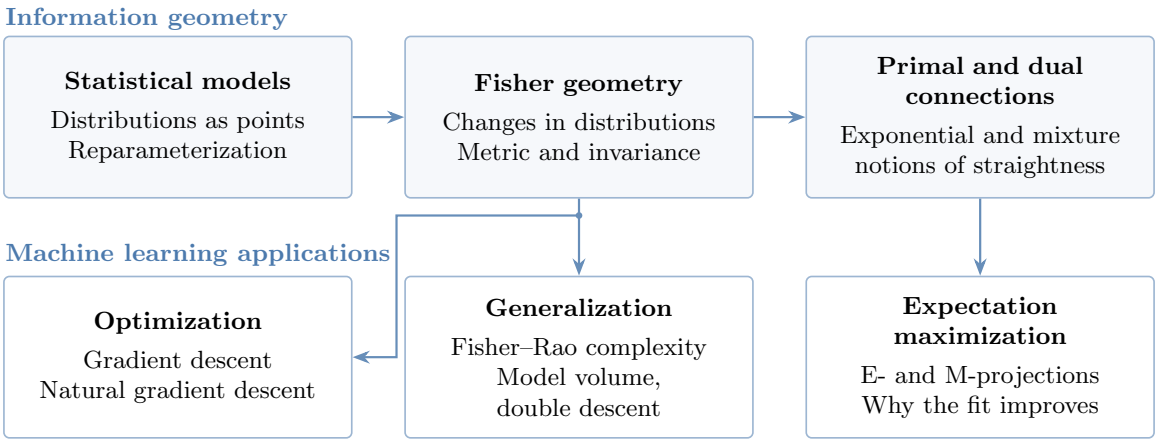
Introduction to Information Geometry

Pascal Esser

Ludwig-Maximilians-Universität München, Department of Mathematics,
Email: esser@math.lmu.de

Lecture overview



A Short Introduction to Machine Learning

Supervised Setting

Data consist of input–output pairs (X, Y) valued in $\mathcal{X} \times \mathcal{Y}$, governed by a joint distribution P. A hypothesis is a measurable map

$$h : \mathcal{X} \to \mathcal{Y}.$$

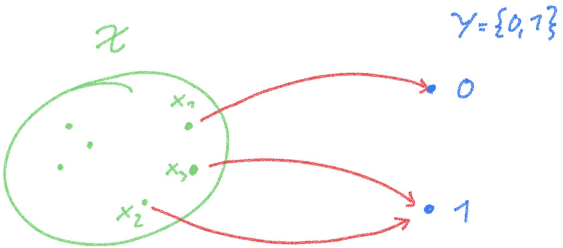
Given a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$, the goal is to minimize the *expected supervised risk*

$$\mathcal{R}_P(h) := \mathbb{E}_{(X,Y)\sim P}[\ell(h(X), Y)]\,.$$

However in practice we can only minimize it's empirical counterpart:

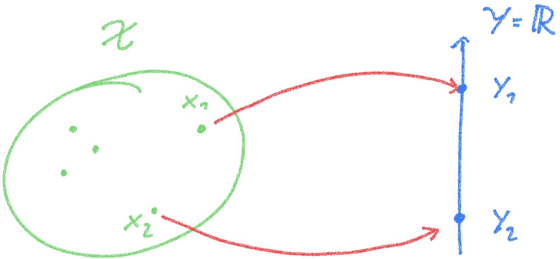
$$\mathcal{R}_{P_n}(h) := \frac{1}{n}\sum_{i=1}^{n} \ell(h(X_i), Y_i).$$

Classification



The data consist of input–output pairs (X, Y) valued in $\mathcal{X} \times \mathcal{Y}$, governed by a joint distribution P. Specifically we now consider $\mathcal{Y} = \{0, 1\}$.

Regression



The data consist of input–output pairs (X, Y) valued in $\mathcal{X} \times \mathcal{Y}$, governed by a joint distribution P. Specifically we now consider $\mathcal{Y} = \mathbb{R}$.

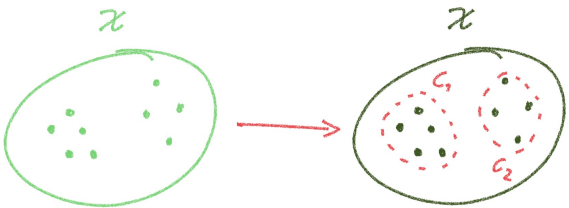
Unsupervised Learning

In unsupervised learning one observes only $X \sim P$. The goal is to infer latent structure such as clusters, a low-dimensional representation, or an approximate density model.

Let $\{p_\xi : \xi \in E\} \subset \mathcal{P}(\mathcal{X})$ be a parametric statistical model and let $D(\cdot\|\cdot)$ be a divergence on probability distributions. The learning task is

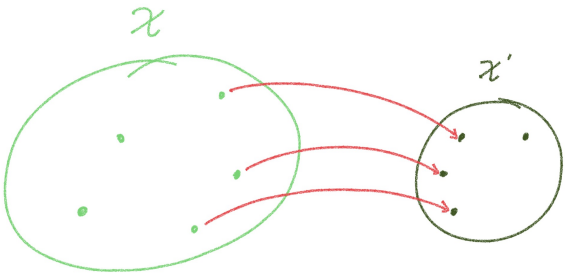
$$\xi^* \in \arg\min_{\xi \in E} D(P \,\|\, p_\xi)\,.$$

Clustering



Starting from an initial parameter space $\mathcal{X}$ we aim to enforce additional structures on the set of data-points. In the particular setting of clustering we aim to group together *"similar"* data-points.

Representation Learning



Starting from a data space $\mathcal{X}$ we compress the dimension of the data points into $\mathcal{X}'$,while preserving some metric on the learned representation.

Information Geometry

Information geometry (IG) is an interdisciplinary field that applies techniques from differential geometry to the study of probability theory and statistics.

It focuses on statistical manifolds, which are Riemannian manifolds whose points correspond to probability distributions.

Information Geometry

- *Intrinsic comparison of models.*
- *Generalization and capacity*
- *Optimization dynamics.*

Statistical Models

Parametric Models

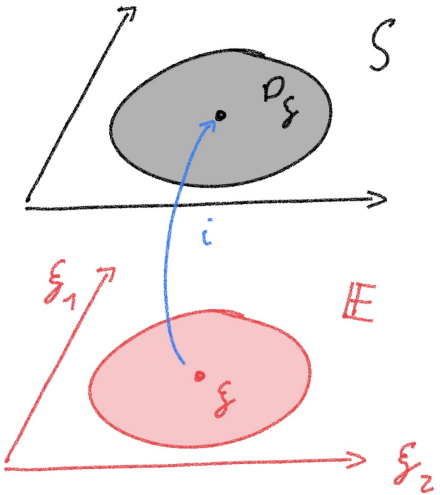
Let $\mathcal{S}$ be the surface of n real-valued variables $\xi^1, \ldots, \xi^n$ where $\mathcal{S}$ is a subset of the infinite dimensional space of functions $\mathcal{P}$.

Mapping between the $\mathbb{E}$ and $\mathcal{P}$ by

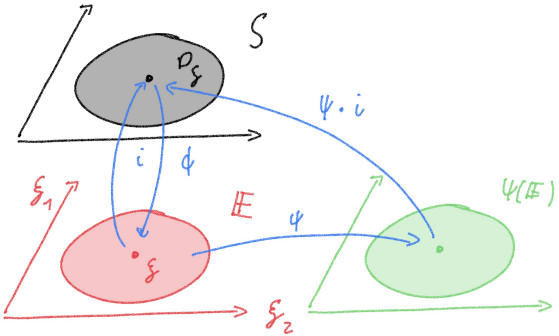
$$\iota : \mathbb{E} \to \mathcal{P}(\mathcal{X})$$
$$\iota(\xi) = p_\xi$$

For the mapping the following should hold:

- ι is a one-to-one mapping
- regularity on $\dim(\mathbb{E})$



Reparameterization



Now additionally consider the inverse function $\phi : \mathcal{S} \to \mathbb{R}^n,\ \phi(p_\xi) = \xi$. From there we add the reparameterization $\psi : \mathbb{E} \to \psi(\mathbb{E}) \subset \mathbb{R}^n$ to a new parameter space, which maps back to the probability space through $\psi \circ \phi : \mathcal{S} \to \psi(\mathbb{E}) \subset \mathbb{R}^n$.

Example: reparameterizing a two-dimensional Gaussian

Let $x \in \mathbb{R}^2$ and $\xi = (\xi^1, \xi^2) \in E = \mathbb{R}^2$. We vary the mean and fix $\Sigma = \mathrm{diag}(1/100, 1)$:

$$p_\xi(x) = \frac{5}{\pi} \exp\left\{ -\frac{1}{2}\left[100(x_1 - \xi^1)^2 + (x_2 - \xi^2)^2\right] \right\}.$$

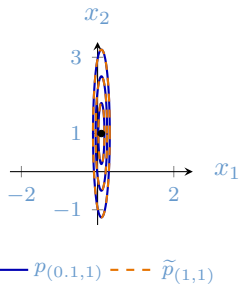
Choose the new coordinates

$$\rho = \psi(\xi) = (10\xi^1, \xi^2), \qquad \psi^{-1}(\rho) = (\rho^1/10, \rho^2).$$

The new density representation is

$$\widetilde{p}_\rho(x) := p_{\psi^{-1}(\rho)}(x), \qquad \boxed{\widetilde{p}_{\psi(\xi)}(x) = p_\xi(x)}.$$

For example, $\xi = (0.1, 1)$ and $\rho = (1, 1)$ describe *exactly the same density* on $\mathbb{R}^2$.



Equal-density contours coincide.

Examples of statistical families

- Bernoulli distributions
- Gaussian distributions
- Gaussian mixtures with fixed components
- Poisson distributions
- Beta distributions
- Categorical distributions
- Gamma distributions
- Binomial distributions with fixed $n \geq 2$
- Poisson mixtures with fixed components
- Dirichlet distributions
- Geometric distributions
- Exponential distributions

Definition (Exponential Family)

Consider $n+1$ real-valued smooth functions $C(x), F_i(x)$ on $\mathcal{X} \subset \mathbb{R}^k$ such that $1, F_1(x), F_2(x), \dots F_n(x)$ are linearly independent, where 1 denoted the constant function. Then define the normalization function:

$$\psi(\xi) = \ln \left(\int_{\mathcal{X}} \exp \left\{ C(x) + \xi^i F_i(x) \right\} dx \right).$$

From there we can define the *exponential family* of probability densities as

$$p(x; \xi) = \exp \left\{ C(x) + \xi^i F_i(x) - \psi(\xi) \right\}$$

with $x \in \mathcal{X}$ and ξ such that $\psi(\xi) < \infty$. Finally choose the parameter space $\mathbb{E}$ to be a non-empty set of elements ξ with $\psi(\xi) < \infty$.

Example: the Gaussian as an exponential family

Expanding the exponent gives

$$\log p_\xi(x) = \log \frac{5}{\pi} - \frac{1}{2}(100x_1^2 + x_2^2) + 100\xi^1 x_1 + \xi^2 x_2 - \frac{1}{2}\big(100(\xi^1)^2 + (\xi^2)^2\big).$$

Introduce the *natural coordinates* $\theta = (100\xi^1, \xi^2)$. Then

$$p_\xi(x) = \exp\Big\{C(x) + \theta^\top F(x) - \psi_{\exp}(\theta)\Big\},$$

where with $\theta \in \mathbb{R}^2$:

$$C(x) = \log \frac{5}{\pi} - \frac{1}{2}(100x_1^2 + x_2^2), \quad F(x) = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad \psi_{\exp}(\theta) = \frac{1}{2}\left(\frac{(\theta^1)^2}{100} + (\theta^2)^2\right),$$

Since $1, x_1, x_2$ are linearly independent and $\psi_{\exp}$ is finite on $\mathbb{R}^2$, this is a regular exponential family.

Definition (Mixture Family)

Let $F_i : \mathcal{X} \to \mathbb{R}$ be n smooth functions, linearly independent on $\mathcal{X}$, satisfying the integral constraint $\int_{\mathcal{X}} F_j(x)dx = 0 \quad 1 \leq j \leq n$. Consider another smooth function $C(x)$ on $\mathcal{X}$ with

$$\int_{\mathcal{X}} C(x) = 1.$$

From there we define the following family of probability densities

$$p(x;\xi) = C(x) + \xi^i F_i(x).$$

The resulting statistical manifold $\mathcal{S}$ is called a *mixture family* and ξ^j are its *mixture parameters.* The dimension of the parameter space $\mathbb{E}$ is called the *order* of the mixture family.

Exponential and mixture families

Family type	Examples
Exponential	Gaussian, Poisson, exponential, gamma, beta, Dirichlet, geometric, and binomial distributions (fixed $n \geq 2$).
Mixture	Gaussian mixtures and Poisson mixtures with fixed components and varying weights.
Both	Bernoulli distributions and categorical distributions.

- **Exponential:** the log-density is affine in natural parameters, apart from normalization.
- **Mixture:** the density is affine in mixture parameters.

Use interior parameter values. For the mixture examples, only the mixing weights vary. The two classes overlap.

Neural networks as statistical models

A NN becomes a statistical model once we specify the distribution of the output given the input.

$$\text{input } x \longrightarrow \text{network } h_\xi(x) \xrightarrow{\text{output}} \text{distribution } p_\xi(y \mid x)$$

Regression: Gaussian output
The network specifies the conditional mean.

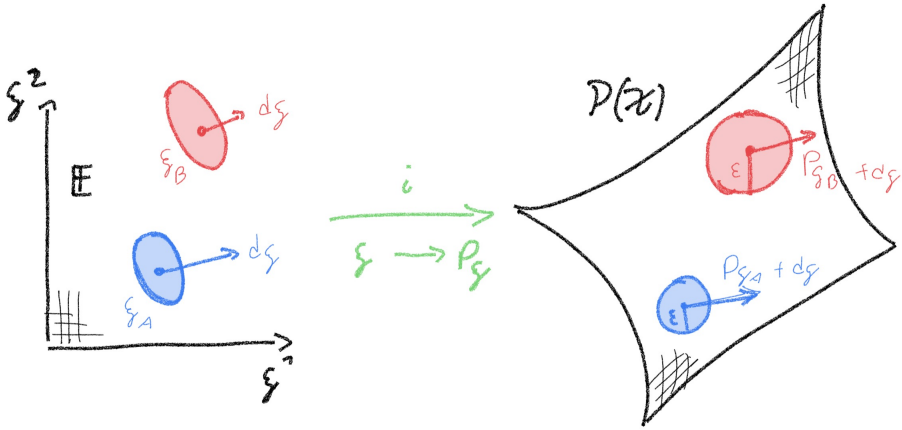
$$Y \mid X = x \sim \mathcal{N}(h_\xi(x), \sigma^2), \qquad \sigma^2 > 0 \text{ fixed.}$$

Fisher Information

The Fisher information matrix (FIM)

Quantifies how much information an observable random variable *carries about an unknown parameter by measuring the* sensitivity of the likelihood function to changes in that parameter*. More specifically, it captures the curvature of the log-likelihood.*

The Fisher information matrix (FIM)



$$\{\xi + w : g_{ij}(\xi) w^i w^j = \varepsilon^2\}$$

The Fisher information matrix (FIM)

Definition (FIM)

Let $\xi = (\xi^1, \dots, \xi^n) \in \mathbb{E}$ and the log likelihood function $l(\xi) = \ln p_\xi(x)$, then the Fisher Information Matrix (FIM) is defined by

$$g_{ij}(\xi) = E_\xi \left[\partial_i l(\xi) \partial_j l(\xi)\right] \quad \forall i, j \in [n]. \tag{1}$$

Example: deriving the Fisher information matrix

For $l(\xi) = \log p_\xi(x)$, the score is

$$\nabla_\xi l(\xi) = \begin{pmatrix} 100(x_1 - \xi^1) \\ x_2 - \xi^2 \end{pmatrix} = \Sigma^{-1}(x - \xi).$$

Since $\mathbb{E}_\xi[(X - \xi)(X - \xi)^\top] = \Sigma$, we obtain

$$G(\xi) = \mathbb{E}_\xi\left[\nabla_\xi l(\xi) \nabla_\xi l(\xi)^\top\right] = \Sigma^{-1}\Sigma\Sigma^{-1} = \begin{pmatrix} 100 & 0 \\ 0 & 1 \end{pmatrix}.$$

- $g_{11} = 100^2 \operatorname{Var}_\xi(X_1) = 100$ and $g_{22} = \operatorname{Var}_\xi(X_2) = 1$.
- $g_{12} = g_{21} = 0$, since the two coordinates are independent.

Intuition: the density is narrower in the first direction, so the same change in its mean is easier to detect there.

Some Properties I

Proposition (The FIM is a Riemannian metric)

Under the regularity and nondegeneracy assumptions the FIM on a statistical model is

1. *symmetric*
2. *positive define*

Example: the Fisher matrix defines a Riemannian metric

For tangent vectors $u, v \in T_\xi E \simeq \mathbb{R}^2$, define

$$g_\xi(u, v) := u^\top G(\xi) v = 100 u^1 v^1 + u^2 v^2.$$

- **Bilinear and symmetric:** linear in each argument, with $g_\xi(u, v) = g_\xi(v, u)$.
- **Positive definite:** $g_\xi(u, u) = 100(u^1)^2 + (u^2)^2 > 0$ for $u \neq 0$. Hence the metric is non-degenerate.
- **Smooth:** the entries of $G(\xi)$ are constant, hence smooth in ξ.

The resulting squared length of an infinitesimal displacement is

$$ds^2 = 100(d\xi^1)^2 + (d\xi^2)^2.$$

Some Properties II

The first key idea is that a change of variables in the sample space does not change the information content *about the parameter* ξ*, because Fisher information depends only on how the likelihood varies with* ξ*, not on how we label or parametrize the outcomes.*

Theorem (FIM is invariant w.r.t. sample space.)

The g_{ij} *is invariant under (invertible, parameter-independent) reparameterization of* $\mathcal{X}$*.*

Example: invariance under a change of sample coordinates

Keep the parameter ξ and transform the observations by

$$y = Ax, \qquad A = \begin{pmatrix} 10 & 0 \\ 0 & 1 \end{pmatrix}.$$

This transformation is invertible and independent of ξ. Thus

$$Y \sim \mathcal{N}(A\xi, I_2), \qquad q_\xi(y) = p_\xi(A^{-1}y)\,|\det A^{-1}| = \frac{1}{10} p_\xi(A^{-1}y).$$

The Jacobian contributes only a constant to the log-density:

$$\nabla_\xi \log q_\xi(y) = \nabla_\xi \log p_\xi(A^{-1}y).$$

Taking the expectation, with $Y = AX$, gives

$$G^Y(\xi) = G^X(\xi) = \begin{pmatrix} 100 & 0 \\ 0 & 1 \end{pmatrix}.$$

Intuition: we can recover every observation x from y, so the information about ξ is unchanged.

Some Properties III

The FIM is not tied to a specific choice of parameters — it behaves like a geometric metric, transforming predictably so that the underlying notion of statistical information remains invariant.

Theorem (FIM is covariant w.r.t. parameter space.)

g_{ij} *is covariant under reparameterization of* $\mathbb{E}$ *(i.e., transforms as a 2-covariant tensor).*

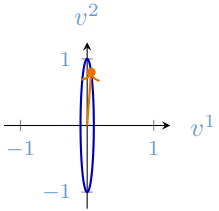
Example: covariance under reparameterization

Keep the observations x and use $\rho = (10\xi^1, \xi^2)$. The inverse Jacobian and transformed Fisher matrix are

$$J := \frac{\partial \xi}{\partial \rho} = \begin{pmatrix} 1/10 & 0 \\ 0 & 1 \end{pmatrix}, \qquad \boxed{\widetilde{G}(\rho) = J^\top G(\xi) J = I_2}.$$

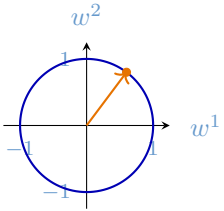
For corresponding tangent vectors $w = (10v^1, v^2)$, we have $v = Jw$ and $v^\top G v = w^\top \widetilde{G} w$.

$100(v^1)^2 + (v^2)^2 = 1$



ξ-coordinates: $v = (0.06, 0.8)$

$(w^1)^2 + (w^2)^2 = 1$



ρ-coordinates: $w = (0.6, 0.8)$

Same Fisher length: both orange vectors have squared length 1.

Optimization

Gradient descent: following the local slope

We choose parameters $\xi \in \mathbb{E}$ to minimize a smooth loss $\mathcal{L}(\xi)$.

- The gradient $\nabla_\xi \mathcal{L}$ collects the partial derivatives $\partial_i \mathcal{L}$ and points in the direction of *steepest Euclidean increase.*

- **Gradient descent (GD)** moves in the opposite direction:

$$\xi_{k+1} = \xi_k - \eta_k \nabla_\xi \mathcal{L}(\xi_k), \qquad \eta_k > 0.$$

Intuition: repeatedly look at the slope where you are and take a step downhill.

Why the parameter scale matters

GD measures the size of a step using Euclidean distance in parameter space.

- Equal changes in two parameters can have very different effects on the distribution p_ξ.
- Even changing the units of a parameter changes GD's behavior.

Example

write $\rho^1 = 10\xi^1$ and apply GD in the new coordinates. With the same learning rate, the step mapped back to ξ^1 becomes

$$\Delta\xi^1 = -\frac{\eta}{100}\,\partial_1\mathcal{L}(\xi).$$

The model is unchanged, but this coordinate now moves 100 times less.

Geometric motivation: measure a step by how much it changes the model distribution.

The Fisher metric measures changes in the model

Write $G(\xi) = (g_{ij}(\xi))$. We work with a regular model where $G(\xi)$ is positive definite. For a small parameter change δ, the Fisher metric gives

$$D_{\mathrm{KL}}(p_\xi \| p_{\xi+\delta}) = E_{p_\xi}\left[\log \frac{p_\xi(X)}{p_{\xi+\delta}(X)}\right] \approx \frac{1}{2}\delta^\top G(\xi)\delta.$$

- A direction with a large Fisher weight changes the distribution substantially, even for a small parameter step.

- A direction with a small Fisher weight allows a larger parameter step for the same statistical change.

Intuition: the Fisher metric provides a local ruler adapted to the model.

Natural gradient descent

The **natural gradient** adjusts the ordinary gradient using the Fisher metric:

$$\widetilde{\nabla}_\xi \mathcal{L}(\xi) = G(\xi)^{-1} \nabla_\xi \mathcal{L}(\xi).$$

Natural gradient descent (NGD) then takes the step

$$\xi_{k+1} = \xi_k - \eta_k \widetilde{\nabla}_\xi \mathcal{L}(\xi_k).$$

This accounts for how sensitive the model is in each direction. Sufficiently small steps decrease the loss when the gradient is nonzero.

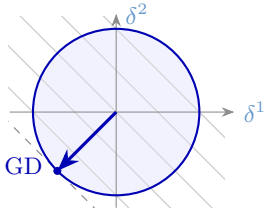
Amari (1998)

Two notions of a small step

Both methods minimize the same local change $\nabla_\xi \mathcal{L}(\xi)^\top \delta$, using a different notion of step size.

GD: small parameter change

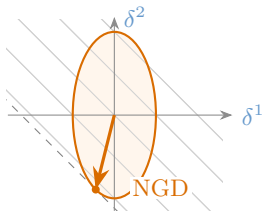
$$\delta^\top \delta \leq \varepsilon^2$$



$$\delta_{\text{GD}} \propto -\nabla_\xi \mathcal{L}$$

NGD: small distribution change

$$\delta^\top G(\xi) \delta \leq \varepsilon^2$$



$$\delta_{\text{NGD}} \propto -G(\xi)^{-1} \nabla_\xi \mathcal{L}$$

Gray lines: equal linearized loss. Illustration: $\nabla_\xi \mathcal{L} = (1, 1)^\top$, $G = \text{diag}(4, 1)$, $\varepsilon = 1$.

Intuition: the metric changes which parameter steps count as equally small.

Which curvature does NGD use?

- **Fisher matrix** $G(\xi)$**:** how strongly a parameter change changes the distribution, measured locally by KL divergence.
- **Loss Hessian** $H(\xi) = \nabla^2_\xi \mathcal{L}(\xi)$**:** how the slope of the loss changes as we move.

Interpretation: NGD accounts for statistical sensitivity. Its benefit depends on how that geometry relates to the loss.

Example: fitting the mean of a Gaussian

Consider a two-dimensional Gaussian with known covariance:

$$p_\xi = \mathcal{N}(\xi, \Sigma), \qquad \Sigma = \begin{pmatrix} 1/100 & 0 \\ 0 & 1 \end{pmatrix}.$$

The target distribution has mean zero. We minimize

$$\mathcal{L}(\xi) = D_{\mathrm{KL}}(p_0 \| p_\xi) = \frac{1}{2}\big(100(\xi^1)^2 + (\xi^2)^2\big).$$

- The distribution is narrow in the first direction and broad in the second.
- The same error in the mean matters more in the first direction. The loss therefore has a narrow valley leading to $\xi = 0$.

Example: correcting the unequal scales

For this Gaussian family, the Fisher matrix is

$$G = \Sigma^{-1} = \begin{pmatrix} 100 & 0 \\ 0 & 1 \end{pmatrix}.$$

The two gradients are therefore

$$\nabla_\xi \mathcal{L} = \begin{pmatrix} 100\xi^1 \\ \xi^2 \end{pmatrix}, \qquad G^{-1}\nabla_\xi \mathcal{L} = \begin{pmatrix} \xi^1 \\ \xi^2 \end{pmatrix}.$$

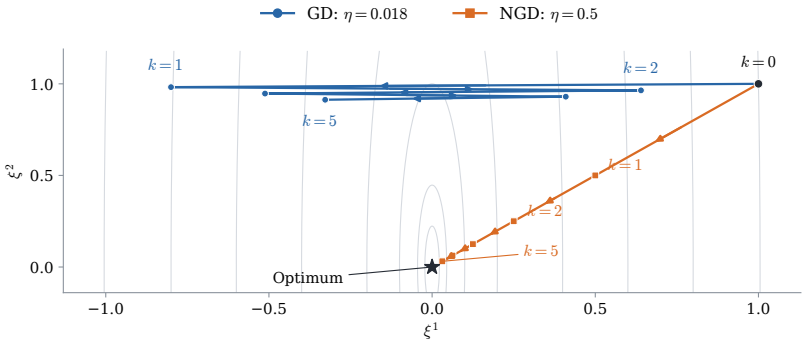
- **GD:** the steep first direction forces a small learning rate, leaving slow progress in the second direction.

- **NGD:** the factor $1/100$ corrects the first component. Both coordinates now contract by the same factor $1 - \eta$.

Here $G = H$, so the correction exactly balances the loss curvature.

Example: following the steps

Start at $\xi_0 = (1, 1)^\top$. For constant learning rates, convergence from every starting point requires $0 < \eta < 0.02$ for GD and $0 < \eta < 2$ for NGD.



The learning rates differ because the Fisher correction permits larger steps. This compares iterations, not runtime.

A saddle: uphill in one direction, downhill in another

Keep the same Gaussian family and $G = \mathrm{diag}(100, 1)$. Now use a *different loss*, represented locally by

$$\mathcal{L}_{\mathrm{s}}(\xi) = \frac{1}{2}\big(100(\xi^1)^2 - (\xi^2)^2\big).$$

- Along ξ^1, moving away from zero increases the loss. Descent pulls us toward zero.
- Along ξ^2, moving away from zero decreases the loss. Descent pushes us farther away once $\xi^2 \neq 0$.

The origin is a **saddle point**: it is a minimum along one axis and a maximum along the other.

At the exact saddle, both methods stop

At a stationary point ξ^*, the gradient vanishes:

$$\nabla_\xi \mathcal{L}(\xi^*) = 0 \quad \Longrightarrow \quad G(\xi^*)^{-1} \nabla_\xi \mathcal{L}(\xi^*) = 0.$$

- Both deterministic GD and NGD stay at the exact saddle when $G(\xi^*)$ is nonsingular.
- Fisher scaling can change the speed and direction of a nonzero gradient.
- A perturbation or noise can start movement away.

Takeaway: NGD can improve the local scaling of the problem, but does not guarantee escape from saddles or faster convergence.

Computational limitations

For n parameters, the Fisher matrix has size $n \times n$.

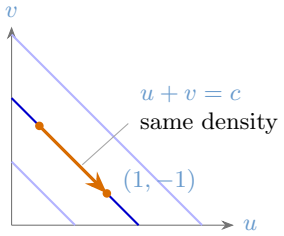
- **Estimating it:** computing all pairwise score products from B samples costs $O(Bn^2)$ after obtaining the scores.
- **Storing it:** a dense matrix requires $O(n^2)$ memory.
- **Using it:** solve $Gv = \nabla_\xi \mathcal{L}$ for the update direction. A general dense factorization costs $O(n^3)$.

Large models use approximations, such as diagonal or block matrices, K-FAC, or iterative solves.

When the Fisher matrix becomes singular

Consider two parameters that determine only one Gaussian mean:

$$p_{u,v} = \mathcal{N}(u+v, 1), \qquad G(u,v) = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \qquad \operatorname{rank} G = 1.$$



For $w = (1, -1)^\top$,

$$p_{u+t,v-t} = p_{u,v}, \qquad Gw = 0, \qquad w^\top G w = 0.$$

Moving along a blue line changes the parameters but preserves the distribution.

The Fisher matrix detects only the direction that changes $u + v$.

Meaning for learning: redundant weights can make G^{-1} undefined. One can use the identifiable coordinate $m = u + v$, or use damping in parameter space.

Generalization

Generalization: performance on new data

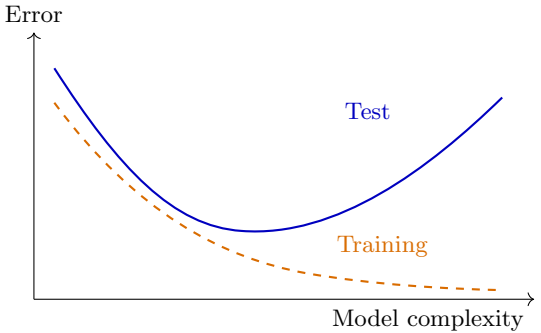
We train a hypothesis h_ξ using n independent samples from P. The goal is good performance on *new samples from the same distribution*.

$$R_{P_n}(h_\xi) = \frac{1}{n}\sum_{i=1}^{n} \ell(h_\xi(X_i), Y_i), \quad R_P(h_\xi) = \mathbb{E}_{(X,Y)\sim P}\big[\ell(h_\xi(X), Y)\big].$$

- **Training error:** how well the model fits the observed data.
- **Generalization gap:** $R_P(h_\xi) - R_{P_n}(h_\xi)$.

A flexible model can fit the training data, including their noise.

The classical picture: a U-shaped test error



The classical goal is to balance *fit* and *complexity*.

The Fisher–Rao norm

Consider a scalar-output network with L hidden layers, no biases, and linear or ReLU activations. For the predictive model $p_\xi(y \mid x) = \mathcal{N}(h_\xi(x), 1)$, define

$$\|\xi\|_{\mathrm{FR}} := \sqrt{\xi^\top G(\xi)\xi}, \qquad G(\xi) = \mathbb{E}_X\left[\nabla_\xi h_\xi(X) \nabla_\xi h_\xi(X)^\top\right].$$

Intuition: the Fisher–Rao norm weights the parameters by how strongly they affect the model's predictions.
For these homogeneous networks, this simplifies to

$$\frac{\|\xi\|_{\mathrm{FR}}}{L+1} = \sqrt{\mathbb{E}_X[h_\xi(X)^2]}.$$

Thus, the normalized norm measures the typical size of the *realized function*. At fixed depth, rescaling neurons without changing the function leaves this quantity unchanged.

Liang et al. (2019)

Generalization controlled by the Fisher–Rao norm

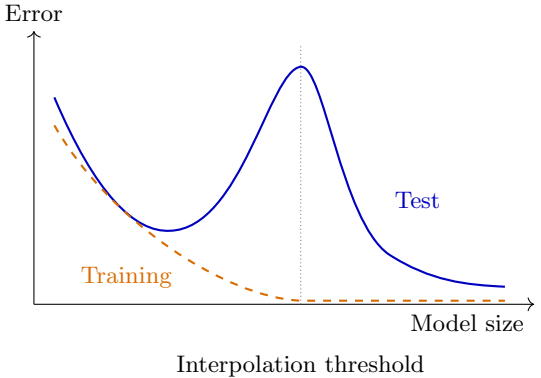
For **deep linear networks**, under mild loss and data assumptions. For n, d-dimensional data With probability at least $1-\delta$, every network satisfying $\|\xi\|_{\mathrm{FR}} \leq B_{\mathrm{FR}}$ obeys

$$R_P(h_\xi) \leq R_{P_n}(h_\xi) + \underbrace{\frac{2B_{\mathrm{FR}}}{L+1}\sqrt{\frac{d}{n}}}_{\text{Fisher–Rao complexity penalty}} + \sqrt{\frac{\log(1/\delta)}{2n}}$$

- **Smaller norm:** a tighter guarantee that training performance transfers to new data.
- **More samples:** the penalty decreases as $n^{-1/2}$. Network width does not enter explicitly.

Liang et al. (2019)

Double descent: test error can fall again



Parameter-count bounds, such as a penalty scaling like $\sqrt{D/n}$, can become uninformative for $D \gg n$. They can remain valid while missing the effect of the learned solution.

Geometric complexity: the volume of a model

Minimum description length (MDL): prefer a model that describes the data using a short code. Both the fit and the model's complexity matter.

$$V := \int_E \sqrt{\det G(\xi)}\, d\xi,$$

$$\mathrm{DL} \approx -\log p_{\hat{\xi}}(S_n) + \underbrace{\frac{\dim E}{2} \log\left(\frac{n}{2\pi e}\right)}_{\text{dimension penalty}} + \underbrace{\log V}_{\text{geometric correction}}.$$

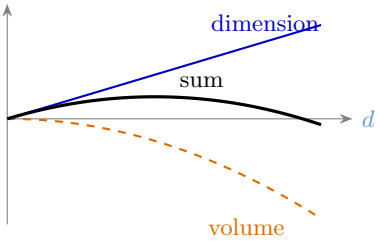
Here S_n denotes the observed sample.

- The Fisher volume measures how much space the model occupies in distribution space.
- A shrinking volume can offset an increasing dimension penalty.

Cheema–Sugiyama (2020)

Model volume and the complexity penalty

For a regular d-dimensional model with finite Fisher volume V_d, the preceding MDL approximation has complexity term



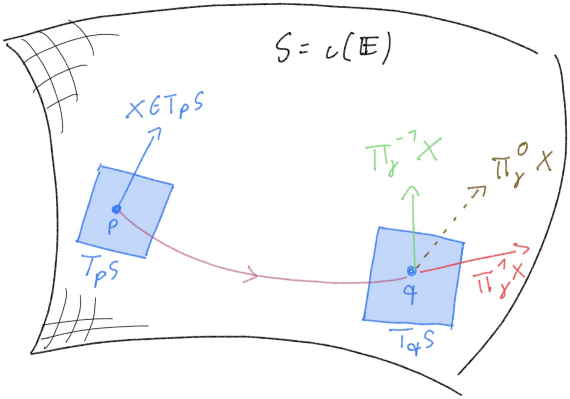
- The dimension penalty grows with d for sufficiently large n.
- In some model families, $\log V_d$ decreases as d grows.
- A sufficiently large decrease can offset the dimension penalty.

Scope: the expansion assumes large n at fixed d. This motivates a complexity mechanism for double descent. A test-error guarantee needs further analysis, especially for singular models.

Cheema–Sugiyama (2020)

Primal and Dual Connections

Connections: comparing directions at different models



Connections: comparing directions at different models

We work on the statistical model $S = \{p_\xi : \xi \in E\}$ with Fisher metric g.

- The metric measures the length of a tangent vector and the angle between two directions at the *same* model.

- A *connection* tells us how to compare directions as we move between models. It defines a derivative $\nabla_X Y$ of a vector field Y along X.

A curve $\gamma(t) = p_{\xi(t)}$ is *autoparallel* when its velocity stays parallel along the curve:

$$\nabla_{\dot\gamma}\dot\gamma = 0.$$

Intuition: a connection specifies what it means to keep moving in the same direction. Different connections can give different notions of straight motion.

Example: two descriptions of a Bernoulli model

For $x \in \{0, 1\}$, let $\mu \in (0, 1)$ be the probability of success:

$$p(x; \mu) = \mu^x (1 - \mu)^{1-x}.$$

The same family has the exponential representation

$$p(x; \xi) = \exp\{\xi x - \psi(\xi)\}, \quad \psi(\xi) = \log(1 + e^{\xi}),$$

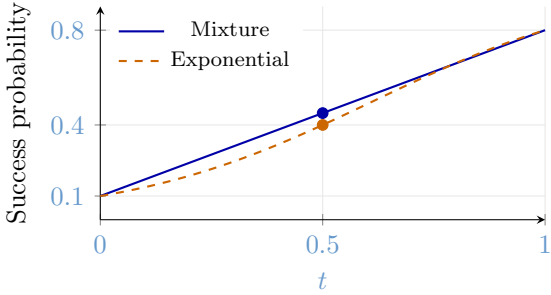
$$\xi = \log \frac{\mu}{1 - \mu}, \qquad \mu = \frac{e^{\xi}}{1 + e^{\xi}}.$$

- μ describes the **mean**: the expected fraction of successes.
- ξ describes the **log-odds**: how strongly the model favours success.

Equal steps in μ and equal steps in ξ change the distribution differently.

Example: two ways to interpolate

Between success probabilities $\mu_0 = 0.1$ and $\mu_1 = 0.8$, choose $t \in [0, 1]$.



Mixture: linear in μ,

$$\mu_m(t) = (1-t)\mu_0 + t\mu_1,$$
$$\mu_m(1/2) = 0.45.$$

Exponential: linear in ξ,

$$\xi_e(t) = (1-t)\xi_0 + t\xi_1,$$
$$\mu_e(1/2) = 0.40.$$

These are straight for $\nabla^{(-1)}$ and $\nabla^{(1)}$, respectively.

In this one-dimensional example, both traverse the same interval of distributions, with different parametrizations in t.

Dual connections: a shared Fisher pairing

Call $\nabla = \nabla^{(1)}$ the *primal* connection and $\nabla^* = \nabla^{(-1)}$ its *dual*. The naming fixes a convention for this section.

Definition. Two connections are dual with respect to g if, for all vector fields X, Y, Z,

$$Z\, g(X,Y) = g(\nabla_Z X, Y) + g(X, \nabla^*_Z Y).$$

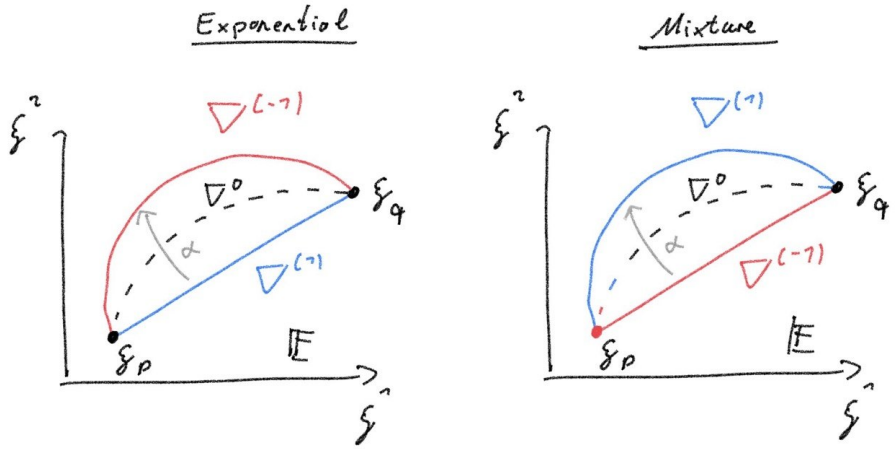
This is a product rule with one connection acting on each argument.

- Transport X using ∇ and Y using ∇^* along the same curve. Their Fisher inner product stays constant.

- Each transport may change lengths. The pair preserves the *cross-pairing* $g(X,Y)$.

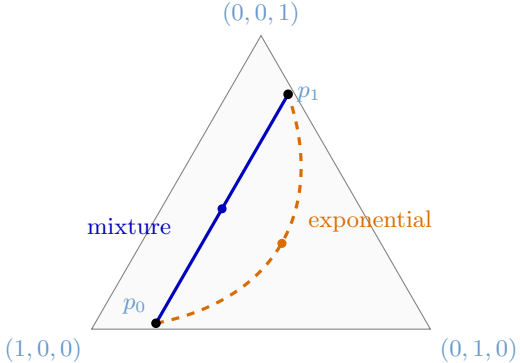
Intuition: the two rules compensate for one another as the Fisher metric changes.

Dual connections



Dual connections: two paths between the same models

For three outcomes, a distribution is a point $p = (p_1, p_2, p_3)$ inside the probability simplex.



Dots on the paths mark $t = 1/2$.

Mixture connection $\nabla^{(-1)}$ Linear interpolation of probabilities.

Exponential connection $\nabla^{(1)}$ Linear interpolation of the log-odds $\log(p_i/p_3)$, $i = 1, 2$.

Intuition: the connections give different notions of straightness. In two dimensions, even the paths through distribution space can differ.

Exponential and mixture connections: two notions of straightness

A connection is *flat* if we can locally choose coordinates in which its autoparallel curves are *straight lines.*

Exponential connection $\nabla^{(1)}$. Straight motion corresponds to linear interpolation of the natural parameters: $\xi(t) = (1-t)\xi_0 + t\xi_1$. The family is therefore $\nabla^{(1)}$**-flat**.

Mixture connection $\nabla^{(-1)}$. Linear interpolation of the mixture parameters gives $p_t = (1-t)p_0 + tp_1$. The family is therefore $\nabla^{(-1)}$**-flat**.

Intuition: exponential straightness interpolates natural parameters; mixture straightness interpolates probabilities. The two connections are dual with respect to the Fisher metric.

Expectation maximization: which coin generated the data?

In each round, choose coin A with probability π or coin B with probability $1-\pi$. Toss that coin twice.

- Observe $x \in \{0, 1, 2\}$, the number of heads. The coin label $z \in \{A, B\}$ stays hidden.
- The unknown head probabilities are a for coin A and b for coin B.

The **joint model** is

$$p(x, A) = \pi \binom{2}{x} a^x (1-a)^{2-x},$$
$$p(x, B) = (1-\pi) \binom{2}{x} b^x (1-b)^{2-x}.$$

Goal: fit $p(x) = p(x, A) + p(x, B)$ to f, the observed relative frequency, while estimating the missing coin assignments.

Two flat families in the space of joint distributions

The positive joint distributions form a **dually flat space**: both connections admit flat coordinates. Consider two families inside it.

Data family $\mathcal{D}$: all ways to fill in the missing labels,

$$\mathcal{D} = \{q : q(x, A) + q(x, B) = f(x)\}.$$

Linear constraints on probabilities make $\mathcal{D}$ **mixture-flat** ($\nabla^{(-1)}$).

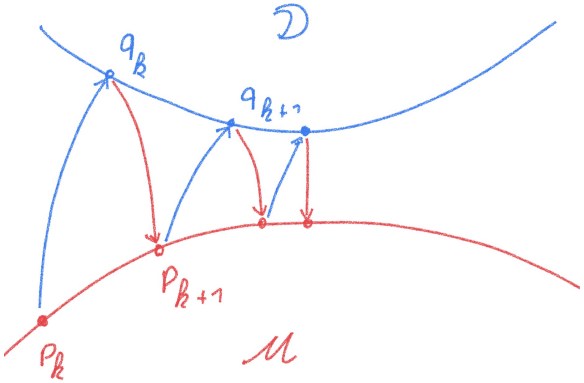
Model family $\mathcal{M}$: all joint two-coin models, with $0 < \pi, a, b < 1$. Their binomial structure is equivalent to

$$2 \log p(1, z) - \log p(0, z) - \log p(2, z) = \log 4, \quad z \in \{A, B\}.$$

Linear constraints on log-probabilities make $\mathcal{M}$ **exponential-flat** ($\nabla^{(1)}$).

EM alternates between the data and model families

Write $D(q\|p) = D_{\mathrm{KL}}(q\|p)$. EM alternately minimizes $D(q\|p)$ over $q \in \mathcal{D}$ and $p \in \mathcal{M}$.



E-step: complete the data using the current model.
M-step: fit a model to that completed data.

The E-step: a projection onto the data family

Keep the current model p_k fixed and choose the nearest completion:

$$q_k = \arg\min_{q \in \mathcal{D}} D(q \| p_k).$$

The solution has a simple form:

$$q_k(x, z) = f(x)\, p_k(z \mid x).$$

- Use the **observed frequencies** $f(x)$ for the visible data.
- Use the model's **conditional probabilities** $p_k(z \mid x)$ to share each observation between the two coins.

This is the **e-projection** onto the mixture-flat family $\mathcal{D}$.

Intuition: the model supplies soft assignments for the missing labels, while the visible frequencies agree exactly with the data.

The M-step: a projection back onto the model family

Keep q_k fixed and fit the joint model:

$$p_{k+1} = \arg\min_{p \in \mathcal{M}} D(q_k \| p) = \arg\max_{p \in \mathcal{M}} \sum_{x,z} q_k(x,z) \log p(x,z).$$

This is the **m-projection** onto the exponential-flat family $\mathcal{M}$. It is ordinary maximum likelihood with fractional counts.

$$\pi_{k+1} = \frac{\text{expected rounds assigned to } A}{\text{sample size}},$$

$$a_{k+1} = \frac{\text{expected heads assigned to } A}{2\,\text{expected rounds assigned to } A}, \qquad \text{and similarly for } b_{k+1}.$$

Intuition: refit each coin using the observations it is expected to explain.

Example: one EM iteration

Observe 100 rounds with head counts $(20, 30, 50)$ for $x = 0, 1, 2$. Start with $(\pi_0, a_0, b_0) = (0.5, 0.25, 0.75)$.

E-step: assign each round fractionally to the two coins.

Heads x	Observed rounds	$p_0(A \mid x)$	Assigned to A	Assigned to B
0	20	0.9	18	2
1	30	0.5	15	15
2	50	0.1	5	45
Total	100		38	62

M-step: refit using these fractional counts.

$$\pi_1 = \frac{38}{100} = 0.38, \quad a_1 = \frac{15 + 2 \cdot 5}{2 \cdot 38} \approx 0.329, \quad b_1 = \frac{15 + 2 \cdot 45}{2 \cdot 62} \approx 0.847.$$

For the observed frequencies, the fit improves:

$$D_{\mathrm{KL}}(f \| p_0(x)) \approx 0.0788, \qquad D_{\mathrm{KL}}(f \| p_1(x)) \approx 0.00203.$$

Why the dual projections improve the fit

For positive distributions in this example, the **Pythagorean identities** read

$$D(q\|p_k) = D(q\|q_k) + D(q_k\|p_k), \qquad q \in \mathcal{D},$$
$$D(q_k\|p) = D(q_k\|p_{k+1}) + D(p_{k+1}\|p), \quad p \in \mathcal{M}.$$

An e-projection meets the mixture-flat family Fisher-orthogonally. An m-projection meets the exponential-flat family Fisher-orthogonally.

Each step preserves or reduces the joint KL divergence. At the E-step optimum, $D(q_k\|p_k) = D(f\|p_k(x))$, so

$$D(f\|p_{k+1}(x)) \le D(f\|p_k(x)).$$

Meaning for learning: the observed-data likelihood cannot decrease. EM can still converge to a local optimum.

Discussion

Some open problems in IG for machine learning

- *Which geometric information is essential?*
  Can we retain the useful parts of the FIM cheaply enough to improve training time?
- *How should we describe redundant models?*
  Can we track the directions that change predictions during training?
- *When does geometry predict test performance?*
  How can we connect the geometry of the model to generalization and double descent?
- *How far can the projection viewpoint extend?*
  Can primal and dual geometry guide useful algorithms beyond flat families?

Some further readings

- *"An elementary introduction to information geometry"* by F. Nielsen
- *"Information Geometry and Its Applications"* by S. Amari
- *"Geometric Modeling in Probability and Statistics"* by O. Calin, C. Udrişte

If you want to discuss further or are interested in some project in this direction send me an email *esser@math.lmu.de*