ECE 598: Computational Power Systems **Winter 2025**

Lecture 10: Decision-focused Analytic

Lecturer: Vladimir Dvorkin *Scribe(s): Qingyuan Xu & Jo Brooks & Wei Ai*

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

10.1 Closed-Loop Analytics

Forecasting is the first step in any closed-loop analytics framework. It serves two essential purposes: (1) reducing uncertainty in model inputs, and (2) increasing confidence in operational decisions.

We distinguish between two types of forecasting frameworks: open-loop analytics and closed-loop analytics. In open-loop systems, a forecast w is used to make a decision $x^\star$ by solving an optimization problem of the form:

$$\min_{x\in\mathcal{X}(w)} c(x,w),$$

but the decision outcome does not influence future forecasts. In contrast, closed-loop analytics introduces a feedback mechanism: the decision outcome $x^\star$ is used to update or inform the forecasting model. This feedback loop enables continual improvement of forecast quality based on the performance of past decisions.

Decision feedback is a defining feature of closed-loop analytics. It allows the forecasting model to learn from the quality of decisions, encouraging joint optimization of both prediction and decision-making components.

10.2 Example: Wind Power Forecast and Electricity Pricing

To understand how forecasting affects electricity markets, consider a case where wind power forecast $\hat{w}$ is used to solve a DC Optimal Power Flow (DC-OPF) problem. The objective is to minimize total generator dispatch cost:

$$\min_{\underline{p}\le p\le\overline{p}} p^\top Cp + c^\top p$$

subject to power balance and transmission constraints:

$$\mathbf{1}^\top(p+\hat{w}-d)=0 \quad \text{(power balance)},$$

$$|F(p+\hat{w}-d)|\le \bar{f} \quad \text{(power flow limits)}.$$

From the dual solution of this optimization problem, we derive the Locational Marginal Prices (LMPs), denoted by $\pi(\hat{w})$, which incorporate both a system-wide uniform price and location-specific adjustments due to congestion:

$$\pi(\hat{w}) = \lambda_b\cdot\mathbf{1} - F^\top(\lambda_f^+ - \lambda_f^-).$$

These LMPs are uniquely determined by the forecast $\hat{w}$, under reasonable assumptions of system convexity.

The error in LMPs—defined as $\delta\pi = \pi(\hat{w}) - \pi(w)$—represents the difference between forecast-induced prices and those resulting from the actual realization of wind power w. This type of analysis can only be done retrospectively, as it requires knowing the actual wind output.

Two sources of error impact the final LMPs: (1) inaccuracies in the wind forecast itself, and (2) errors introduced during the conversion of wind energy into mechanical and ultimately electrical power. Depending on the direction of these errors, system operators or consumers may be disproportionately affected.

Furthermore, because LMPs are derived from network topology and flow constraints, there is inherent randomness and potential unfairness in how forecast errors translate to pricing errors across different locations. In some areas of the grid, particularly where congestion is frequent, small forecast deviations can have a large pricing impact.

This spatial sensitivity is evident in simulations on the IEEE 118-bus system with a wind farm at bus 39. LMP errors vary across the grid, with the slack bus consistently showing the lowest error. This occurs because the slack bus is excluded from certain rows in the power flow matrix F, effectively protecting it from the direct effects of wind power forecast deviations.

10.3 Integrating forecasting and decision-making

10.3.1 Forecast Problem

We start with a dataset of input-output pairs

$$\mathcal{D} = \{(\varphi_1, w_1), \ldots, (\varphi_k, w_k)\},$$

where each φ_i is a feature vector (e.g., weather data) and w_i is the observed output (e.g., wind power).

A forecasting model W_θ maps input features φ to predictions $\hat{w}$, where θ are the parameters to be learned:

$$W_\theta(\varphi) = \hat{w}.$$

In prediction-focused learning, we find θ by minimizing the average squared error between predictions and true values:

$$\min_\theta \ \frac{1}{2k} \sum_{i=1}^{k} \|W_\theta(\varphi_i) - w_i\|_2^2.$$

10.3.2 Decision Problem and Regret

We consider a parametrized optimization problem formulated as:

$$\begin{aligned} \min_{x \in \mathrm{R}^n} \quad & c(x) \\ \text{s.t.} \quad & g(x, w) \leq 0 \end{aligned}$$

Here, $x = (x_1, \ldots, x_n) \in \mathrm{R}^n$ is the decision variable, $c(x)$ is the objective function, and $g(x, w)$ is a constraint function that depends on the input w.

The decision problem is therefore parametrized by the forecasted input $\widehat{w}$. Since the constraints depend on w, the feasible region and the optimal solution both change accordingly. The solution $x^\star$ becomes a function of w, written as $x^\star(w)$.

Decision regret measures the loss from using an imperfect forecast. It is defined as the difference between the objective function values based on the predicted input $\widehat{w}$ and the actual realization w:

$$\text{Regret} = c\left(x^\star(\widehat{w})\right) - c\left(x^\star(w)\right)$$

This reflects how forecast error affects decision performance.

10.3.3 Decision-Focused Learning

The decision-focused objective minimizes the average decision regret across k historical records, resulting in the following bilevel optimization structure:

$$\begin{aligned}
\min_{\theta} \quad & \frac{1}{2k}\sum_{i=1}^{k} \left\| c\left(x^\star(\widehat{w}_i)\right) - c\left(x^\star(w_i)\right)\right\|_2^2 & \\
\text{subject to} \quad & \widehat{w}_i = \mathcal{W}_\theta(\varphi_i) & \forall i = 1,\ldots,k \\
& x^\star(\widehat{w}_i) = \arg\min_{x} \; c(x) & \\
& \text{subject to} \quad g(x, \widehat{w}_i) \leq 0 & \forall i = 1,\ldots,k
\end{aligned}$$

The upper-level problem optimizes the parameters θ of the prediction model $\mathcal{W}_\theta$, using features φ_i. The lower-level problem computes the optimal decision $x^\star$ based on the forecasted value $\widehat{w}_i$ for each sample i.

This formulation allows end-to-end learning focused directly on decision quality, rather than prediction accuracy alone.

This structure raises two key questions: How is this different from standard prediction-focused learning? And how can we efficiently solve the resulting bilevel optimization?

10.4 Solution Methodology

10.4.1 KKT-based solution

10.4.2 Gradient descent-based solution

Recall the optimization objective:

$$\min_{\theta} \; \mathcal{R} = \frac{1}{2k}\sum_{i=1}^{k} \left\| c\left(x^\star(\mathbb{W}_\theta(\varphi_i))\right) - c_i^\star\right\|_2^2$$

Taking derivative of the regret with respect to model parameter θ by chain rule

$$\frac{\partial \mathcal{R}}{\partial \theta} = \frac{1}{k}\sum_{i=1}^{k} \left(c\left(x_i^\star\right) - c_i^\star\right)^\top \cdot \frac{\partial c(x_i^\star)}{\partial x_i^\star} \cdot \frac{\partial x_i^\star}{\partial \mathbb{W}_\theta(\varphi_i)} \cdot \frac{\partial \mathbb{W}_\theta(\varphi_i)}{\partial \theta}$$

Regret is then minimized over iterations:

$$\theta \leftarrow \theta - \alpha \frac{\partial \mathcal{R}}{\partial \theta}$$

The challenging part is to compute the last unknown gradient $\frac{\partial x^\star}{\partial \mathbb{W}_\theta(\varphi_i)}$. The key idea to tackle this problem is **implicit differentiation** for optimization-based models :

Although $x^\star$ is not given in direct-form, it is an optimal solution to an optimization problem, thus satisfies a set of optimality conditions (e.g., KKT conditions). These can be differentiated implicitly to track how $x^\star$ changes with respect to the data.

Example: Implicitly Differentiation for Convex QPs

We consider the following convex quadratic program:

$$\begin{aligned} \min_x \quad & \frac{1}{2}x^\top Cx + c^\top x \\ \text{s.t.} \quad & Ax = b, \quad Gx \leq h \end{aligned}$$

Its Lagrangian is:

$$\mathcal{L}(x, \nu, \lambda) = \frac{1}{2}x^\top Cx + c^\top x + \nu^\top (Ax - b) + \lambda^\top (Gx - h)$$

We apply the KKT conditions:

$$\begin{aligned} \text{Stationarity:} \quad & Cx^\star + c + A^\top \nu^\star + G^\top \lambda^\star = 0 \\ \text{Primal feasibility:} \quad & Ax^\star - b = 0 \\ \text{Complementarity:} \quad & \text{diag}(\lambda^\star)(Gx^\star - h) = 0 \end{aligned}$$

We differentiate the KKT conditions using the product rule. For example,

$$\frac{d}{db}(Cx^\star) = dC \cdot x^\star + C \cdot dx^\star$$

Applying this rule to all terms, the differential form of the KKT system is:

$$\begin{bmatrix} C & G^\top & A^\top \\ \text{diag}(\lambda^\star)G & \text{diag}(Gx^\star - h) & 0 \\ A & 0 & 0 \end{bmatrix} \begin{bmatrix} dx \\ d\lambda \\ d\nu \end{bmatrix} = \begin{bmatrix} -dC \cdot x^\star - dc - dG^\top \lambda^\star - dA^\top \nu^\star \\ -\text{diag}(\lambda^\star) \cdot dGx^\star + \text{diag}(\lambda^\star)dh - \text{diag}(Gx^\star - h)d\lambda \\ -dAx^\star + db \end{bmatrix}$$

Suppose \$b\$ is given by $b = \mathbb{W}_\theta(\varphi_i)$.

Set all other differentials to zero ($dC = dG = dA = dc = 0$, etc.), and let $db = I$.

Then the system simplifies to:

$$\begin{bmatrix} C & G^\top & A^\top \\ \text{diag}(\lambda^\star)G & \text{diag}(Gx^\star - h) & 0 \\ A & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} dx \\ d\lambda \\ d\nu \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix}$$

Solve this linear system to obtain:

$$\frac{\partial x^\star}{\partial b} = \frac{\partial x^\star}{\partial \mathbb{W}_\theta(\varphi_i)} = (A^T)^{-1}$$

Remarks:

- Requires solving an optimization problem at every iteration
  (Q. Why? Because $x^\star$ is defined via an optimization problem! Recall that we require to solve $c(x^*)$ at each update !)
- More versatile: no structural assumptions are needed on the model $\mathbb{W}_\theta$

  (Thus we can adapt advance machine learning framework for gradient-based decision focused learnning)
- Also scalable to larger problems in terms of:

$$\text{number of variables } n, \quad \text{constraints } m, \quad \text{samples } k$$

10.4.2.1 Example : Electricity pricing and Wind Forecasting

Motivation: Forecasting wind generation for market operation impacts locational marginal prices (LMPs). Traditional models minimize forecast error, but may unintentionally lead to unfair or biased pricing outcomes.

Figure in slide 15

We are accessible to historical data set $\{(\varphi_i, \mathbf{w}_i)\}_{i=1}^k$ – wind generation records with features (wind speed, direction, pitch angle, etc) and true generation $\mathbf{w}_i$.

We compare two models performance :

- **DeepWP:** Standard neural network trained via $\|\widehat{\mathbf{w}} - \mathbf{w}\|$.
- **DeepWP$^+$:** Price-aware model with loss $\|\widehat{\mathbf{w}} - \mathbf{w}\| + \|\pi(\widehat{\mathbf{w}}) - \pi(\mathbf{w})\|$ where $\pi(\cdot)$ is derived from a market-clearing optimization layer. DeepWP$^+$ reduces downstream pricing errors even at the cost of slightly biased power predictions.

Setup: Models are deployed on the IEEE 118-bus system with PowerModels.jl and a market-clearing solver.

Result: Heatmaps show *mean LMP error* $\delta\pi$ across buses:

Figure in slide 18

- **DeepWP:** High spatial disparities, with $\delta\pi \in [-4, 1]$ \$/MWh
- **DeepWP$^+$:** Reduced disparities, $\delta\pi \in [-1, 1]$ \$/MWh

Network topology amplifies price gaps due to congestion and uneven access to generation. **Decision-focused learning can mitigate—but not eliminate—these disparities.**

Figure in slide 19

We also notice that DeepWP$^+$ sacrifices point-wise accuracy in favor of improved price outcomes. $\widehat{\mathbf{w}}$ from DeepWP$^+$ exhibits intentional bias—over-predicting in specific wind speed ranges to reduce downstream price errors.

Figure in slide 20

We further examine training dynamics : RMSE of forecast $\widehat{\mathbf{w}}$ continues to drop for DeepWP; while RMSE of price $\pi(\widehat{\mathbf{w}})$ drops significantly under DeepWP$^+$.

10.4.2.2 Example in Tutorial

We are now focusing on a standard regression problem:

$$\hat{y}_i = \omega^\top x_i \quad \Rightarrow \quad \min_{\omega} \ \frac{1}{2k} \sum_{i=1}^{k} \left(\omega^\top x_i - y_i\right)^2$$

Gradient descent update in the form of

$$\omega \leftarrow \omega - \eta \cdot \frac{1}{k} \sum_{i=1}^{k} \left(\omega^\top x_i - y_i\right) x_i$$

Suppose our downstream decision problem is

$$\min_{z \geq 0} \ \frac{1}{2}(z - \hat{y})^2 + \lambda z^2$$

which has closed form solution

$$z^\star(\hat{y}) = \max\left(0, \frac{\hat{y}}{1 + 2\lambda}\right)$$

We now use the decision output as training target:

$$\min_{\omega} \ \frac{1}{2k} \sum_{i=1}^{k} \left(z^\star(\omega^\top x_i) - z^\star(y_i)\right)^2$$

Using chain rule:

$$\frac{d}{d\omega} \ z^\star(\omega^\top x_i) = \frac{dz^\star}{d\hat{y}_i} \cdot \frac{d\hat{y}_i}{d\omega} = \max\{\frac{1}{1 + 2\lambda}, 0\} \cdot x_i$$

Thus gradient descent update becomes:

$$\omega \leftarrow \omega - \eta \cdot \frac{1}{k} \sum_{i=1}^{k} \left(z^\star(\omega^\top x_i) - z^\star(y_i)\right) \cdot \max\{\frac{1}{1 + 2\lambda}, 0\} \cdot x_i$$

$$= \omega - \eta \cdot \frac{1}{k} \sum_{i=1}^{k} \left(\max\left(0, \frac{\omega^T x_i}{1 + 2\lambda}\right) - z^\star(y_i)\right) \cdot \max\{\frac{1}{1 + 2\lambda}, 0\} \cdot x_i$$

10.4.3 Multiparametric Programming (MPP)

In large-scale applications with more than thousands of variables and constraints, repeatedly solving the full optimization model during each iteration (e.g., in stochastic gradient descent) can be computationally

prohibitive. Multiparametric Programming (MPP) addresses this challenge by parameterizing the DC-OPF problem with respect to the wind power forecast, $\hat{w}$, and pre-solving for the dependence of the optimal solution on $\hat{w}$.

Compact DC-OPF Formulation We begin by introducing a typical DC-OPF formulation where p is the generator dispatch vector, $\hat{w}$ is the wind power forecast, and d is the demand:

$$\begin{aligned} \min_{p} \quad & \frac{1}{2}\, p^\top C\, p \;+\; c^\top p \\ \text{subject to} \quad & \underline{p} \;\le\; p \;\le\; \overline{p} \\ & \mathbf{1}^\top \big(p + \hat{w} - d\big) \;=\; 0 \\ & \big|F\big(p + \hat{w} - d\big)\big| \;\le\; \overline{f} \end{aligned}$$

Here,

- C and c are cost coefficients,
- $\underline{p}$ and $\overline{p}$ are the lower and upper generation limits respectively,
- F is the power flow distribution matrix,
- $\overline{f}$ is the vector of transmission limits.

The compact formulation of the DC-OPF problem is given by:

$$\begin{aligned} \min_{p} \quad & \frac{1}{2}\, p^\top C p + c^\top p \\ \text{s.t.} \quad & A\, p \le B\, \hat{w} + b \end{aligned}$$

Matrices A, B and vector b are constructed by collecting all of the constraints from the original DC-OPF model:

$$A = \begin{pmatrix} I \\ -I \\ -\mathbf{1}^\top \\ F \\ -F \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ 0 \\ \mathbf{1}^\top \\ F \\ -F \end{pmatrix}, \quad b = \begin{pmatrix} \overline{p} \\ -\underline{p} \\ -\mathbf{1}^\top d \\ \overline{f} \\ \overline{f} \end{pmatrix}$$

Here the power balance equality constraint is replaced with an inequality because in order to minimize costs, the optimization will automatically push the slack to zero.

Derivation and Closed-Form Solution The Lagrangian function is defined as

$$L(p, \lambda) = \frac{1}{2}\, p^\top C\, p + c^\top p + \lambda^\top \Big(\tilde{A}\, p - \tilde{B}\, \hat{w} - \tilde{b}\Big),$$

where $\lambda \in \mathbb{R}^k$ is the vector of dual variables.

The stationarity and primal feasibility conditions for optimality (assuming C is invertible) are:

$$C\,p + c + \tilde{A}^\top \lambda = 0 \tag{1a}$$

$$\tilde{A}\,p - \tilde{B}\,\hat{w} - \tilde{b} = 0 \tag{1b}$$

Assuming C is invertible, we obtain

$$p = -C^{-1}\Big(c + \tilde{A}^\top \lambda\Big) \tag{2}$$

Substitute (2) into (1b):

$$\tilde{A}\,p - \tilde{B}\,\hat{w} - \tilde{b} = \tilde{A}\Big[-C^{-1}\Big(c + \tilde{A}^\top \lambda\Big)\Big] - \tilde{B}\,\hat{w} - \tilde{b} = 0$$

The solution for λ is:

$$\lambda = -\Big(\tilde{A}\,C^{-1}\,\tilde{A}^\top\Big)^{-1}\tilde{B}\,\hat{w} - \Big(\tilde{A}\,C^{-1}\,\tilde{A}^\top\Big)^{-1}\Big(\tilde{b} + \tilde{A}\,C^{-1}\,c\Big)$$

For simplicity, we define

$$D = -\Big(\tilde{A}\,C^{-1}\,\tilde{A}^\top\Big)^{-1}\tilde{B}, \quad d = -\Big(\tilde{A}\,C^{-1}\,\tilde{A}^\top\Big)^{-1}\Big(\tilde{b} + \tilde{A}\,C^{-1}\,c\Big)$$

Thus, we obtain the affine expression for the dual variable:

$$\lambda = D\,\hat{w} + d \tag{3}$$

Returning to (2) and substituting (3) in, we have:

$$p = -C^{-1}\Big[c + \tilde{A}^\top\big(D\,\hat{w} + d\big)\Big] = -C^{-1}\,\tilde{A}^\top D\,\hat{w} - C^{-1}\,c - C^{-1}\,\tilde{A}^\top d$$

Defining the following terms:

$$\Pi = -C^{-1}\,\tilde{A}^\top\,D, \quad \pi = -C^{-1}\,c - C^{-1}\,\tilde{A}^\top\,d,$$

we obtain the affine mapping for p:

$$p = \Pi\,\hat{w} + \pi \tag{4}$$

In summary, under the assumption that the set of active constraints (represented by $\tilde{A}$, $\tilde{B}$, $\tilde{b}$) remains fixed over a region of the parameter space, the optimal solution (p, λ) of the quadratic program can be expressed as affine functions of the parameter $\hat{w}$:

$$\boxed{\begin{aligned}\lambda &= D\,\hat{w} + d\\ p &= \Pi\,\hat{w} + \pi\end{aligned}}$$

Critical Regions In practice, the forecast parameter $\hat{w}$ is drawn from a set

$$W = \bigcup_{i=1}^{k} [0,1]^i.$$

For any $\hat{w}$ lying within a neighborhood where the active constraints remain unchanged, the affine solution holds. We define this neighborhood, termed a critical region (CR), as:

$$R = \{\hat{w} \in W \mid (A\Pi - B)\hat{w} < b - A\pi, \quad D\,\hat{w} + d \geq 0\}$$

Within each critical region, the optimal solution can be computed directly from the affine mapping without re-solving the DC-OPF.

Sample $\hat{w}$ from W and check whether it belongs to an already identified critical region. If not, solve the DC-OPF to determine the new active constraint set and then construct the corresponding critical region.

For a 5-Bus PJM testbed with 3 variable net loads, we can identify that it has 7 distinct critical regions, each having a fixed set of active constraints.

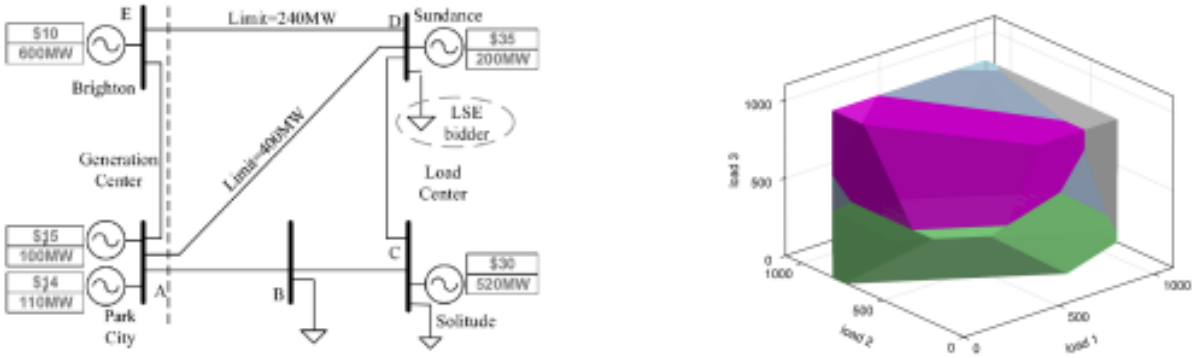

Figure 10.1: 5-Bus PJM system with its critical regions

Integration into Decision-Focused Learning We consider a decision-focused learning problem:

$$\min_{\theta} \; R \;\equiv\; \frac{1}{2k}\sum_{i=1}^{k}\Big\| \, p^*\big(W_\theta(\phi_i)\big) \;-\; p_i^* \, \Big\|_2^2$$

where p_i^* is the observed decision under the true parameters, and $p^*(W_\theta(\phi_i))$ is the decision made using the forecast from the model W_θ.

Using stochastic gradient descent, the gradient of R with respect to θ is:

$$\frac{\partial R}{\partial \theta} \;=\; \Big(p^*\big(W_\theta(\phi_i)\big) - p_i^*\Big) \;\cdot\; \frac{\partial p^*}{\partial W_\theta(\phi_i)} \;\cdot\; \frac{\partial W_\theta(\phi_i)}{\partial \theta}$$

Since we assume a known affine mapping in a given critical region,

$$p^*\big(W_\theta(\phi_i)\big) \;=\; \Pi\, W_\theta(\phi_i) + \pi,$$

it follows that

$$\frac{\partial p^*}{\partial W_\theta(\phi_i)} \;=\; \Pi^\top$$

Hence the gradient becomes

$$\frac{\partial R}{\partial \theta} \;=\; \Big(\Pi\, W_\theta(\phi_i) \;+\; \pi \;-\; p_i^*\Big) \;\cdot\; \Pi^\top \;\cdot\; \frac{\partial W_\theta(\phi_i)}{\partial \theta}$$

Advantages

- If we know in advance which critical region $W_\theta(\phi_i)$ belongs to, then we already have an affine mapping for p^*. We do not need to re-solve the full DC-OPF at each SGD iteration.
- In some applications, we may similarly keep track of the dual variable λ. If λ has a closed-form affine mapping, we can define a dual decision error objective and perform gradient updates.