainingsdatensatz ergibt eine Merkmalsmatrix $\widetilde{\mathbf{X}} = \mathbf{X} + \Delta\mathbf{X}$ und einen Label-Vektor $\widetilde{\mathbf{y}} = \mathbf{y} + \Delta\mathbf{y}$, mit Störungs-Matrix $\Delta\mathbf{X}$ und Vektor $\Delta\mathbf{y}$. Das Einsetzen dieser in die Normalgleichungen ergibt die gestörten Normalgleichungen

$$\widetilde{\mathbf{X}}^\top \widetilde{\mathbf{X}} \widetilde{\mathbf{w}} = \widetilde{\mathbf{X}}^\top \widetilde{\mathbf{y}}. \tag{4}$$

Die Matrix-Störungstheorie quantifiziert, wie stark $\widetilde{\mathbf{w}}$ von einer Lösung $\widehat{\mathbf{w}}$ der ungestörten Normalgleichungen (2) abweicht (Golub and Van Loan, 2013, Sect. 5.3.6).

Für eine reine Label-Störung, die ein einzelnes Label des Trainingsdatensatzs durch einen Ausreißer oder einen anderen korrumpierten Wert ersetzt, ist $\Delta\mathbf{X} = \mathbf{0}$, und die gestörten und ungestörten Normalgleichungen teilen sich die Koeffizienten-Matrix $\mathbf{X}^\top\mathbf{X}$. Angenommen, $\mathbf{X}$ hat vollen Spalten-rank, sodass $\mathbf{X}^\top\mathbf{X}$ invertible ist und die Lösung eindeutig ist. Das Subtrahieren der ungestörten Normalgleichungen (2) von den gestörten Normalgleichungen (4) ergibt

$$\widetilde{\mathbf{w}} - \widehat{\mathbf{w}} = \left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\Delta\mathbf{y} = \mathbf{X}^+\Delta\mathbf{y},$$

mit der pseudoinverse $\mathbf{X}^+$. Der Effekt der Störung des Trainingsdatensatzs wird dadurch quantifiziert,

$$\|\widetilde{\mathbf{w}} - \widehat{\mathbf{w}}\|_2 \leq \left\|\mathbf{X}^+\right\|_2 \|\Delta\mathbf{y}\|_2.$$

Wenn $\mathbf{X}$ vollen Spalten-rank hat, hat die pseudoinverse die Norm $\|\mathbf{X}^+\|_2 = 1/\sqrt{\lambda_{\min}(\mathbf{X}^\top\mathbf{X})} = \sqrt{\kappa(\mathbf{X}^\top\mathbf{X})}/\|\mathbf{X}\|_2$, wobei $\kappa\left(\mathbf{X}^\top\mathbf{X}\right) = \lambda_{\max}/\lambda_{\min}$ die condition number von $\mathbf{X}^\top\mathbf{X}$ ist (siehe pseudoinverse). Im Kontext von Fig. 1 ist $\mathbf{X} = \mathbf{1}$ (der Einsen-Vektor), sodass $\mathbf{X}^+\Delta\mathbf{y} = (1/m)\sum_{r=1}^{m}\Delta y^{(r)}$. Fig. 2 illustriert eine solche reine Label-Störung des Trainingsdatensatzs aus Fig. 1: Eine einzelne Label-Störung verschiebt $\widehat{w}$ um ein Drittel davon, wie Fig. 2 zeigt.

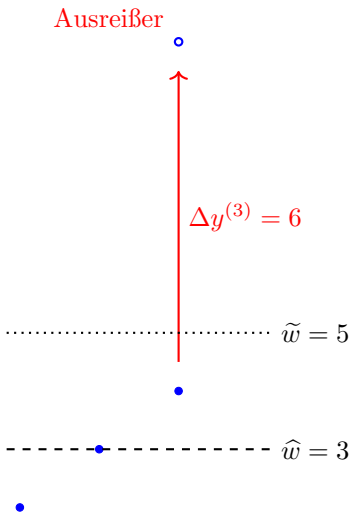


Abbildung 2: Die Störung $\Delta y^{(3)} = 6$ (roter Pfeil) ersetzt das dritte Label des Trainingsdatensatzs aus Fig. 1 durch einen Ausreißer: gefüllte Kreise bezeichnen die ursprünglichen Datenpunkte, wie in Fig. 1, und der offene Kreis den gestörten Wert. Der Ausreißer erhöht den Durchschnitt der Labels: der gelernte Modellparameter verschiebt sich um $\widetilde{w} - \widehat{w} = 6/3 = 2$, von der ungestörten Lösung $\widehat{w} = 3$ (gestrichelte Linie) zur gestörten Lösung $\widetilde{w} = 5$ (gepunktete Linie).

Der Effekt von Störungen kann auch auf der Ebene einer spezifischen Optimierungsverfahren untersucht werden. Zum Beispiel wird das Update der online GD zu

$$\begin{aligned}\mathbf{w}^{(t+1)} &= \mathbf{w}^{(t)} + 2\eta\big(\widetilde{y}^{(t)} - \big(\mathbf{w}^{(t)}\big)^\top \widetilde{\mathbf{x}}^{(t)}\big)\widetilde{\mathbf{x}}^{(t)} \\ &= \mathbf{w}^{(t)} + 2\eta\big(y^{(t)} - \big(\mathbf{w}^{(t)}\big)^\top \mathbf{x}^{(t)}\big)\mathbf{x}^{(t)} + \boldsymbol{\varepsilon}^{(t)}.\end{aligned}$$

Hier ist $\boldsymbol{\varepsilon}^{(t)}$ ein Störungsterm, der von der Datenpunkt-Störung $\big(\Delta\mathbf{x}^{(t)}, \Delta y^{(t)}\big)$ und den aktuellen Modellparameter $\mathbf{w}^{(t)}$ abhängt. Ein begrenzter Störungsterm verschiebt jede Iterierte um einen begrenzten Betrag. Das gestörte online GD-Update hat im Allgemeinen keinen Fixpunkt, da ein einzelner gestörter Datenpunkt $\widetilde{\mathbf{w}}$ nur dann unverändert lässt, wenn $\widetilde{\mathbf{w}}$ ihn interpoliert. Der Gradientenabstieg-Schritt-Operator $\mathcal{F}^{(\eta)}$, der auf dem gestörten Trainingsdatensatz berechnet wird, hat stattdessen als Fixpunkte genau die Lösungen der gestörten Normalgleichungen (4), durch die gleiche Berechnung wie für den ungestörten Trainingsdatensatz. Darunter ist die gestörte Lösung $\widetilde{\mathbf{w}}$, deren Verschiebung von $\widehat{\mathbf{w}}$ oben berechnet wurde.

Siehe auch: regression, lineares Modell, least squares, squared error loss, ERM, Gradientenabstieg, Ridge-Regression, Lasso, pseudoinverse.

## Querverweise

## Literatur


1. Bertsekas (2016). Nonlinear Programming. 3rd ed., Athena Scientific, Belmont, MA, USA.

2. Bishop (2006). Pattern Recognition and Machine Learning. Springer Science+Business Media, New York, NY, USA.

3. Boyd and Vandenberghe (2004). Convex Optimization. Cambridge Univ. Press, Cambridge, U.K.

4. Golub and Van Loan (2013). Matrix Computations. 4th ed., The Johns Hopkins Univ. Press, Baltimore, MD, USA.

5. Pedregosa et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12(85), pp. 2825–2830.

6. Lehmann and Casella (1998). Theory of Point Estimation. 2nd ed., Springer-Verlag, New York, NY, USA.

7. Meyer (2000). Matrix Analysis and Applied Linear Algebra. SIAM, Philadelphia, PA, USA.

8. Paszke et al. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: Adv. Neural Inf. Process. Syst., pp. 8026–8037. Curran Associates Inc.

9. Strang (2016). Introduction to Linear Algebra. 5th ed., Wellesley-Cambridge Press, Wellesley, MA, USA.

10. Hastie et al. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd ed., Springer Science+Business Media, New York, NY, USA.

11. Nesterov (2004). Introductory Lectures on Convex Optimization: A Basic Course. Kluwer Academic, Boston, MA, USA.

12. Papoulis and Pillai (2002). Probability, Random Variables, and Stochastic Processes. 4th ed., McGraw-Hill Higher Education, New York, NY, USA.